

Review Article

Vision-Based Tracking of Uncooperative Targets

Suresh K. Kannan,¹ Eric N. Johnson,¹ Yoko Watanabe,² and Ramachandra Sattigeri³

¹ School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

² Department of Systems Control and Flight Dynamics, ONERA, 31055 Toulouse, France

³ Guided Systems Technologies, Stockbridge, GA 30281, USA

Correspondence should be addressed to Suresh K. Kannan, kannan@gatech.edu

Received 2 July 2010; Revised 6 November 2010; Accepted 28 February 2011

Academic Editor: Yu Gu

Copyright © 2011 Suresh K. Kannan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a summary of a subset of the extensive vision-based tracking methods developed at Georgia Tech. The problem of a follower aircraft tracking an uncooperative leader, using vision information only, is addressed. In all the results presented, a single monocular camera is used as the sole source of information used to maintain formation with the leader. A Kalman filter formulation is provided for the case where image processing may be used to estimate leader motion in the image plane. An additional piece of information, the subtended angle, also available from computer vision algorithm is used to improve range estimation accuracy. In situations where subtended angle information is not available, an optimal trajectory is generated that improves range estimation accuracy. Finally, assumptions on the target acceleration are relaxed by augmenting a Kalman Filter with an adaptive element.

1. Introduction

The most commonly used set of sensors required for accurate state estimation of a flight vehicle is a combination of an Inertial Measurement Unit (IMU) and a Global Positioning System (GPS). In most cases, heading information may be estimated when the vehicle is in motion. A magnetometer may be used if an accurate measurement is required even if the vehicle is not in motion, a helicopter for example. The IMU and Magnetometer rely on measuring quantities that are always available. The GPS signal however, is an external signal that is human generated. By its very nature of being a low-power signal, it is susceptible to jamming. Indeed, GPS simulators are available that are able to emulate GPS signals within the confines of a research lab. In the case of flight vehicles flying through terrain such as an urban city or enclosed areas, where GPS signals degrade, the translational variable estimates such as position and velocity also degrade in accuracy. Thus, there is a desire to develop reliable and affordable navigation alternatives that do not require an external signal.

Birds, insects, or, for that matter, most organisms with the ability to see the visual spectrum are able to navigate

(in flight or on the ground), quite reliably using vision information. Highly evolved organisms, especially predators, use a pair of eyes to allow stereo vision ability in order to estimate range to targets. In contrast to the IMU, most animals use relatively low-quality mechanical sensors equivalents, but use vision extensively to determine relevant state information. If a human pilot can visually acquire a target, he or she can usually fly to it; hence an automated system should be able to replicate the same behavior. From as early as the year 2000, the UAV Research Facility (UAVRF) at Georgia Tech has been interested in producing navigation solutions using a single vision camera. In this paper, we consider the specific problem of navigating (following) a target object (leader) using a single vision camera. One of the earliest problems addressed was the estimation, control, and guidance of a glider [1, 2] towards a target window opening, after launch, in order to deliver a simple payload. The only sensor available on the glider was a single camera. Subsequently, under a Multi-University Research Initiative (MURI) program, the Active-Vision Control Systems project, initiated in 2005, a large body of theoretical work, simulation, and flight testing.

Vision-based estimation and navigation methods depend greatly on the application at hand. Some early work on landing an unmanned vehicle using vision have been reported in [3], where multiple views of a landing target from a single camera are used to provide motion estimates in order to land an unmanned helicopter on a moving platform. A more recent result [4] uses natural landmarks and a feature tracking algorithm to provide SLAM solutions for high-altitude UAVs. Both these methods use homography-based techniques to compute vehicle relative translations and rotations. An impressive result is provided in recent works by Chiba University and Universite de Technologie de Compiègne and is reported in [5]. Here, optic flow and inertial measurements are used to compute the position and velocity of a vehicle carrying a single monocular camera. An interesting and useful aspect of [5] is the combined analysis of image processing and controller-plant stability results. In many of these works and related references therein, the approach used is strongly influenced by the task at hand. However, if one were to separate the different phases of the estimation problem, it is the image processing data reduction tasks that caters most directly to the environment the vehicle operates in. Once raw relative motion data is available, an estimator is generally used to fuse the vision-based data with other sensors to maintain an estimate of the vehicle's state.

The focus of this paper is to present tracking and estimation methods to effectively track an uncooperative target using monocular vision only and fly in formation with it. Even though the leader may have its own navigation solution, this information is not used (for practical flight testing purposes, however, in order to setup the initial formation, leader GPS information is indeed used. It is then slowly phased out with vision information being phased in for leader relative position estimation purposes) by the follower aircraft. The goal is to use monocular vision in order to estimate and track the relative position to the target. In this paper, the terms leader and target are used interchangeably. The leader-follower problem appears in many ways. For example, navigating relative to a known landmark (fixed target) or avoiding an obstacle, both, involve estimating the target position and maintaining a safe offset from it. This paper will address only the tracking and estimation part of the problem. The particular image processing algorithms used in our application are described in [6].

The overall approach is to use a vision camera to produce a 2-D image plane representation of the scene. This setup is shown in Figure 1. The 2-D image produced is discrete, has a finite resolution determined by the number of pixels, and can only see a subset of the scene because of a limited field-of-view. This 2-D image is the only source of information on the target, that may be *measured*. If the target undergoes any motion *relative* to the follower, those changes are readily visible in the follower's 2-D image of the scene, as long as the target is in the field-of-view. Image processing techniques may be used to track the target's movement on the image plane. Changes in range from camera to target will not be easily observable due to the difficulty in determining depth perception with monocular vision. The bioinspired solution would be to use stereo vision to improve range estimates.

Using just a single camera, the range estimate will have to be improved using other techniques. It is known that the target in question has certain dimensions. Image processing techniques may be used to compute to angle that the 2-D image of the target subtends on the image plane (essentially, size of the blob) and may be used to improve range estimates. If the target is far away, or small, the finite resolution of the 2-D images may prevent this approach. In such cases, the camera's trajectory relative to the target may be modified. Unlike stereo vision where two perspectives of the scene at the same instant is available, monocular vision can only rely on image sequences and camera motion to improve range estimates. Additionally, sophisticated image processing may be used to leverage any shape/size information that may be available in the 2-D image.

The paper is structured as follows, Section 2 provides a problem statement that illustrates the equations governing the relative dynamics. Image processing techniques are based on [6] and extended to use the Fast Level Set Marching Method [7] for computational speed. Section 3 provides the Extended Kalman Filter formulation for the navigation problem where image processing is able to provide subtended angle information. Simulation and Flight Test results for this case are provided. Next, the scenario where subtended angle information is not available due to lack of image processing or when the target is very far away is considered in Section 4. Here, the problem of minimizing the variance on the range estimation errors is treated as an optimization problem, and the resulting guidance trajectory is compared to the nominal guidance policy. Simulation results are provided. Until Section 4, the Kalman Filter formulation assumes a model for target maneuvering. For example, in Section 3.3 flight test results are described where the target is moving in a circle. This is used as a priori knowledge and used to model the target acceleration characteristics. In order to mitigate this problem of having to assume a target model, Section 5 introduces a Kalman Filter augmented with an Adaptive Element that is able to deal with unmodeled/unknown target accelerations. Simulation results for this case are also provided. In general, the following sections outline the core theory associated with each method and provides simulation and/or flight test results. In each section, the reader is referred to publications where further details may be found. The Extended Kalman Filter (EKF) formulation used for Section 3 is described in the Appendix.

2. Problem Formulation

Let \mathbf{X}_v and \mathbf{V}_v be a vehicle's position and velocity vectors expressed in an inertial frame (denoted by F_I). Suppose that the vehicle dynamics can be modeled as the following simple linear system:

$$\dot{\mathbf{X}}_v(t) = \mathbf{V}_v(t), \quad \dot{\mathbf{V}}_v(t) = \mathbf{a}_v(t), \quad (1)$$

where \mathbf{a}_v is the vehicle's acceleration input. It is assumed that all the vehicle states are available through its own-ship

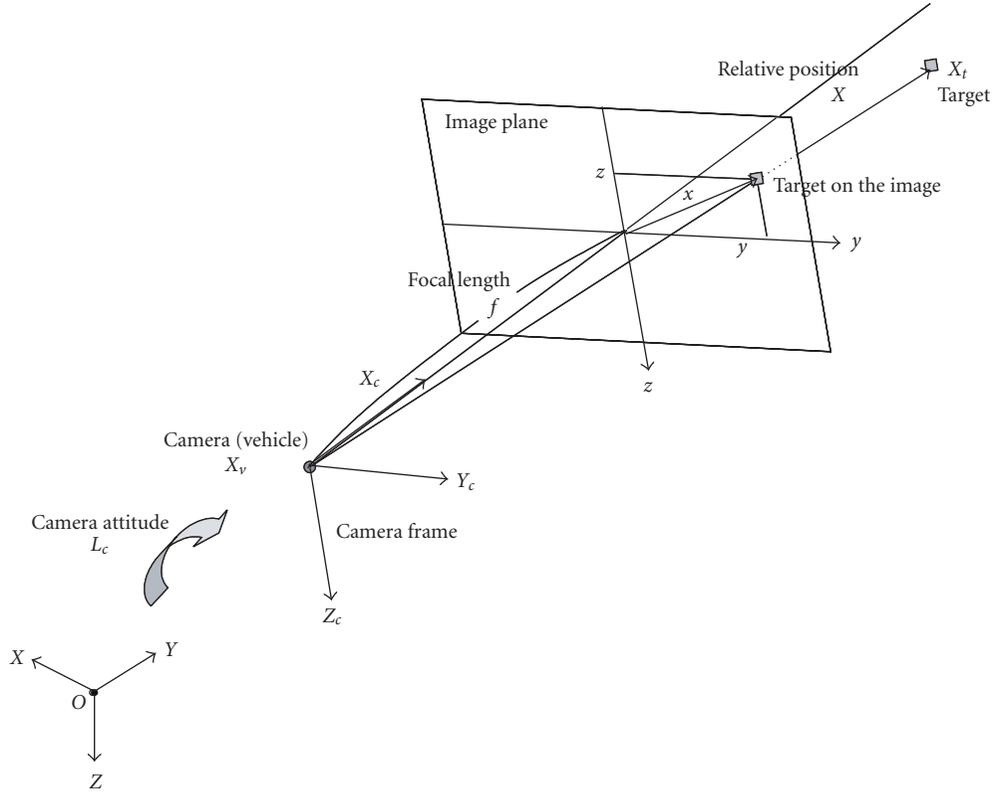


FIGURE 1: Pinhole camera model.

navigation system. The target dynamics are similarly given by

$$\dot{\mathbf{X}}_t(t) = \mathbf{V}_t(t), \quad \dot{\mathbf{V}}_t(t) = \mathbf{a}_t(t), \quad (2)$$

where \mathbf{X}_t , \mathbf{V}_t , and \mathbf{a}_t are the target's position, velocity and acceleration vectors, respectively. Relative position, velocity and acceleration of the target with respect to the vehicle are defined by

$$\begin{aligned} \mathbf{X}(t) &= \mathbf{X}_t(t) - \mathbf{X}_v(t), \\ \mathbf{V}(t) &= \mathbf{V}_t(t) - \mathbf{V}_v(t), \\ \mathbf{a}(t) &= \mathbf{a}_t(t) - \mathbf{a}_v(t). \end{aligned} \quad (3)$$

Then the relative motion dynamics are formulated as follows:

$$\dot{\mathbf{X}}(t) = \mathbf{V}(t), \quad \dot{\mathbf{V}}(t) = \mathbf{a}(t). \quad (4)$$

Since the target's maneuver is unknown in most cases, we need to apply some model for \mathbf{a}_t based on its prior knowledge. For example, the following target model is called the Singer model [8]:

$$\dot{\hat{\mathbf{a}}}_t(t) = -\alpha_s \hat{\mathbf{a}}_t(t) + \mathbf{w}(t), \quad \alpha_s > 0, \quad (5)$$

where $\mathbf{w}(t)$ represents zero mean Gaussian noise.

In this problem, for simplicity, it is assumed that a 2-D passive vision sensor is mounted at the center of gravity of the vehicle (this assumption can be easily removed by including

extra rotational dynamics caused by an offset in the relative motion model.) We also assume that an image processor which is able to detect a target position in an image frame is available. Let L_c denote a known camera attitude represented by a rotation matrix from the inertial frame F_I to a camera frame which is denoted by F_C . A camera frame is taken so that the camera's optical axis aligned with its X_c axis. Then the relative position expressed in F_C will be

$$\mathbf{X}_c(t) = L_c(t)\mathbf{X}(t) = \begin{bmatrix} X_c(t) & Y_c(t) & Z_c(t) \end{bmatrix}^T. \quad (6)$$

Assuming a pinhole camera model shown in Figure 1, the target position in the image $\mathbf{x}(t)$ is given by

$$\mathbf{x}(t) = \begin{bmatrix} y(t) \\ z(t) \end{bmatrix} = \frac{f}{X_c(t)} \begin{bmatrix} Y_c(t) \\ Z_c(t) \end{bmatrix}, \quad (7)$$

where f is the focal length of the camera. In this paper, $f = 1$ is used without loss of generality. More target information will be available if the image processor can detect the target's shape, size, contours, or other characteristics in addition to its center position \mathbf{x} . The vision-based navigation objective is to estimate the unknown target states from the image processor outputs and the known vehicle and camera states.

3. Vision-Based Formation Flight

In this section, an Extended Kalman Filter (EKF) is designed to estimate the relative state of the leader aircraft with respect

to the follower using vision information. More details on the EKF are provided in the Appendix. The EKF is implemented and tested in an image-in-the-loop 6 DoF multiple UAV flight simulation and in actual flights of UAVs. An own-ship navigation filter and a flight guidance and control system have already been developed and implemented [9]. The image processor used is developed for realtime target tracking by Ha et al. [10, 11]. In flight experiments, closed-loop vision-based formation flights of two UAVs have been successfully achieved [12]. Estimation results obtained in both simulations and flight tests are illustrated in this section.

3.1. Navigation Filter Design

3.1.1. Process Model. In the EKF design for this application, an estimation state vector is taken as follows:

$$\mathbf{x} = \left[\mathbf{u}^T \quad \dot{\mathbf{u}}^T \quad \frac{1}{r} \quad \frac{\dot{r}}{r} \quad b \right]^T, \quad (8)$$

where \mathbf{u} is an unit vector pointing from the follower to the leader and r is a range between the two aircraft. They are defined by

$$\mathbf{u} = \frac{\mathbf{X}}{\|\mathbf{X}\|}, \quad r = \|\mathbf{X}\|, \quad (9)$$

where \mathbf{X} is the relative position vector. In bearing-only tracking problems, it is very common to use an inverse of range $1/r$ instead of the range itself to reduce the nonlinearity associated with its dynamics [13]. Moreover, the unit vector is chosen as an estimation state, instead of bearing and elevation angles which are commonly used [13], to avoid a singularity. b in (8) is a constant wingspan of the leader airplane. When using the random walk model for the target's velocity, the process model is written as follows:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{u}} \\ -\left(\|\dot{\mathbf{u}}\|^2 - \frac{1}{r} \mathbf{u} \cdot \mathbf{a}_v\right) \mathbf{u} - 2\frac{\dot{r}}{r} \dot{\mathbf{u}} - \frac{1}{r} \mathbf{a}_v \\ -\frac{1}{r} \frac{\dot{r}}{r} \\ \|\dot{\mathbf{u}}\|^2 - \frac{1}{r} \mathbf{u} \cdot \mathbf{a}_v - \left(\frac{\dot{r}}{r}\right)^2 \\ 0 \end{bmatrix} + \mathbf{w} \quad (10)$$

$$= \mathbf{f}(\mathbf{x}, \mathbf{a}_v) + \mathbf{w},$$

where \mathbf{a}_v is the follower's acceleration input and \mathbf{w} is the process noise.

3.1.2. Measurement Model. It is well-known that the range information is not always observable when only a 2-D image position of a target center is measured from a single camera [14]. To guarantee range observability, the line-of-sight from the camera to the target needs to be in motion [15]. However, in the formation flight application, the objective is to maintain a constant offset from the leader, leading to poor range estimates. To overcome this observability issue, an angle that is subtended by the leader's size (e.g., wingspan)

is introduced as an additional measurement. The subtended angle α is defined by

$$\alpha = 2 \tan^{-1} \frac{b}{2r}. \quad (11)$$

In defining this subtended angle, there are some assumptions made regarding the target's orientation with respect to the camera. The most common configuration for pursuit is with the camera directly behind the target. This results in the wing span being the most convenient parameter to represent the size of the target. Hence, the difference between the camera and target heading angles is assumed to be small. If for example the heading of the target is offset 90 degrees from the camera heading, it is the fuselage length that will be visible to any image processing. Additionally, as long as the difference between the camera and target pitch angles is small, there will be no ambiguity between pitch and roll orientation of the target. The image processor developed in [10] tracks the target's contour in images, and can thus extract the target size. Figure 2 shows an example of the image tracking result. The image processor processes raw images captured by the onboard camera, and then outputs, in image coordinates, the center and the two wing-tips of the leader airplane. Let \mathbf{x}_c be an image coordinate of the center position, and let \mathbf{x}_l and \mathbf{x}_r be those of the left and right wing-tip positions, respectively. Since a pinhole camera model is assumed, the subtended angle measurement can be calculated by using \mathbf{x}_l and \mathbf{x}_r as follows:

$$\alpha = 2 \tan^{-1} \frac{\|\mathbf{x}_l - \mathbf{x}_r\|}{2}. \quad (12)$$

The measurement vector in the EKF is chosen as

$$\mathbf{z} = \left[\mathbf{u}_c^T \quad \alpha \right]^T, \quad (13)$$

where \mathbf{u}_c is the unit vector expressed in the camera frame F_C . The unit vector is chosen as a measurement instead of the two angles for the same reason as choosing it as an estimation state. The measurement vector \mathbf{z} can be expressed as a nonlinear function of the estimation state \mathbf{x} . The measurement model is given by

$$\mathbf{z}_k = \begin{bmatrix} L_{c_k} \mathbf{u}_k \\ 2 \tan^{-1} \frac{b_k}{2} \left(\frac{1}{r}\right)_k \end{bmatrix} + \boldsymbol{\nu}_k = \mathbf{h}(\mathbf{x}_k) + \boldsymbol{\nu}_k, \quad (14)$$

where L_{c_k} is the known camera attitude at the time step t_k and $\boldsymbol{\nu}_k$ is a measurement error. The EKF prediction and update procedures discussed in the Appendix are applied to the process and measurement models given in (10) and (14).

3.1.3. State Constraint Enforcement. Since the unit vector \mathbf{u} defined in (9) always satisfies $\|\mathbf{u}\| = 1$, the estimates of \mathbf{u} and its changing rate $\dot{\mathbf{u}}$ in the state vector used in the EKF design have the following constraints:

$$\|\mathbf{u}\| = 1, \quad \mathbf{u}^T \dot{\mathbf{u}} = 0. \quad (15)$$



FIGURE 2: Image tracking result: (a) original image taken by the onboard camera, (b) center and two wing-tip positions of the leader airplane are detected.

These constraints are naturally satisfied in the EKF prediction procedure (A.3), but not in the EKF update procedure (A.9). Therefore, the constraints (15) need to be enforced after the EKF update at each time step as follows:

$$\begin{aligned}\hat{\mathbf{u}}_k &= \frac{\hat{\mathbf{u}}_k}{\|\hat{\mathbf{u}}_k\|}, \\ \hat{\mathbf{u}}_k &= \hat{\mathbf{u}}_k - \left(\hat{\mathbf{u}}_k^T \hat{\mathbf{u}}_k\right) \hat{\mathbf{u}}_k.\end{aligned}\quad (16)$$

Calise [16] proposed another method to enforce the state constraints in the EKF design. In his work, the Kalman gain was computed by minimizing the augmented Lagrangian that includes the estimation error and the state constraint. This results in simply adding the correction term to the original EKF update laws (see (A.8), (A.9)). However, this approach is not used in this work to avoid using a singular measurement error covariance matrix.

3.1.4. Leader's State Estimation. Once the estimation state is updated by using a measurement at t_k in the EKF, the leader's estimated position and velocity can be calculated from the updated estimate $\hat{\mathbf{x}}_k$ and from the known follower's state as shown below

$$\begin{aligned}\hat{\mathbf{X}}_{t_k} &= \frac{\hat{\mathbf{u}}_k}{\left(\hat{1}/r\right)_k} + \mathbf{X}_{v_k}, \\ \hat{\mathbf{V}}_{t_k} &= \frac{\hat{\mathbf{u}}_k}{\left(\hat{1}/r\right)_k} + \frac{\hat{\mathbf{u}}_k}{\left(\hat{1}/r\right)_k} \cdot \left(\frac{\hat{r}}{r}\right)_k + \mathbf{V}_{v_k}.\end{aligned}\quad (17)$$

Since a nonaccelerating target is assumed in the EKF design, the leader's acceleration is estimated by $\hat{\mathbf{a}}_{t_k} = \mathbf{0}$. These estimated leader states are fed back to the guidance system to maintain the formation.

3.2. Simulation Results. Figure 3 shows a 6-DOF multi-airplane flight simulation in a formation flight configuration Figure 3(a) and the airplane model used in this simulation Figure 3(b). The airplane is the MURI research airplane with

a wingspan of 15.7 (ft) and overall length of 11 (ft). Flight control is achieved using an adaptive neural network-based controller that generates actuator signals based on the navigation system outputs and commanded position, velocity, and attitude [17]. The follower airplane has a camera, whose position, orientation and field of view generates simulated images. The synthetic images are processed and provide the locations of the leader's center and the two wing-tips. The synthetic image and the image processor outputs are shown in the left bottom window of the simulation interface in Figure 3.

In simulation, the leader airplane flies straight with a constant speed of 100 (ft/sec). The follower aircraft is commanded to maintain a given position *relative* to the leader. The formation is maintained by using the estimation result. Figure 4 compares the estimated (in blue) and the true (in green) relative positions and velocities. For example, at the start of the time scale (20 s) in Figure 4, the follower is commanded to keep the leader 100 ft ahead ($X = 100$ ft), stay directly behind ($Y = 0$ ft), and fly at the same altitude ($Z = 0$ ft) as the leader.

The vision-based estimation results match the corresponding true states well. There is a bias in the range (X position) estimation, that is, difference between the two solid lines. Even though the range estimation accuracy is improved after the vehicle's lateral motion due to the relative position command changes, a small bias remains. This is due to a measurement bias in the leader's size. However, the estimation is sufficiently accurate to achieve closed-loop vision-based formation flight. Formation breakup happens when the image processor can no longer track the target. Depending on the image processor, this happens when the target leaves or partially leaves the visual field of the camera. This may occur if X estimates are inaccurate enough where the leader overshoots the target or lags so far behind the target that the image processor is no longer able to segment the target contours. The higher likelihood of formation breakup is in the orthogonal directions, where the Y and Z relative position estimates have enough errors where the target leaves the camera's field of view. Subsequent to validation of the



FIGURE 3: (a) 6 DoF Multi-airplane Flight Simulation Interface, (b) MURI research airplane.

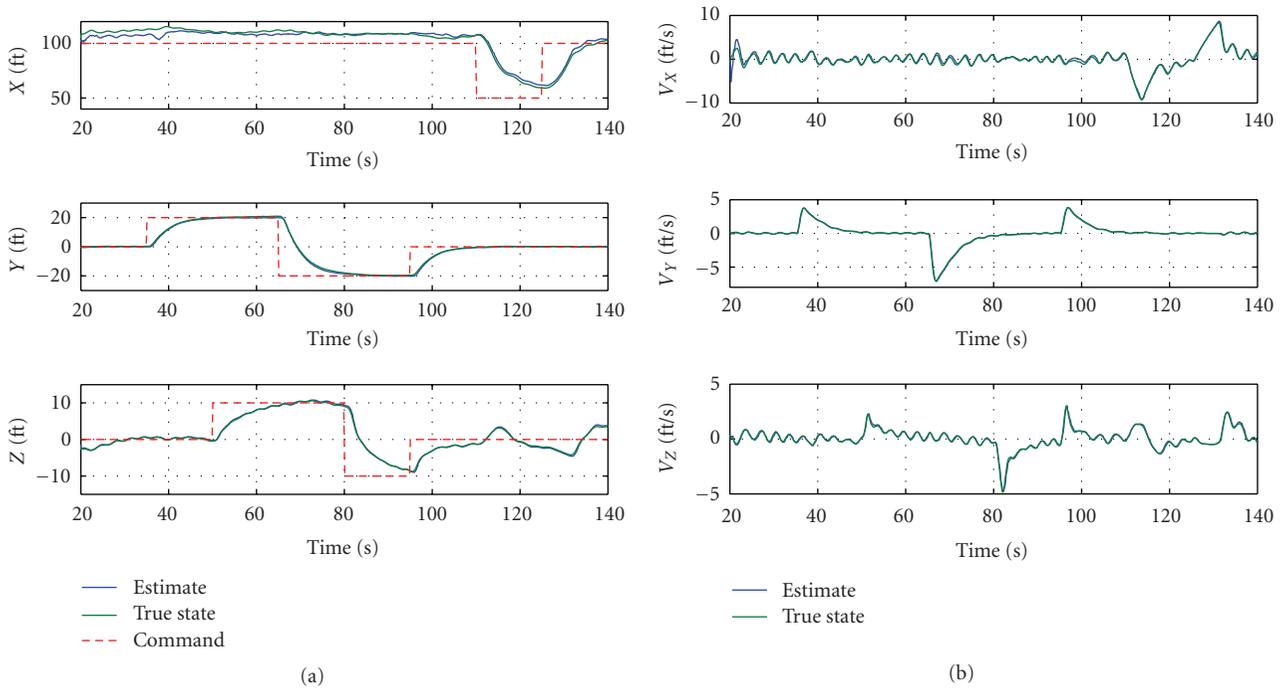


FIGURE 4: Estimated versus true relative position and velocity.

EKF-based estimation performance in simulation, flight tests were conducted and is discussed in the next section.

3.3. Flight Test Results

3.3.1. Platform Aircraft. The simulation-tested algorithms from the previous sections were implemented on two different types of UAVs, a rotary-wing aircraft, the GTMax, and a fixed-wing aircraft, the GTYak. Initially, the GTMax helicopter (based on Yamaha's RMax helicopter) was used as a follower aircraft. In later tests, the fixed-wing aircraft, the GTYak (scale model of the Yak-54), was deployed as a follower. For the vision-based guidance and control tests, the UAVs were equipped with cameras fixed to their bodies. The real-time image processor was implemented and executed within the flight computers of the GTMax and the GTYak.

The GTEdge fixed-wing aircraft was deployed as the leader for all tests. The GTEdge is a 33% scale model of the aerobatic Edge 540T airplane and is highly maneuverable with vertical hover capabilities [18]. All three aircraft of both types (fixed-wing and rotary-wing) use the same adaptive trajectory tracking controller described in Johnson and Kannan [17] for flight control purposes. Table 1 summarizes the physical specifications of the three UAVs used in the flight tests.

3.3.2. EKF Modification for Circular Target Maneuvers. During flight tests, the GTEdge was commanded to fly in a circle with a constant ground speed $V_t = 65$ (ft/s) and a constant angular rate $\omega_t = 0.1$ (rad/s). Unlike the straight level flight case used in the simulations, a circling motion was chosen because of a limited flight test field size. Since the leader's motion does not satisfy the zero acceleration assumption,

TABLE 1: Some specifications of UAV's used for formation flights.

Name	GTMax (follower) Tamaha RMax	GTyak (follower) Yak-54	GTEdge (leader) Edge540T
Model			
Wingspan (ft)	—	8.50	8.75
Rotor diameter (ft)	10.2	—	—
Overall length (ft)	11.9	7.90	7.80
Empty weight (lbs.)	157	40	35
Engine (cc)	246	100	100



FIGURE 5: GTYak and GTEdge in formation.

the EKF design discussed in Section 3.1 was slightly modified. Suppose that it is known that the leader is flying in a circle at a constant altitude. In the modified EKF design, the leader's lateral acceleration a_{lat} is added as an extra estimation state of the EKF. Since a_{lat} should be constant in a circling motion, the dynamics can be modeled by

$$\dot{a}_{lat} = 0 + w_{a_{lat}}, \quad (18)$$

where $w_{a_{lat}}$ is zero mean Gaussian noise. By assuming that the leader's lateral acceleration is perpendicular to the velocity vector in the horizontal X - Y plane, the leader's acceleration is estimated by

$$\mathbf{a}_t = \frac{a_{lat}}{\sqrt{V_{X_t}^2 + V_{Y_t}^2}} \begin{bmatrix} -V_{Y_t} \\ V_{X_t} \\ 0 \end{bmatrix}, \quad (19)$$

where the leader's velocity vector $\mathbf{V}_t = [V_{X_t} \ V_{Y_t} \ V_{Z_t}]^T$ is given in (17). In the process model (10), \mathbf{a}_v should be replaced by $\mathbf{a}_v - \mathbf{a}_t$.

3.3.3. Formation Flight Results. The first sustained closed-loop vision-based formation flight between the GTMax and the GTEdge was achieved in June, 2006. This may have been the first time automated formation flight based on vision only has ever been accomplished. Results pertaining to this particular flight test between a rotary-wing and fixed-wing aircraft are available in [12, 19].

Following these tests, it was concluded that a follower vehicle which is comparable in performance to the leader would allow flights at higher speeds and accommodate more maneuvering. Hence the GTYak was chosen to act as the leader in future experiments. The closed-loop vision-based formation flight between two fixed-wing airplanes, the GTYak and the GTEdge, was achieved in July, 2007. In this flight test, the GTEdge flew in a circle with $V_t = 75$ (ft/sec) and $\omega_t = 0.1$ (rad/sec). Since the GTYak has more maneuverability than the GTMax, a higher speed was used. The GTYak was commanded to hold a relative position of $\mathbf{X}_{com} = [80 \ 0 \ -20]^T$ (ft) from the leader. Figure 5 shows the GTYak and the GTEdge flying in formation. The image processor keeps track of every time it loses lock on the target, even briefly. This is shown in the Figure 6(a) along with image processor outputs. Although the image processor sometimes failed to capture the leader airplane, even though it was in view, it was able to recover and lock onto the leader after several frames of no lock. The right plot of Figure 6 shows estimated and true relative positions. Figure 7 compares the vision-based (solid-blue) and GPS-INS-based estimates (dashed-green) of velocity and acceleration. The vision-based estimates agree well with the GPS-INS solutions and is sufficient to maintain formation.

4. Optimal Trajectory for Range Observability

There exist cases where in (12) it may not be possible to measure α , the subtended angle, reliably. The monocular cameras used for vision have finite resolution. At large values of the range r , of the follower from the target, the image of the target on the image plane will not have enough pixel width to reliably compute subtended angle information. At these large ranges, \mathbf{x}_l and \mathbf{x}_r are close to each other and will result in a large variance in α measurements. Another situation where α may not be available is when sufficient processing power is not available to compute α . In either case, the inability to measure the subtended angle results in poor estimates of the range r . This then becomes a "Bearings-Only" [13] tracking problem where we want to estimate depth from image sequences [20]. Keeping in mind

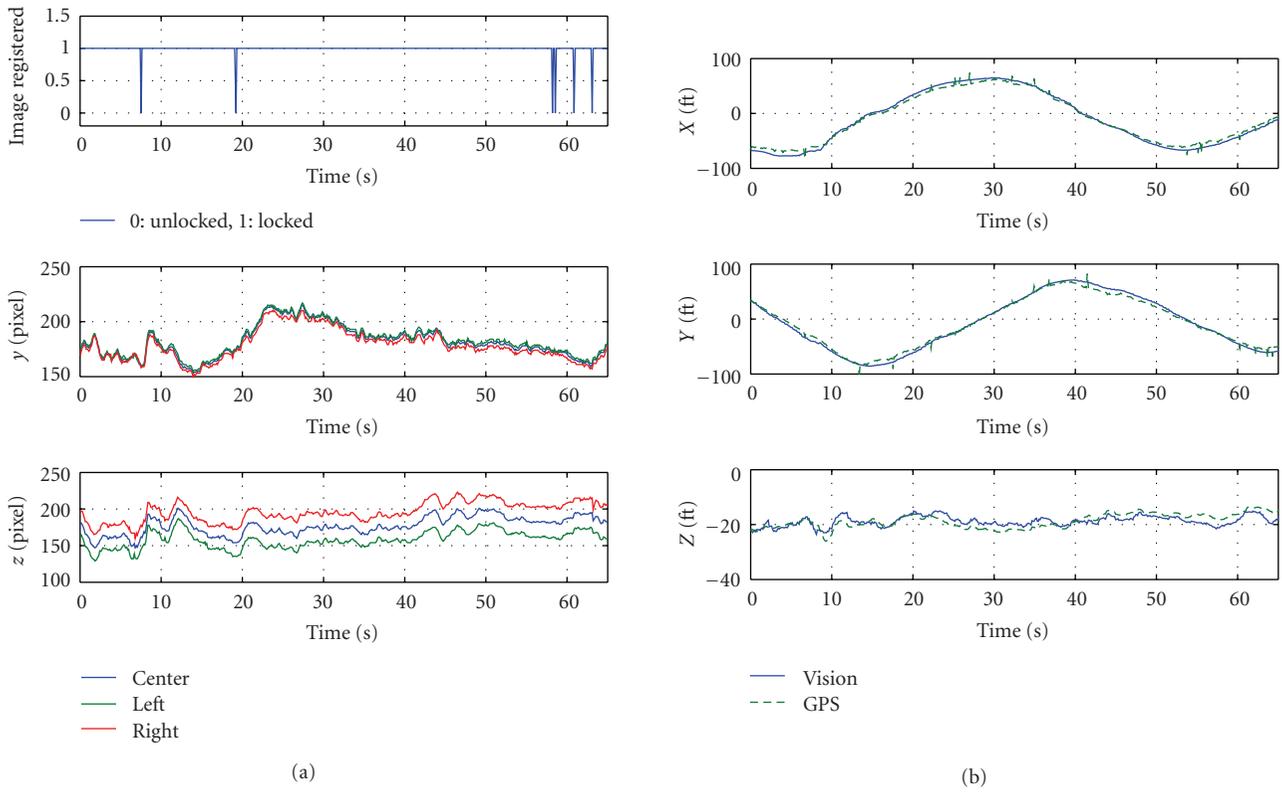


FIGURE 6: (GTyak-GTEdge) (a) image processor outputs, (b) vision-based versus GPS-based estimated relative position.

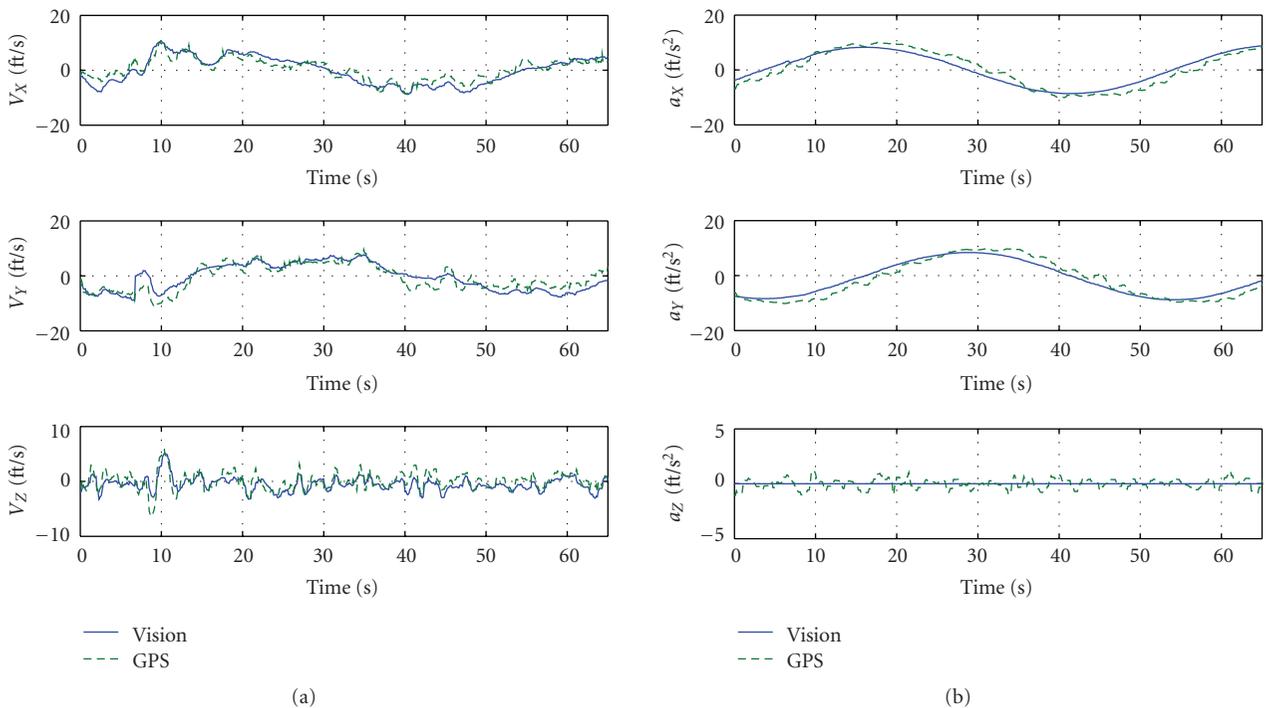


FIGURE 7: (GTyak-GTEdge) Vision-based versus GPS-based estimated relative velocity (a), and acceleration (b).

that we have used the unit vector \mathbf{u} to parameterize any bearing information to avoid singularities, we are interested in improving the accuracy of the range estimation.

A rigorous evaluation of the observability of depth information may be computed using the extended output Jacobian-based observability condition put forth in the works of Mariottini [21, 22]. Some of the motivation for using a nonlinear observability condition arises from the geometry of the problem being nonlinear where a linearization of the equations is nonobservable [23]. However, intuitively, β , the bearing to target, has to be nonzero in order for range to be observable. We can come to the same conclusion by examining P_k , the error covariance matrix of our Kalman Filter implementation described in the Appendix. The objective is to maximize P_k^{-1} , which may be achieved by maximizing the term corresponding to the range estimation error in the term $H_k^T R_k^{-1} H_k$. This particular term, if expanded, is given by

$$\left(\frac{\hat{u}_{y_k}^-}{\sigma_{\xi_y}}\right)^2 + \left(\frac{\hat{u}_{z_k}^-}{\sigma_{\xi_z}}\right)^2. \quad (20)$$

We also desire to limit the magnitude of control in y and z directions, that is, minimize a_y, a_z . The optimization problem may then be stated as

$$\min_{a_y, a_z} \frac{1}{2} \int_0^{t_f} \left\{ -\left(\frac{\hat{u}_{y_k}^-}{\sigma_{\xi_y}}\right)^2 - \left(\frac{\hat{u}_{z_k}^-}{\sigma_{\xi_z}}\right)^2 + K_y^2 a_y^2 + K_z^2 a_z^2 \right\} dt \quad (21)$$

subject to the camera dynamics given by (1), since it is attached to the vehicle and here we assume it is not rotating with the follower's body frame. Here, t_f represents the terminal time when the vehicle reaches the destination.

In order to derive an analytical solution to the optimization, the following assumptions are made.

- (i) The camera motion is limited to the X - Y plane, and the Z position is held constant. This allows $\hat{u}_{z_k}^-$ to be assumed constant and $a_z = 0$. Thus the second and fourth terms of the performance index in (21) may be eliminated.
- (ii) To further simplify the problem, \dot{u}_x may be assumed constant and $a_x = 0$.

This reduces the problem to having just two states u_y and \dot{u}_y . The boundary conditions are given by the initial and terminal states

$$\begin{aligned} u_y(0) &= u_{y0}, \\ \dot{u}_y(0) &= \dot{u}_{y0}, \\ u_y(t_f) &= u_{yf}, \\ \dot{u}_y(t_f) &= \dot{u}_{yf}. \end{aligned} \quad (22)$$

The analytical solution for this problem may be obtained by solving the Euler-Lagrange equations [24]. With the Hamiltonian defined as

$$H = -\frac{1}{2} \frac{u_y^2}{\sigma_{\xi_y}^2} + \frac{1}{2} K_y^2 a_y^2 - \lambda_1 \dot{u}_y + \lambda_2 a_y, \quad (23)$$

the resulting Euler-Lagrange equations may be expressed as

$$\begin{aligned} \dot{\lambda}_1 &= -\frac{\partial H}{\partial u_y} = \frac{u_y}{\sigma_{\xi_y}^2}, \\ \dot{\lambda}_2 &= -\frac{\partial H}{\partial \dot{u}_y} = \lambda_1, \end{aligned} \quad (24)$$

$$\frac{\partial H}{\partial a_y} = K_y^2 a_y + \lambda_2 = 0,$$

where λ_1 and λ_2 are Lagrange multipliers. A differential equation for the follower's lateral acceleration a_y can be derived and is given by

$$\frac{d^4 a_y}{dt^4} = \frac{1}{(\sigma_{\xi_y}^2 K_y)^2} a_y. \quad (25)$$

The solutions for $u_y(t)$, \dot{u}_y , and $a_y(t)$ is given by

$$\begin{aligned} u_y(t) &= -\frac{\sigma_{\xi_y}}{K_y} (A \sin \omega t + B \cos \omega t - C e^{\omega t} - D e^{-\omega t}), \\ \dot{u}_y(t) &= \frac{\sqrt{\sigma_{\xi_y} K_y}}{K_y} (A \cos \omega t - B \sin \omega t - C e^{\omega t} + D e^{-\omega t}), \\ a_y(t) &= -\frac{1}{K_y^2} (A \sin \omega t + B \cos \omega t + C e^{\omega t} + D e^{-\omega t}), \end{aligned} \quad (26)$$

where the constants A, B, C, D may be found by satisfying the boundary conditions given by (22), and the parameter ω is given by

$$\omega = \frac{1}{\sqrt{K_y \sigma_{\xi_y}}}. \quad (27)$$

Small values of K_y give larger lateral maneuvers, resulting in better estimates for the range information. In practical applications, too large a lateral maneuver can quickly result in the target moving out of the view of the camera. If this is a situation where the target is so far away that the subtended angle cannot be used, small values of K_y are acceptable because larger lateral deviation will not necessarily move the object out of the camera's field of view. However, if this is a situation where the target is close, and the image processor is such that subtended angle is not available, K_y will have to be increased as the range decreases. In general, we want to keep K_y as high as possible resulting in less maneuvering in the follower. Thus, the variance of the range estimation error may be used to schedule K_y by decreasing K_y as the variance increases.

Figure 8 shows the estimation and guidance results with and without the optimal guidance policy. In both cases, a step command is issued in the x -axis and y -axis, from an initial commanded relative position of [100, 10, 0] (ft) to [50, 0, 0] (ft). In Figure 8(b) the guidance policy is only turned on from $t = 20$ s to $t = 60$ s. The y -axis relative position plot in Figure 8(b) shows the meandering lateral

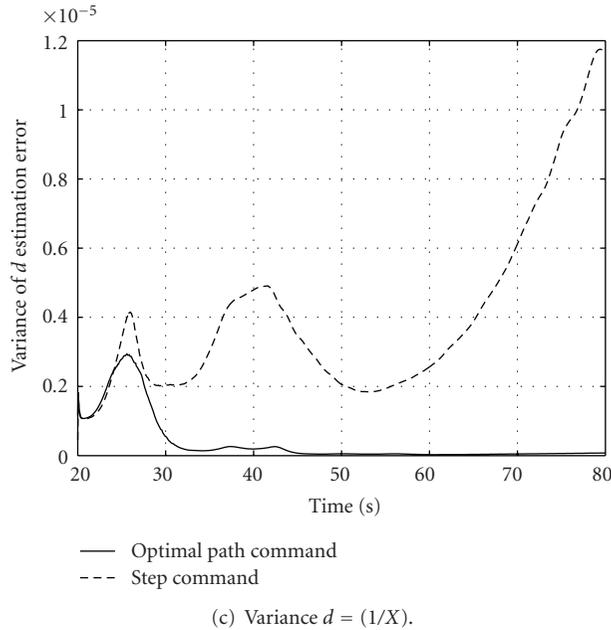
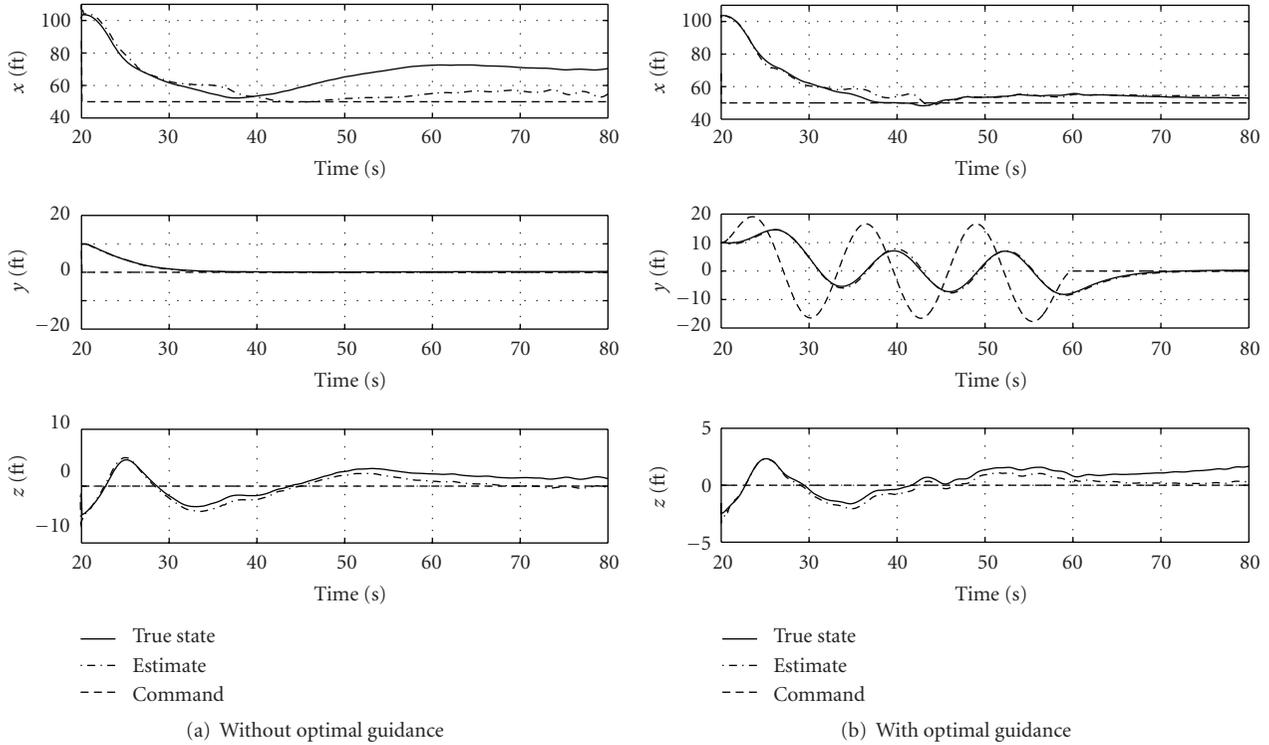


FIGURE 8: Optimal trajectory for range estimation.

path generated due to the optimal policy. In Figure 8(a), the lack of an optimal policy results in a large 20 ft steady state error between the true state of the aircraft and the estimated state of the aircraft along the x -axis. This estimation error results in a large error between the command and the true state. In contrast, Figure 8(b), has smaller estimation errors and hence smaller command tracking errors. Finally, Figure 8(c) shows a plot of the variance in estimation of

$d = 1/X$ and conclusively illustrates the advantages of the optimal policy in reducing estimation error.

5. Adaptive State Estimation

One of the major challenges for target tracking arises from target motion uncertainty. This uncertainty refers to the fact that an accurate dynamic model of the target being tracked

is generally not available to the tracker. In addition, any measurements of the target being tracked are corrupted by noise and time delays. A Kalman filter or one of its several variants is usually used as the target state estimator, but its performance may be seriously degraded unless the estimation error due to unknown target maneuvers and other uncertainties is accounted for. Some of the most popular approaches to handling the unknown target maneuvers in the target state estimator design are based on the so-called model-based filtering techniques [25]. The models may: (1) approximate the nonrandom target maneuver as a random process with certain properties, or (2) describe typical target trajectories by some representative motion models. Examples of the former include the simple white-noise acceleration model [26], the slightly more sophisticated Markov process based Singer model [8], and the more complex interacting multiple model technique [27]. A comprehensive survey of models may be found in Li and Jilkov [25]. When there is some a priori knowledge of the target maneuver, for example, if it is known that the target is maneuvering in a circle, circular motion models like (19) can be used for designing the target state estimator. In general, for the model-based approaches to target state estimation, filter performance may not be satisfactory when the target maneuver does not comply with the model, and every approach can be defeated with a suitably chosen target maneuver. Neural Network-(NN-) based adaptive estimation and filtering techniques for state estimation design have been proposed to compensate for the modeling errors that arise due to nonlinearities and unmodeled dynamics [28, 29]. In these approaches, a nominal time-varying estimator is augmented with the output of an adaptive NN that is trained online with the residuals of the nominal estimator and with delayed values of available system measurements as inputs.

Using an adaptive approach can eliminate the need to have apriori knowledge of the target maneuvering such as those given by (5) and (19).

5.1. Problem Formulation. Consider the following bounded single-input-single-output (SISO) nonlinear system:

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + Bg(\mathbf{x}(t), \mathbf{z}(t)), \quad (28)$$

$$\dot{\mathbf{z}}(t) = \mathbf{f}_z(\mathbf{x}(t), \mathbf{z}(t)), \quad (29)$$

$$y(t) = C\mathbf{z}(t), \quad (30)$$

with initial conditions $\mathbf{x}(0) = \mathbf{x}_0$ and $\mathbf{z}(0) = \mathbf{z}_0$. Here $\mathbf{x} \in Dx \subseteq \mathcal{R}^{n_x}$ and $\mathbf{z} \in Dz \subseteq \mathcal{R}^{n_z}$ are the states of the system. \mathbf{x} represents the states that have been modeled, and \mathbf{z} represents unknown/unmodeled states. Dz, Dx are compact sets and $\mathbf{f}_z(\mathbf{x}, \mathbf{z}) : \mathcal{R}^{n_x} \times \mathcal{R}^{n_z} \rightarrow \mathcal{R}^{n_z}$ is unknown, but bounded, and represents the unmodeled dynamics. $g(\mathbf{x}, \mathbf{z}) : \mathcal{R}^{n_x} \times \mathcal{R}^{n_z} \rightarrow \mathcal{R}$ is unknown, but is uniformly bounded and continuous. It is essentially through $g(\mathbf{x}, \mathbf{z})$ that the unmodeled dynamics \mathbf{f}_z couples into the system dynamics. The dimension of the \mathbf{z} dynamics need not be known, but must be bounded. That is, n_z is unknown but bounded. The output is represented by $y \in \mathcal{R}$. The matrices (A, B, C) are known, and the pair (A, C) is assumed to be observable.

The function $g(\mathbf{x}, \mathbf{z})$ acts as the unknown system input or disturbance to the nominal linear system given by the matrices (A, B, C) . In the context of target tracking, $\mathbf{f}_z(\mathbf{x}, \mathbf{z})$ represents unknown/unmodeled target dynamics. The objective is to design a state estimator to estimate the states \mathbf{x} of the system in (28) with bounded estimation error in the presence of the unknown system input $g(\mathbf{x}, \mathbf{z})$. We start by approximating $g(\mathbf{x}, \mathbf{z})$ using a neural network parametrization given by

$$g(\mathbf{x}, \mathbf{z}) = \mathbf{W}^T \boldsymbol{\sigma}(\bar{\boldsymbol{\mu}}) + \epsilon(\boldsymbol{\mu}), \quad (31)$$

where \mathbf{W} is the *ideal* weights that results in a function approximation error ϵ . We assume that $\|\mathbf{W}\|_F \leq W^*$, $\|\epsilon(\bar{\boldsymbol{\mu}})\| \leq \epsilon^*$, and $|\bar{\boldsymbol{\mu}}| \leq \mu^*$. $\boldsymbol{\sigma}(\bar{\boldsymbol{\mu}}) = [\sigma_1(\bar{\boldsymbol{\mu}}), \dots, \sigma_N(\bar{\boldsymbol{\mu}})]^T$ is a vector of sigmoidal functions. W^* is a bound on the Frobenious norms of the ideal, unknown weights \mathbf{W} . ϵ^* is a bound on the Neural Network function approximation error ϵ . N is the number of neurons. $\bar{\boldsymbol{\mu}}$ is the input vector given by

$$\bar{\boldsymbol{\mu}}(y(t), d) = [1, \Delta_d^{(0)} y^T(t) \cdots \Delta_d^{n-1} y^T(t)], \quad (32)$$

where

$$\begin{aligned} \Delta_d^{(0)} y^T(t) &= y^T(t) \\ \Delta_d^{(k)} y^T(t) &= \frac{\Delta_d^{(k-1)} y^T(t) - \Delta_d^{(k-1)} y^T(t-d)}{d} \end{aligned} \quad (33)$$

$k = 1, 2, \dots$

$d > 0$ represents a time delay. The sigmoidal functions are smooth and uniformly bounded that is, $|\sigma_i(\bar{\boldsymbol{\mu}})| \leq 1$. The above set of equations originating with (31) essentially states that a continuous function of the states of (28) that is, \mathbf{x}, \mathbf{z} can be approximated to an arbitrary degree of accuracy by a Neural Network over a compact domain by using inputs $\bar{\boldsymbol{\mu}}(y(t), d)$ that are a finite sample of the output (y) history of the system.

5.2. Adaptive Element augmented Kalman Filter. One may design a time-varying filter to estimate the states of (28) as follows:

$$\begin{aligned} \hat{\mathbf{x}}(t) &= A\hat{\mathbf{x}}(t) + K(t)(y(t) - \hat{y}(t)) + Bv_{\text{ad}}, \quad \hat{\mathbf{x}}(0) = \hat{\mathbf{x}}_0, \\ \hat{y}(t) &= C\hat{\mathbf{x}}(t), \end{aligned} \quad (34)$$

where the Kalman gain $K(t)$ may be obtained through the following matrix differential Riccati equation [30],

$$\begin{aligned} \dot{P}(t) &= AP(t) + P(t)A^T - P(t)C^T R^{-1} CP(t) + Q, \\ K(t) &= P(t)C^T R^{-1}, \end{aligned} \quad (35)$$

where $P(0) = P_0 > 0$, $Q = Q^T > 0$, $R = R^T > 0$ are design constants. The solution $P(t)$ is bounded, symmetric, positive definite, and continuously differentiable. In (34), the term Bv_{ad} represents the Neural Network augmentation to

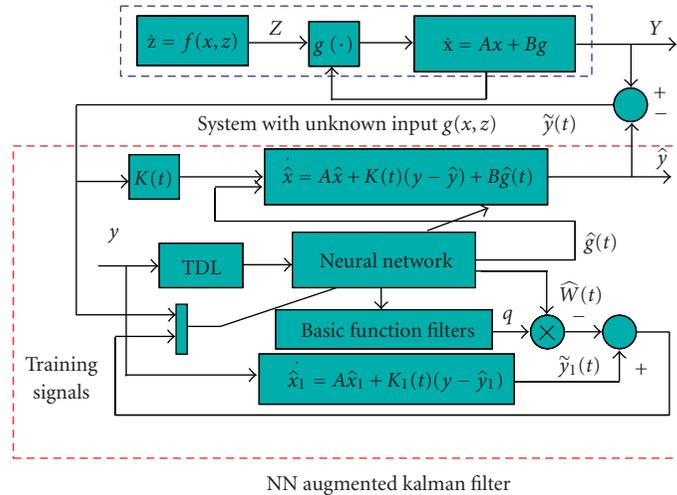


FIGURE 9: Kalman filter augmented with an adaptive element.

the Kalman Filter. v_{ad} represents the output of the Neural Network and is given by

$$v_{ad} = \widehat{\mathbf{W}}(t)^T \sigma(\bar{\mu}), \quad (36)$$

where $\widehat{\mathbf{W}}(t)$ is the estimate of the weight vector \mathbf{W} and v_{ad} is designed to approximate the bounded disturbance $g(\mathbf{x}, \mathbf{z})$. The difference between the output $y(t)$ and our estimate of the output $\hat{y}(t)$, that is, the residual signal $\tilde{y}(t) = y(t) - \hat{y}(t)$ may be used as a training signal to train the Neural Network. A second training signal is usually derived to improve the effectiveness of the NN adaptation performance. Further details on the second training signal and the Neural Network weight update laws may be found in [28]. Figure 9 illustrates a block diagram of the system. The TDL block represents the delay line give by (32).

5.3. Simulation Results. The detailed equations for this particular rendering of the estimation problem for target tracking may be found in [29]; the results, however, are presented here. In contrast to using an estimate for the target acceleration such as that given by (19), with the adaptive element augmentation, the target \mathbf{a}_t need not be assumed apriori and is simply treated as unmodeled dynamics $\mathbf{f}_z(\mathbf{x}, \mathbf{z})$ which results in an unknown input $g(\mathbf{x}, \mathbf{z})$ that the adaptive element can approximate.

In this scenario, the leader aircraft is flying in a circle in the horizontal plane at a constant heading rate. The follower aircraft is tasked with maintaining specified separation distances along the x , y , and z axes of the follower body-fixed frame. The follower is first put into the desired formation using only GPS-communicated data of the leader inertial position, velocity, and acceleration. The leader GPS data is communicated at about 5 Hz and is filtered to produce leader state estimates at the rate required (50 Hz) by the follower aircraft guidance and flight control algorithms. Once the leader aircraft is at the desired separation distance, the image processing and target state estimation algorithms

are switched on. The update rate of the image processing in simulation 10 Hz. The estimates of the leader position, velocity, and acceleration from using the vision-based target state estimator are blended in with the corresponding GPS estimates to produce the leader state estimates that are used in the guidance and flight control algorithms for formation keeping. The formation separation commands for the results shown below are given by $[dx, dy, dz]_{commanded} = [60, 15, 10]$ ft.

Figure 10(a) shows various flags used during the simulation and serves to indicate the method in which this scenario was tested. The IP flag, if 1, indicates the image processing system returned a valid result. If 0, it generally implies that the target was not in view. The use of vision information in the navigation solution was slowly blended into information available from the leader's navigation solution which is communicated to the follower. Hence a zero for useVision indicates a fully cooperative leader and a one indicates a completely uncooperative target. The NN switch flag indicates where the adaptive element is active. At $t = 140$ s, the vision estimates are blended in (at 50%) and the reliance on vision information is slowly increased, where at about $t = 290$ s, only vision information is used to estimate the leader's position, velocity, and acceleration. In doing so, Figure 10(b) illustrates the ability of the follower to maintain formation at the commanded $[dx, dy, dz]$. Figures 10(c) and 10(d) show the estimates of the leader's velocity and error in those estimates respectively. Similarly, Figures 10(e) and 10(f) show the estimates, of the leader's acceleration and error in those estimates respectively. In general, the ability of the follower to track the desired offset command is sufficient to keep formation with 15 ft in each axis.

Note that an initially cooperative target, that is, using the leader's GPS position (useVision = 0), is only assumed in order to position the leader within the follower's field of view before the experiment may begin. This assumption does not affect the ability of the Neural Network to improve

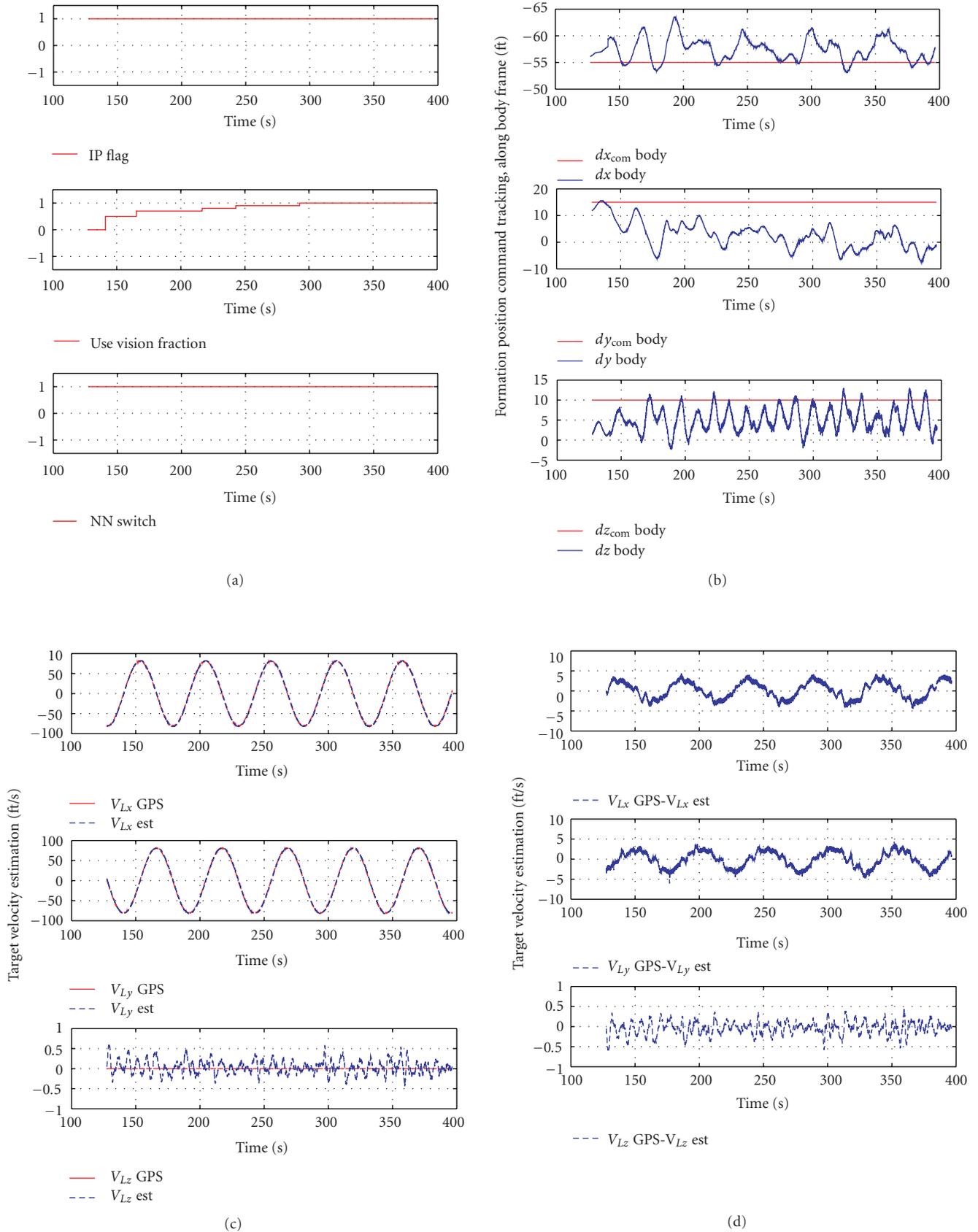


FIGURE 10: Continued.

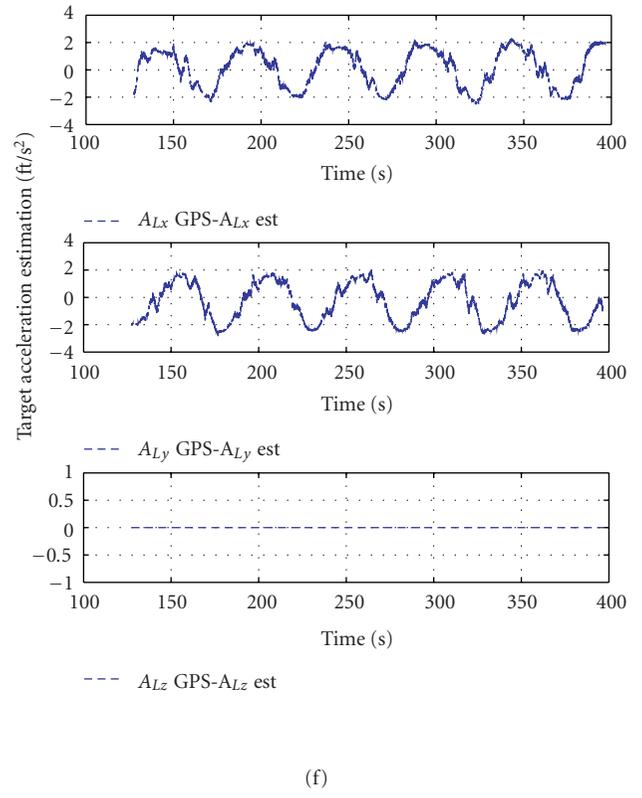
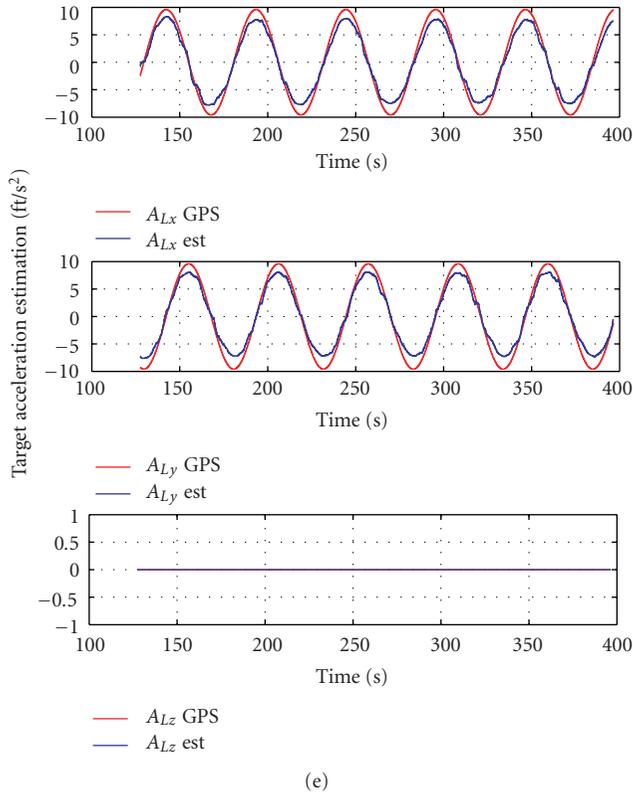


FIGURE 10: Adaptive estimation simulation results.

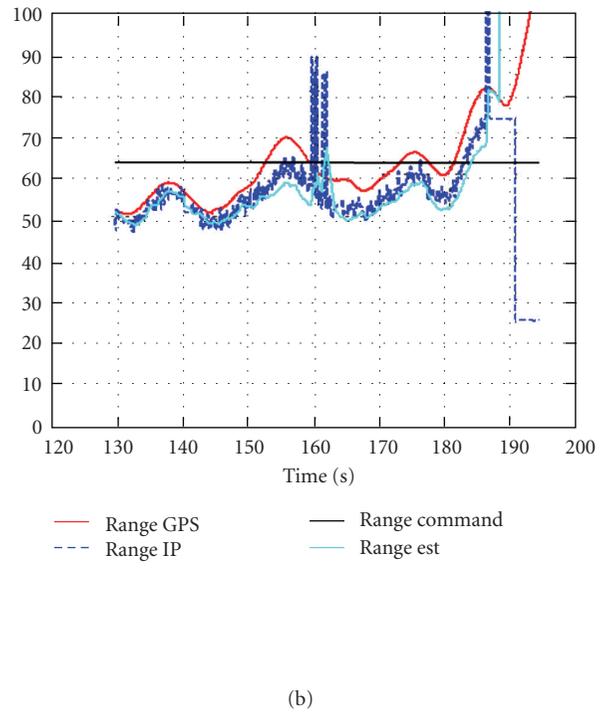
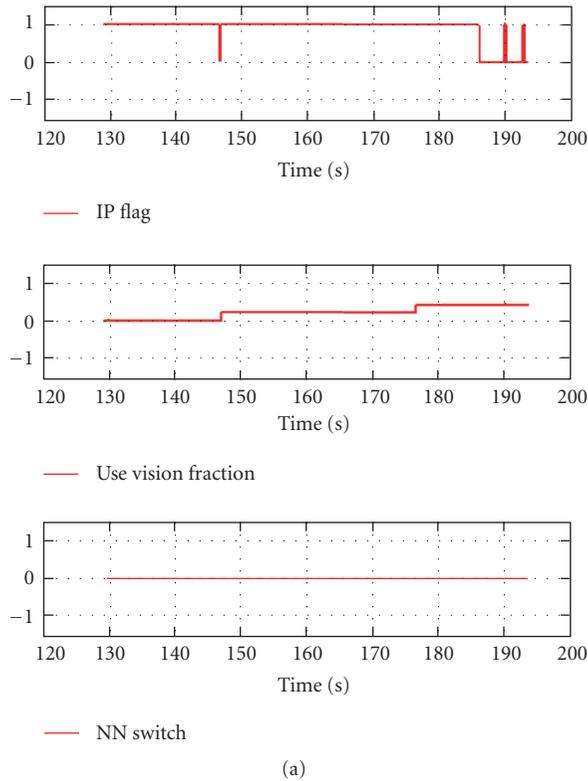


FIGURE 11: Range estimation without the adaptive element.

tracking results. Other methods such as external radar-based information may be used in a real situation where the target is truly uncooperative.

In order to demonstrate the effect of adaptation on this problem, the adaptive element was switched off leaving just the Kalman filter to estimate the leader's position and velocity. In this case only the flags (Figure 11(a)) and the range estimation error (Figure 11(b)) are shown. Even at a low blending in of vision information (20%), the IP flag temporarily goes to zero indicating a loss of vision tracking. As the vision information is further blended in, the range estimates diverges and the leader aircraft drifts out of the field-of-view, indicated by the repeated zeroing of the IP flag, eventually resulting in divergence of the range estimate. This clearly indicates the requirement of an adaptive element when no a priori information on target maneuvers is assumed.

6. Conclusions

It is shown that even though the range observability is problematic when using monocular vision, especially when target subtended angle information is not available, it is practical to generate follower maneuvers that improve range observability. A combination of flight test and simulation results are presented for vision-based tracking of an uncooperative target when subtended angle information is available. Initially, a target maneuver model which reflects the circular maneuvers performed in simulation and flight tests is assumed to be available. The achievement of the closed-loop formation flight verified the estimation performance of the vision-based relative navigation filter design. When the requirement for a target maneuver model is removed, the lack of a model was found to be detrimental to leader state estimation performance, and formation flight was not possible. By treating the target as unmodeled dynamics, an adaptive element was augmented with a Kalman Filter allowing leader states to be estimated. The adaptive element was found to be critical in maintaining formation geometry.

Appendix

Extended Kalman Filter

Since the 2-D vision-based measurement (7) is a nonlinear function with respect to the 3-D relative state, an extended Kalman filter (EKF) is applied to estimate the relative states from the measurement. The Kalman filter is a recursive solution to the least squares method for a linear filtering problem [31]. Since the filter was introduced by Kalman in 1960, it has been the subject of extensive research and application particularly in the area of autonomous navigation [32]. The EKF is an extension of the standard linear Kalman filter so that it can be applied to nonlinear systems by linearizing the system about the predicted estimate at each time step [30, 33, 34]. Even though the convergence of its estimate cannot be theoretically proven, good estimation performance of the EKF has been demonstrated in many practical applications.

A general formulation of the EKF is presented in this section. Consider the following nonlinear system:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) + \mathbf{w}(t), \quad (\text{A.1})$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \boldsymbol{\nu}_k, \quad (\text{A.2})$$

where \mathbf{x} is a state vector, \mathbf{u} is a system input, \mathbf{w} is a random zero-mean process noise, \mathbf{z}_k is a measurement at time step k , and $\boldsymbol{\nu}_k$ is a discrete measurement noise. Let $\hat{\mathbf{x}}_k^-$ and $\hat{\mathbf{x}}_k$ be the predicted and updated estimates of $\mathbf{x}(t_k)$, and let P_k^- and P_k be their estimated error covariance matrices. A state estimate at t_{k-1} is propagated to the next time step t_k through a first-order Euler integration of the original nonlinear system (A.1),

$$\hat{\mathbf{x}}_k^- \simeq \hat{\mathbf{x}}_{k-1} + \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1})(t_k - t_{k-1}). \quad (\text{A.3})$$

Then the system dynamics and the measurement model ((A.1)-(A.2)) are linearized about $\hat{\mathbf{x}}_k^-$ and discretized as follows:

$$\mathbf{x}_k = \Phi_k \mathbf{x}_{k-1} + \Gamma_k \mathbf{u}_{k-1} + \mathbf{w}_{k-1}, \quad (\text{A.4})$$

$$\mathbf{z}_k = H_k \mathbf{x}_k + \boldsymbol{\nu}_k,$$

where

$$\Phi_k = e^{F_k(t_k - t_{k-1})},$$

$$\Gamma_k = \int_{t_{k-1}}^{t_k} e^{F_k(t_k - s)} G_k ds,$$

$$\mathbf{w}_{k-1} = \int_{t_{k-1}}^{t_k} e^{F_k(t_k - s)} \mathbf{w}(s) ds,$$

$$F_k = \left. \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k^-, \mathbf{u}=\mathbf{u}_{k-1}}, \quad (\text{A.5})$$

$$G_k = \left. \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k^-, \mathbf{u}=\mathbf{u}_{k-1}},$$

$$H_k = \left. \frac{\partial \mathbf{h}(\mathbf{x}_k)}{\partial \mathbf{x}_k} \right|_{\mathbf{x}_k=\hat{\mathbf{x}}_k^-}.$$

Now the linear discrete Kalman filtering algorithm can be applied to the linearized system (A.4). The predicted and updated error covariance matrices and the Kalman gain are calculated by

$$P_k^- = \Phi_k P_{k-1} \Phi_k^T + Q_k, \quad (\text{A.6})$$

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}, \quad (\text{A.7})$$

$$P_k = (I - K_k H_k) P_k^-, \quad (\text{A.8})$$

where Q_k and R_k are covariance matrices of the discrete process noise \mathbf{w}_k and measurement noise $\boldsymbol{\nu}_k$. Finally, the updated state estimate is obtained by

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k (\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-)). \quad (\text{A.9})$$

Since a camera's field of view is limited and the image processor may sometimes fail to capture the target, the vision-based measurement is not always available. When this happens, only the EKF prediction procedure ((A.3) and (A.6)) is performed. The absence of a measurement corresponds to having a measurement with an infinitely large noise. When $R_k = \infty$ in (A.7), the Kalman gain K_k becomes zero. It results in $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^-$ and $P_k = P_k^-$, and nothing will be changed in the EKF update procedure (see (A.8) and (A.9)).

References

- [1] A. A. Proctor, E. N. Johnson, and T. B. Apker, "Vision-only control and guidance for aircraft," *Journal of Field Robotics*, vol. 23, no. 10, pp. 863–890, 2006.
- [2] C. DeWagter, A. Proctor, and E. N. Johnson, "Vision-only aircraft flight control," in *Proceedings of the 22nd Digital Avionics Systems Conference*, Indianapolis, Ind, USA, October 2003.
- [3] O. Shakernia, R. Vidal, C. Sharp, Y. Ma, and S. Sastry, "Multiple view motion estimation and control for landing an unmanned aerial vehicle," in *Proceedings of the International Conference on Robotics and Automation*, Washington, DC, USA, May 2002.
- [4] F. Caballero, L. Merino, J. Ferruz, and A. Ollero, "Vision-based odometry and SLAM for medium and high altitude flying UAVs," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 54, no. 1–3, pp. 137–161, 2009.
- [5] F. Kendoul, K. Nonami, I. Fantoni, and R. Lozano, "An adaptive vision-based autopilot for mini flying machines guidance, navigation and control," *Autonomous Robots*, vol. 27, no. 3, pp. 165–188, 2009.
- [6] J.-C. Ha, C. Alvino, G. Pryor, M. Niethammer, E. Johnson, and A. Tannenbaum, "Active contours and optical flow for automatic tracking of flying vehicles," in *Proceedings of the American Control Conference (AAC '04)*, pp. 3441–3446, July 2004.
- [7] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 93, no. 4, pp. 1591–1595, 1996.
- [8] R. A. Singer, "Estimating optimal tracking filter performance for manned maneuvering targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 6, no. 4, pp. 473–483, 1970.
- [9] E. N. Johnson and S. K. Kannan, "Adaptive trajectory control for autonomous helicopters," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 3, pp. 524–538, 2005.
- [10] J. Ha, C. Alvino, G. Pryor, M. Niethammer, E. Johnson, and A. Tannenbaum, "Active contours and optical flow for automatic tracking of flying vehicles," in *Proceedings of the American Control Conference (AAC '04)*, pp. 3441–3446, Boston, Mass, USA, July 2004.
- [11] J. Ha, E. N. Johnson, and A. Tannenbaum, "Real-time visual tracking using geometric active contours for the navigation and control of UAVs," in *Proceedings of the American Control Conference (ACC '07)*, pp. 365–370, July 2007.
- [12] E. N. Johnson, A. J. Calise, Y. Watanabe, J. Ha, and J. C. Neidhoefer, "Real-time vision-based relative aircraft navigation," *Journal of Aerospace Computing, Information and Communication*, vol. 4, no. 4, pp. 707–738, 2007.
- [13] V. J. Aidala and S. E. Hammel, "Utilization of modified polar coordinates for bearings-only tracking," *IEEE Transactions on Automatic Control*, vol. 28, no. 3, pp. 283–294, 1983.
- [14] L. Matthies, T. Kanade, and R. Szeliski, "Kalman filter-based algorithms for estimating depth from image sequences," *International Journal of Computer Vision*, vol. 3, no. 3, pp. 209–238, 1989.
- [15] Y. Watanabe, E. N. Johnson, and A. J. Calise, "Optimal 3-D guidance from a 2-D vision sensor," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pp. 319–328, August 2004.
- [16] A. J. Calise, "Enforcing an algebraic constraint in extended Kalman filter design," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pp. 1879–1891, Indianapolis, Ind, USA, August 2007.
- [17] E. N. Johnson and S. K. Kannan, "Adaptive flight controller for an autonomous unmanned helicopter," in *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 2002.
- [18] E. N. Johnson, M. A. Turbe, A. D. Wu, S. K. Kannan, and J. C. Neidhoefer, "Flight test results of autonomous fixed-wing UAV transitions to and from stationary hover," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pp. 5144–5167, August 2006.
- [19] E. N. Johnson, Y. Watanabe, J. Ha, A. J. Calise, and A. R. Tannenbaum, "Image processor, estimation, guidance, and flight test of vision-based formation flight," in *Proceedings of the 3rd International Symposium on Innovative Aerial/Space Flyer Systems*, 2006.
- [20] E. W. Frew and S. M. Rock, "Trajectory generation for constant velocity target motion estimation using monocular vision," in *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 2003.
- [21] G. L. Mariottini, F. Morbidi, D. Prattichizzo et al., "Vision-based localization for leader-follower formation control," *IEEE Transactions on Robotics*, vol. 25, no. 6, pp. 1431–1438, 2009.
- [22] G. L. Mariottini, S. Martini, and M. B. Egerstedt, "A switching active sensing strategy to maintain observability for vision-based formation control," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 2009.
- [23] A. Bicchi, D. Prattichizzo, A. Marigo, and A. Balestrino, "On the observability of mobile vehicles localization," in *Proceedings of the IEEE Mediterranean Conference on Control and Automation*, Sardinia, Italy, June 1998.
- [24] E. Bryson and Y. Ho, *Applied Optimal Control*, Taylor & Francis, London, UK, 1975.
- [25] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking, part I: dynamic models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, 2003.
- [26] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*, Wiley-Interscience, New York, NY, USA, 2001.
- [27] H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple-model algorithm for systems with Markovian switching coefficients," *IEEE Transactions on Automatic Control*, vol. 33, no. 8, pp. 780–783, 1988.
- [28] R. J. Sattigeri and A. J. Calise, "Neural network augmented kalman filtering in the presence of unknown system inputs," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, Colo, USA, August 2006.

- [29] R. J. Sattigeri, E. N. Johnson, A. J. Calise, and J.-C. Ha, "Vision-based target tracking with adaptive target state estimator," in *Proceedings of the AIAA Guidance Navigation and Control Conference*, Hilton Head, SC, USA, August 2007.
- [30] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, John Wiley & Sons, New York, NY, USA, 1997.
- [31] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME- Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.
- [32] G. Welch and G. Bishop, "An introduction to the kalman filter," Tech. Rep. TR 95-041, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 2004.
- [33] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artech House, Norwood, Mass, USA, 2004.
- [34] P. Zarchan and H. Musoff, *Fundamentals of Kalman Filtering: A Practical Approach*, AIAA, Reston, Va, USA, 3rd edition, 2004.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

