JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 24, No. 5, September–October 2001

Partial Aircraft State Estimation from Visual Motion Using the Subspace Constraints Approach

Pini Gurfil[*]
Princeton University, Princeton, New Jersey 08544
and
Hector Rotstein[†]
Technion—Israel Institute of Technology, 32000 Haifa, Israel

The estimation of an aircraft motion from the optical flow observed by a downward-looking body-fixed camera is discussed. The estimation is based on the so-called subspace constraint, which arises when points stationary on the environment are tracked on the image plane. The constraint can be combined with the aircraft dynamics, giving rise to a nonlinear estimation problem. The problem was solved using an implicit extended Kalman filter. The suggested algorithm was implemented in a simulation, which verified that the angle of attack, the angle of sideslip, and the angular body pitch, yaw, and roll rates could be estimated. The estimation was shown to be unbiased with a Monte Carlo method. Furthermore, the standard deviations of the estimation errors converged to reasonable values after a relatively small time interval. An important feature of the method is that good performance was achieved even when tracking a relatively small number of feature points, implying modest real-time computational needs. The estimated signals could be used either for navigation or control.

Nomenclature

b	=	wingspan
C_D	=	drag coefficient
C_K	=	induced drag coefficient
C_L	=	lift coefficient
C_l	=	rolling moment coefficient
C_m	=	pitching moment coefficient
C_n	=	yawing moment coefficient
C_Y	=	side force coefficient
c	=	mean wing cord
h	=	altitude
I_x	=	rolling moment of inertia
I_{xz}	=	product of inertia about xz axis
I_y	=	pitching moment of inertia
I_z	=	yawing moment of inertia
m	=	mass
N	=	number of feature points
p	=	roll rate
q	=	pitch rate
r	=	yaw rate
S	=	reference wing area
T	=	engine thrust
u	=	axial velocity component
u_g	=	wind gust along x_B axis
V_a	=	total airspeed
v	=	side velocity component
v_g	=	wind gust along y_B axis
w	=	normal velocity component
w_g	=	wind gust along z_B axis
α	=	angle of attack
β	=	angle of sideslip
δ_a	=	ailerons deflection angle
δ_e	=	elevator deflection angle
δ_r	=	rudder deflection angle
θ	=	body pitch angle
ρ	=	air density
σ	=	standard deviation
ϕ	=	body roll angle
ψ	=	body yaw angle

Subscripts

B	=	body coordinates
C	=	camera coordinates
E	=	Earth coordinates
x	=	$\partial(\cdot)/\partial x$

I. Introduction

THIS paper deals with the problem of estimating the states of an aircraft from the visual motion sensed by a camera combined with the nonlinear dynamics governing the aircraft motion. Specifically, suppose that the aircraft is provided with a body-fixed, downward-looking camera capable of acquiring images at a given frame rate. Optical flow then refers to the apparent motion of structures in the image plane of the camera resulting from the relative motion of the aircraft.[1] Given that the optical flow results from the motion of the aircraft and the structure of the scene, it is a classical problem in computer vision to investigate the inverse problem of computing egomotion from optical flow.[2] Many researchers have addressed the visual motion estimation problem using a variety of methods, such as analysis of topological manifolds,[2] epipolar constraints,[3] and implicit subspace constraints.[4] Applications of visual motion estimation include autonomous navigation,[5–7] tracking,[8] path-planning,[9] robotics,[10] and many more (see, for example, Ref. 2 and the references therein).

The existing models for motion estimation may be classified, depending on the scene descriptors employed, as point, line, curve, or model based. The simplest case is when the scene is described by a number of feature points in the Euclidean three-dimensional space.[2,4] For line-based schemes, see Refs. 6 and 7. The point-based methods may be further classified in terms of the camera model in question. The simplest cases assume ideal perspective projection (pinhole model).[2,11] That is, the location of the three-dimensional points on the scene are transformed into the two-dimensional plane by simple similarity rules.

Variations of the aircraft state estimation from the visual motion problem have been considered in the literature before. For example, in Ref. 6, the velocity-to-heightratio was estimated by using a stabilized camera and a real-time processing of visual cues. In Ref. 7 the authors used a stabilized camera and prestored terrain data to devise

Received 18 November 1999; revision received 28 October 2000; accepted for publication 29 October 2000. Copyright © 2000 by Pini Gurfil and Hector Rotstein. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission.

[*] Research Associate, Department of Mechanical and Aerospace Engineering, J228 Engineering Quadrangle; pgurfil@princeton.edu. Member AIAA.

[†] Senior Lecturer, Department of Electrical Engineering; hector@ee.technion.ac.il.

an autonomous navigation system. This implies that a digital terrain map with the desired accuracy should be stored in the navigation computer memory and that the relatively expensive computational task of terrain matching must be performed in real time. As opposed to this, the present work is not based on a prestored terrain profile. Moreover, the sensor is not stabilized, so that no gimbals and gimbal-control system are required, hence providing a more reliable and cost-effective means for vehicle navigation. Note that in contrast to other works,[7,12] no simplifications regarding aircraft dynamics are assumed. Throughout the analysis, a full six-degree-of-freedom nonlinear aircraft model is used.

This work uses a point-based model as the scene descriptor, together with a pinhole camera model. Thus, the scene is described by a number of feature points extracted from the environment using some existing image processing algorithm. The projections of these points on the image plane move according to an appropriate transformation of the aircraft motion from a body- to a camera-fixed reference frame. Assuming that the points are static (or, more generally, all belong to a rigid body) implies that the movement of the projections is constrained by an implicit kinematic relationship known as the subspace constraint.[4] Ideally, the visual motion estimation problem expressed via the subspace constraint could have been solved exactly up to an ambiguity factor by means of least-squares inversion.[4] However, because the measurement of these projections is inevitably corrupted by noise, a recursive solution scheme is required. In the present work, an implicit extended Kalman filter (IEKF) is used as an effective tool for obtaining a consistent, unbiased solution of the problem.

The visual subspace-constraint-based estimation algorithm used in this work has several novel features. Perhaps the most remarkable is that simulations have shown that it is possible to estimate the angle of attack, angle of sideslip, and the angular pitch, yaw, and roll rates in body coordinates using solely two-dimensional images. This information is of paramount importance for navigation,[13] control,[14] path planning,[15] and aerodynamic parameters identification[16] of aerospace vehicles. This ability suggests that conventional sensors such as a triad of rate gyros,[13] could, in principle, be replaced by a simple charge-coupled device (CCD) camera combined with the necessary software, which in addition to estimating the angular velocity vector, also provides valuable estimation of the angle of attack and angle of sideslip, otherwise estimated from specialized onboard sensors. In some, but of course not all, applications, this can be a very interesting feature.

The performance of the proposed algorithm is analyzed by simulation, using both a deterministic scenario and a Monte Carlo method. The simulations have validated the effectiveness of the suggested algorithm. The small number of feature points used, seven altogether, imply that no extensive real-time computational ability was required.

This work is organized as follows. The next section deals with the formulation of the visual motion estimation problem and presents the mathematical model used throughout. Section III presents the derivation of the implicit measurement equation, called the subspace constraint. Section IV formulates the equations of an IEKF used for a recursive solution of the estimation problem. Section V describes the simulation we used to verify the analysis and also considers issues of observability and robustness. Section VI concludes the discussion.

II. Problem Formulation: Mathematical Modeling

Suppose that an electro-optical sensor (e.g., a downward-looking CCD camera) is mounted rigidly on a moving aircraft. Aircraft dynamics are described by a six-degree-of-freedom nonlinear model, consists of the standard 12 differential equations,[13,17] as detailed in Appendix A. The following state vector is chosen:

$$\boldsymbol{x} = [u, v, w, p, q, r, \phi, \theta, \psi, Z_E, X_E, Y_E]^T \in \Re^{12} \tag{1}$$

where u, v, and w are the axial, side, and normal velocity components, respectively; p, q, and r the body roll, pitch, and yaw rates; ϕ, θ, and ψ the roll, pitch, and yaw angles; and Z_E, X_E, and Y_E the aircraft c.g. location in Earth coordinates, to be defined shortly.

The control vector includes ailerons deflection angle δ_a, rudder deflection angle δ_r, and elevator deflection angle δ_e:

$$\boldsymbol{u} = [\delta_a, \delta_r, \delta_e]^T \in \Re^3 \tag{2}$$

In addition, a vector of independent zero-mean white noise processes with power spectral density $\tilde{Q}$ is introduced to account for modeling errors:

$$\boldsymbol{w} = [w_1, w_2, \ldots, w_{12}]^T \in \Re^{12} \tag{3}$$

Consequently, the following nonlinear model is rendered:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) + G\boldsymbol{w}, \qquad \boldsymbol{x}(0) = \boldsymbol{x}_0$$

$$E\{\boldsymbol{w}(t)\boldsymbol{w}^T(t-\tau)\} = \tilde{Q}\delta(t-\tau) \tag{4}$$

The body-fixed downward-looking camera is a sensor, whose measurements (or constraints, in the computer vision literature) provide information about both the linear and rotational motion of the aircraft, which can hopefully be estimated from a sequence of images.

In the present work, the scene observed by the camera is a collection of N feature points in the three-dimensional space, which are projected accordingly to N points on the image plane. Projections on the image plane can be tracked by, for example, a correlation tracker, so that the position of the image points on the image plane relative to the field-of-view (FOV) center is known up to some tracking noise. It is assumed that these points are distinguishable from one frame to another. To formulate the problem mathematically, the dynamics of the aircraft should be expressed in camera axes. To accomplish this goal, the following coordinate systems are introduced (Fig. 1):

1) E is the Earth-fixed inertial reference frame. It originates at an arbitrary point on the Earth's surface at sea level. Here, x_E points north, y_E points east, and z_E completes the setup to yield a Cartesian right-hand system.

2) B is the body-fixed reference frame. It originates at the aircraft c.g. Here, x_B points toward the nose tip, y_B points toward the right wing, and z_B completes the setup to yield a Cartesian right-hand system.

3) C is the camera-fixed reference frame. It originates at the camera center of focus (COF). Here, x_C points toward the FOV center, y_C points toward the right half of the FOV, and z_C completes the setup to yield a Cartesian right-hand system.

System C is rigidly attached and rotated with respect to system B by a fixed inclination angle θ_S. It is assumed that the distance between the origins of B and C is negligible, that is, the COF is located near the aircraft c.g. This assumption is not essential, but facilitates the mathematical formulation.

The three-dimensional motion of the aircraft observed from reference frame E can be translated to reference frame C by using standard directional cosines matrices (DCMs). The DCMs utilized hereafter are as follows:

1) The transformation from E to B, denoted by T_B^E, is obtained by three consecutive rotations around the appropriate axes by the Euler angles ψ, θ, and ϕ:

$$T_B^E = \begin{bmatrix} \cos\theta \cdot \cos\psi & \cos\theta \cdot \sin\psi & -\sin\theta \\ \sin\phi \cdot \sin\theta \cdot \cos\psi - \cos\phi \cdot \sin\psi & \sin\phi \cdot \sin\theta \cdot \sin\psi + \cos\phi \cdot \cos\psi & \sin\phi \cdot \cos\theta \\ \cos\phi \cdot \sin\theta \cdot \cos\psi + \sin\phi \cdot \sin\psi & \cos\phi \cdot \sin\theta \cdot \sin\psi - \sin\phi \cdot \cos\psi & \cos\phi \cdot \cos\theta \end{bmatrix} \tag{5}$$

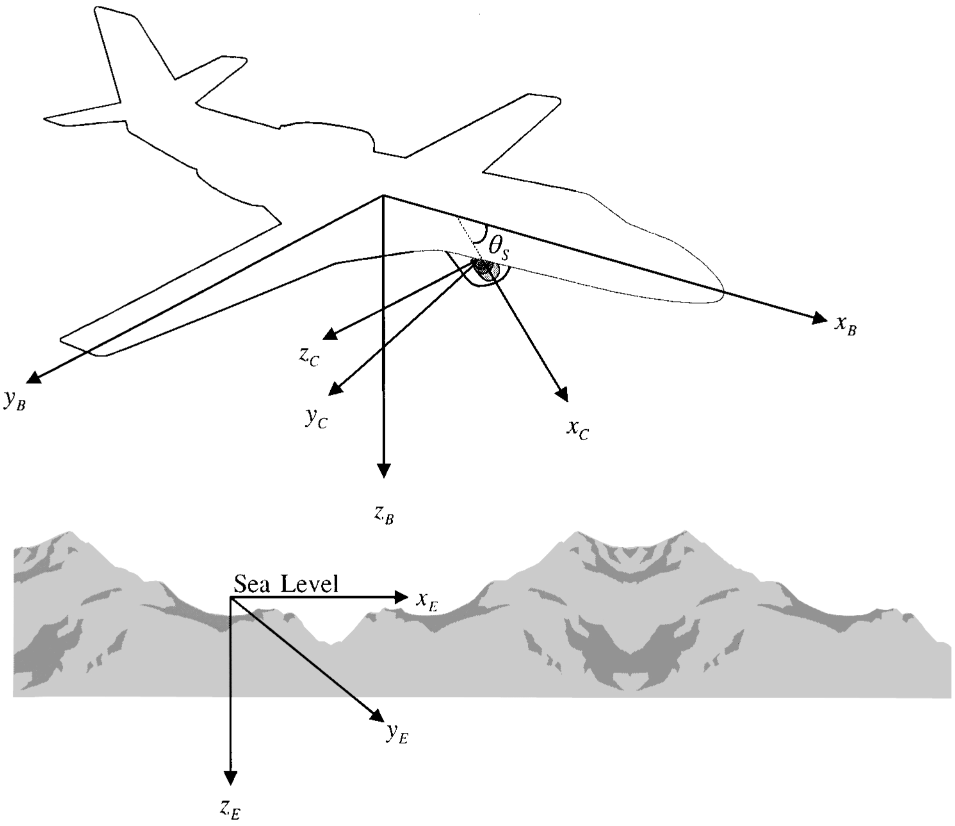

Fig. 1 Earth-, body-, and camera-fixed reference frames.

2) The transformation from B to C, denoted by T_C^B, is obtained by a single rotation around $y_B \equiv y_C$ by the inclination angle θ_S:

$$T_C^B = \begin{bmatrix} \cos\theta_S & 0 & -\sin\theta_S \\ 0 & 1 & 0 \\ \sin\theta_S & 0 & \cos\theta_S \end{bmatrix} \tag{6}$$

The line-of-sight (LOS) vector $\boldsymbol{p}$ to some feature point i in the scene E is defined by

$$(\boldsymbol{p}_E)_i = [X_F, Y_F, Z_F]_i^T - [X_E, Y_E, Z_E]_i^T \tag{7}$$

Where $[X_F, Y_F, Z_F]_i^T$ are the coordinates of the feature point and $[X_E, Y_E, Z_E]_i^T$ are the coordinates of the camera, both in reference frame E.

The LOS vector $(\boldsymbol{p}_E)_i$ can be transformed to C according to

$$(\boldsymbol{p}_C)_i = \left(T_C^B T_B^E \boldsymbol{p}_E\right)_i \tag{8}$$

to yield $[X_C, Y_C, Z_C]_i^T$.

The angular velocity vector of the aircraft in body coordinates $\omega_B = [p, q, r]^T$ can be transformed to the camera coordinate system C via multiplying by T_C^B, that is,

$$\boldsymbol{\omega}_C = [\,p_C \quad q_C \quad r_C]^T = T_C^B \boldsymbol{\omega}_B \tag{9}$$

A similar procedure is employed to transform the body velocity vector $\boldsymbol{v}_B = [u, v, w]^T$ to camera coordinates:

$$\boldsymbol{v}_C = [u_C \quad v_C \quad w_C]^T = T_C^B \boldsymbol{v}_B \tag{10}$$

Equations (4) and (7–10) give a complete description of the aircraft dynamics in camera coordinates. To complete the formulation, the perspective projection[2] or pinhole model is used to calculate the projection transformation from the scene as described in camera coordinates three-dimensional into the image plane two-dimensional. Extension of the current approach to alternative camera models is possible.

The perspective projection can be formulated as the nonlinear operator $\pi : \Re^2 \rightarrow \Re^3$, satisfying

$$\begin{aligned} \varepsilon_i &= \pi(\boldsymbol{p}_C)_i = \pi[X_C, Y_C, Z_C]_i^T \\ &= [-Z_C/X_C, Y_C/X_C]_i^T \triangleq [\varepsilon_V, \varepsilon_H]_i^T \end{aligned} \tag{11}$$

where ε_V and ε_H can be interpreted as the LOS components of a feature point as projected onto the image plane in the vertical and horizontal directions, respectively.

Under the assumption that the aircraft moves rigidly relative to the scene (i.e., there are no flexible modes between the c.m. and the center of projection of the camera), with translation velocity $\boldsymbol{v}_C$ and a rotational velocity ω_C, the three-dimensional coordinates of each feature point evolve according to

$$\begin{aligned} (\dot{\boldsymbol{p}}_C)_i &= \boldsymbol{\omega}_C \times (\boldsymbol{p}_C)_i + \boldsymbol{v}_C, \qquad [\boldsymbol{p}_C(0)]_i = \boldsymbol{p}_{Ci0} \\ \eta_i &= \pi(\boldsymbol{p}_C)_i + \boldsymbol{n}_i, \qquad \forall i = 1, 2, \ldots, N \end{aligned} \tag{12}$$

where $\boldsymbol{n}_i$ represents an error in measuring the position of projection of point i, π is the ideal perspective projection operator defined in Eq. (11), and η_i indicates the noisy projection of ε_i.

Equations (12) may be regarded as a nonlinear dynamic model having unknown parameters $\boldsymbol{p}_C$, and $\boldsymbol{v}_C$. Solving the visual motion estimation problem consists of reconstruction of the egomotion parameters $\boldsymbol{p}_C$ and $\boldsymbol{v}_C$ from all visible points, that is, estimating the states of the system from its noisy outputs. The motion reconstruction will be dealt via a least-squares inversion procedure,[4] which is the subject of the next section.

III. Generation of an Implicit Measurement Equation: Subspace Constraint

Equations (11) and (12) can be manipulated to yield an implicit measurement equation. This equation can then be used to formulate an estimation algorithm for the aircraft state vector. The treatment in this section is based on Ref. 4.

Recall that by Eq. (11)

$$\varepsilon_V = -Z_C/X_C, \qquad \varepsilon_H = Y_C/X_C \tag{13}$$

A differentiation of Eq. (13) yields

$$\begin{bmatrix} \dot{\varepsilon}_V \\ \dot{\varepsilon}_H \end{bmatrix}_i = \begin{bmatrix} -\dfrac{\varepsilon_V}{X_C} & 0 & -\dfrac{1}{X_C} \\ -\dfrac{\varepsilon_H}{X_C} & \dfrac{1}{X_C} & 0 \end{bmatrix}_i (\dot{\boldsymbol{p}}_C)_i \tag{14}$$

The expression for $(\dot{\boldsymbol{p}}_C)_i$ given in Eq. (12) can be rewritten as

$$(\dot{\boldsymbol{p}}_C)_i = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & Z_C & -Y_C \\ 0 & 1 & 0 & -Z_C & 0 & X_C \\ 0 & 0 & 1 & Y_C & -X_C & 0 \end{array}\right]_i \begin{bmatrix} \boldsymbol{v}_C \\ \boldsymbol{\omega}_C \end{bmatrix} \tag{15}$$

Substituting Eq. (14) into Eq. (15) yields

$$\dot{\boldsymbol{\varepsilon}}_i = \begin{bmatrix} \dot{\varepsilon}_V \\ \dot{\varepsilon}_H \end{bmatrix}_i = \left[\begin{array}{ccc|ccc} -\dfrac{\varepsilon_V}{X_C} & 0 & -\dfrac{1}{X_C} & -\varepsilon_H & 1+\varepsilon_V^2 & \varepsilon_V\varepsilon_H \\ -\dfrac{\varepsilon_H}{X_C} & \dfrac{1}{X_C} & 0 & \varepsilon_V & \varepsilon_V\varepsilon_H & 1+\varepsilon_H^2 \end{array}\right]_i \times \begin{bmatrix} \boldsymbol{v}_C \\ \boldsymbol{\omega}_C \end{bmatrix} \tag{16}$$

Further manipulation of Eq. (16) and transformation into matrix notation yields

$$\dot{\boldsymbol{\varepsilon}}_i = \begin{bmatrix} \dot{\varepsilon}_V \\ \dot{\varepsilon}_H \end{bmatrix}_i = \left[\begin{array}{c|c} \dfrac{1}{X_C}\mathcal{A} & \mathcal{B} \end{array}\right]_i \begin{bmatrix} \boldsymbol{v}_C \\ \boldsymbol{\omega}_C \end{bmatrix} \tag{17}$$

The key observation now is that the computation of the translation velocity can be decoupled from the computation of depth and rotational velocity. To see this, write

$$\dot{\boldsymbol{\varepsilon}}_i = \begin{bmatrix} \dot{\varepsilon}_V \\ \dot{\varepsilon}_H \end{bmatrix}_i = [\mathcal{A}_i\boldsymbol{v}_C \quad \mathcal{B}_i] \begin{bmatrix} 1/(X_C)_i \\ \boldsymbol{\omega}_C \end{bmatrix} \tag{18}$$

Because the velocity $\boldsymbol{v}_C$ multiplies each of the $1/(X_C)_i$, its magnitude can be recovered up to some scaling constant. This expresses the well-known ambiguity principle.[18] Consequently, we restrict $\boldsymbol{v}_C$ to have a unit (or constant) norm.

This is of unique importance when the problem is discussed in the context of aircraft state estimation from visual motion. Because only the relations between body velocity components can be estimated, a natural scheme for evaluating functions of these relations, such as the angle of attack α and the angle of sideslip β (see Appendix A for equations) can be established. The estimation of these angles is of prime importance, due to their role in the design of flight control,[14] path-planning,[15] and aerodynamic parameters identification[16] algorithms.

To generalize the treatment to all of the feature points tracked, stack the N equations, each of form (18), to get

$$\dot{\boldsymbol{\varepsilon}} = \begin{bmatrix} \dot{\boldsymbol{\varepsilon}}_1 \\ \dot{\boldsymbol{\varepsilon}}_2 \\ \vdots \\ \dot{\boldsymbol{\varepsilon}}_N \end{bmatrix} = \tilde{C}(\boldsymbol{\varepsilon}, \boldsymbol{v}_C) \times \begin{bmatrix} \dfrac{1}{(X_C)_1} & \dfrac{1}{(X_C)_2} & \cdots & \dfrac{1}{(X_C)_N} & \boldsymbol{\omega}_C \end{bmatrix}^T \in \Re^{2N} \tag{19}$$

where

$$\tilde{C}(\boldsymbol{\varepsilon}, \boldsymbol{v}_C) = \left[\begin{array}{ccc|c} \mathcal{A}_1\boldsymbol{v}_C & 0 & 0 & \mathcal{B}_1 \\ 0 & \ddots & 0 & \vdots \\ 0 & 0 & \mathcal{A}_N\boldsymbol{v}_C & \mathcal{B}_N \end{array}\right] \in \Re^{2N\times(N+3)} \tag{20}$$

At this point, one can solve in the least-squares sense for the inverse depth and rotation,

$$[1/(X_C)_1 \quad 1/(X_C)_2 \quad \cdots \quad 1/(X_C)_N \quad \boldsymbol{\omega}_C]^T = \tilde{C}^{+}(\boldsymbol{\varepsilon}, \boldsymbol{v}_C)\dot{\boldsymbol{\varepsilon}} \tag{21}$$

with $\tilde{C}^+$ denoting the pseudoinverse, $\tilde{C}^+ = (\tilde{C}^T\tilde{C})^{-1}\tilde{C}^T$. Substituting Eq. (21) into the differential equation (19) yields the equation

$$\dot{\boldsymbol{\varepsilon}} = \tilde{C}\tilde{C}^{+}\dot{\boldsymbol{\varepsilon}} \tag{22}$$

which forms the implicit constraint called[4] the subspace constraint:

$$\left(I_{2N\times 2N} - \tilde{C}\tilde{C}^{+}\right)\dot{\boldsymbol{\varepsilon}} \triangleq \tilde{C}^{\perp}\dot{\boldsymbol{\varepsilon}} = \boldsymbol{0} \tag{23}$$

with I denoting the identity matrix of appropriate dimension.

The subspace constraint can be exploited for recovering the direction of translation by solving the nonlinear optimization problem[4]:

$$\hat{\boldsymbol{v}}_C = \arg \min_{\|\boldsymbol{v}_C\in\Re^3\| = 1} \left\|\tilde{C}^{\perp}(\boldsymbol{\varepsilon}, \boldsymbol{v}_C)\dot{\boldsymbol{\varepsilon}}\right\| \tag{24}$$

In other words, one seeks for the best vector in the two-dimensional sphere such that $\dot{\varepsilon}$ is the null space of the orthogonal projector onto the orthogonal complement of the range of $\tilde{C}(\varepsilon, \boldsymbol{v}_C)$. If $\tilde{C}$ were invertible, the preceding constraint would be satisfied trivially for all directions of translation. However, when $2N > N+3$, $\tilde{C}\tilde{C}^+$ has rank at most $N+3$ (Ref. 4), and, therefore, $\tilde{C}^{\perp}$ is not identically zero. Consequently, it is assumed in what follows that the number of feature points selected is at least four, thus guaranteeing a nontrivial solution to the optimization problem.

To form a measurement equation suitable for estimation, the subspace constraint should be rephrased to account for feature points tracking noise. When this is the case, the constraint cannot be solved exactly, but rather be approximated by a recursive algorithm such as the IEKF, to be discussed in the next section.

IV. IEKF Formulation

When the feature points projections on the image plane are not measured exactly (i. e., the tracking is contaminated by noise), we may write

$$\boldsymbol{\eta} = \boldsymbol{\varepsilon} + \boldsymbol{n} \tag{25}$$

Thus, Eq. (23) takes the form

$$\tilde{C}^{\perp}(\boldsymbol{\eta}, \boldsymbol{v}_C)\dot{\boldsymbol{\eta}} = \tilde{\boldsymbol{n}} \tag{26}$$

where $\tilde{\boldsymbol{n}}$ is a residual noise induced by the measurement noise $\boldsymbol{n}$. Note that although $\boldsymbol{n}$ is assumed a zero-mean white noise, $\tilde{\boldsymbol{n}}$ is by no means white.

Equation (26) describes an implicit measurement equation in terms of the noisy measurements and the aircraft velocity vector calculated in camera axes. Because the aircraft dynamics (4) were formulated in body axes, it is convenient to rephrase Eq. (26) to yield an implicit measurement equation with the velocity vector taken in body axes as well. This is performed by recalling the relationship given in Eqs. (10) and (6). Thus, Eq. (26) is written as

$$\tilde{C}^{\perp}(\boldsymbol{\eta}, u, v, w, \theta_S)\dot{\boldsymbol{\eta}} = \tilde{\boldsymbol{n}} \tag{27}$$

The dynamic model (4) coupled with the implicit measurement equation (27) are now suitable for the estimation procedure.

The estimation algorithm of choice was an extended Kalman filter (EKF) adapted to deal with implicit measurements and, thus, called an IEKF. Because body dynamics are continuous in time and the measurement procedure is discrete in nature, the IEKF shall be expressed using the continuous-discrete formulation[19] suited for the implicit measurement case. The computational algorithm proceeds as follows.

1) If $\hat{x}(k\mid k)$ is known, compute $\hat{x}(k+1\mid k)$ by integrating numerically the state-space equation

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{f}(\hat{\boldsymbol{x}}, \boldsymbol{u}), \qquad \hat{\boldsymbol{x}}(0) = E[\boldsymbol{x}(0)] \tag{28}$$

Denote $\hat{\boldsymbol{x}}(t_{k+1}) = \hat{\boldsymbol{x}}(k \mid k)$ and, accordingly, the solution at t_{k+1} by $\hat{\boldsymbol{x}}(t_{k+1}) = \hat{\boldsymbol{x}}(k+1 \mid k)$.

2) With $\hat{\boldsymbol{x}}(t)$, propagate the covariance $P(t \mid k) = E\lfloor(\boldsymbol{x} - \hat{\boldsymbol{x}})(\boldsymbol{x} - \hat{\boldsymbol{x}})^T\rfloor \in \Re^{12\times 12}$ by integrating

$$\dot{P}(t \mid k) = A(\hat{\boldsymbol{x}}, \boldsymbol{u})P(t \mid k) + P(t \mid k)A^T(\hat{\boldsymbol{x}}, \boldsymbol{u}) + GQG^T \qquad t_k < t < t_{k+1} \tag{29}$$

with the initial condition

$$P(t_k \mid k) = \begin{cases} P(0) & \text{if} \quad t_k = 0 \\ P(k \mid k) & \text{if} \quad t_k \neq 0 \end{cases} \tag{30}$$

and where

$$A(\hat{\boldsymbol{x}}, \boldsymbol{u}) \triangleq \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{x}}\right|_{x = \hat{x}(k+1|k)} \in \Re^{12\times 12} \tag{31}$$

is the Jacobian of $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$ defined for the end of the sampling interval t_{k+1} and $\tilde{Q} \in \Re^{12\times 12}$ is the power spectral density of the white process noise $\boldsymbol{w}$ [see, Eq. (4)].

3) Set $P(k+1 \mid k) = P(t_{k+1} \mid t_k)$. Compute the Kalman gain matrix $K(k+1) \in \Re^{12\times 2N}$ from

$$K(k+1) = P(k+1 \mid k)L^T(k) \times \left[L(k)P(k+1 \mid k)L^T(k) + D(k)R_\eta D^T(k)\right]^{-1} \tag{32}$$

where the matrices L, D, and R_η are given by

$$L \triangleq \left\{\left[\left.\frac{\partial \tilde{C}^\perp(\boldsymbol{\eta}, \boldsymbol{v}_C)\dot{\boldsymbol{\eta}}}{\partial \boldsymbol{v}_C}\right|_{\eta, \dot{\eta}, \hat{v}_C} \cdot T_C^B\right] 0_{2N\times 9}\right\} \in \Re^{2N\times 12}$$

$$D \triangleq \left.\frac{\partial \tilde{C}^\perp(\boldsymbol{\eta}, \boldsymbol{v}_C)\dot{\boldsymbol{\eta}}}{\partial[\boldsymbol{\eta}\dot{\boldsymbol{\eta}}]}\right|_{\eta, \dot{\eta}, \hat{v}_C} \in \Re^{2N\times 4N}$$

$$R_\eta \triangleq \operatorname{cov}\begin{bmatrix} \boldsymbol{\eta} \\ \dot{\boldsymbol{\eta}} \end{bmatrix} \in \Re^{4N\times 4N} \tag{33}$$

with T_C^B given in Eq. (6) and $0_{2N\times 9}$ denoting a matrix of zeros with the given dimensions. R_η is one of the IEKF performance tuning parameters. Notice that the computation of the gain matrix involves the inversion of a relatively large matrix, but this computation can be handled by modern navigation computers.

L, D, and R_η result from the linearization of Eq. (27), which is essential for the formulation of the IEKF. The linearization is discussed thoroughly in Appendix B.

4) Compute the measurement update (correction) of the estimate from

$$\hat{\boldsymbol{x}}(k+1 \mid k+1) = \hat{\boldsymbol{x}}(k+1 \mid k) + K(k+1)\tilde{C}^\perp\dot{\boldsymbol{\eta}}(k) \tag{34}$$

Notice the equivalence between an ordinary EKF and an IEKF as expressed in the last term of the right-hand side of Eq. (34).

5) Finally, update the matrix P by using

$$P(k+1) = [I - K(k+1)L(k)]P(k+1 \mid k)[I - K(k+1)L(k)]^T + K(k+1)D(k)R_\eta D^T(k)K^T(k+1) \tag{35}$$

The modeling and estimation were implemented in a six-degree-of-freedom simulation of an aircraft, illustrating the suggested ideas and examining the performance of the proposed algorithm. The simulation is the subject of the next section.

V. Simulation

A. General Description

A simulation illustrating the suggested ideas was implemented in MATLAB®. First, the nonlinear differential equations describing the aircraft dynamics, which are given in Appendix A, were realized. The platform selected was a general aviation airplane, called Navion, with aerodynamic data adopted from Ref. 20. The values of the aerodynamic coefficients and the configuration data are given in Appendix C. The aircraft performed a slow descent combined with a roll–yaw maneuver, so that all three maneuver channels were excited. Appendix C elaborates on the computation of initial conditions and fin deflections that shaped the aircraft trajectory. Notice that an open-loop operation was assumed (i.e., no fin or throttle commands were provided by control algorithms) because the main interest of this work was to test the estimation algorithm. In addition, the first part of the analysis assumed that there were no wind gusts.

The next step was to choose the number of feature points. In general, it was found that a modest number of points suffices to provide satisfactory performance, and, consequently, a fixed number of seven feature points were used during the simulation. Although a larger set of points somewhat improves the performance of the algorithm, this improvement did not pay for the increase in computational burden.

The FOV characteristics of the camera, however, appear to be of prime importance for two main reasons. First, a wider FOV allows a better distribution of the fixation points, preventing the ill conditioning resulting from all points lying very close to each other. Second, for wider FOVs there is no need to select and track a new feature point instead of a point that exceeds the FOV limits. Note that if some point goes out of the FOV limits, a random selection of a new point will cause a discontinuous behavior of the estimated signals. Nevertheless, this discontinuity is negligible when the number of points is large enough. For the example, the FOV was computed from the data sheet of a Sony® video camera with a 0.5-in. detector and a 30-mm lens; the resulting FOV was approximately 18 × 13.5 deg. Note that the ability to deal with points entering and leaving the FOV in an almost seamless manner is one of the main features of the approach.[4]

The camera inclination angle also provides a degree of freedom for controlling the performance of the algorithm. This angle is set based mainly on two considerations: the aircraft flight trajectory characteristics (altitude, body angles) and the camera FOV dimensions. FOV size and inclination are not independent because the larger the FOV, the smaller the sensitivity to inclination angle. The link between inclination and trajectory stems from the need to provide an image with sufficient details, such that a correlation tracker would be able to track the feature points. For example, when the aircraft performs a steep ascent and the inclination angle is small, the scene image consists of mainly sky and clouds. On the other hand, at cruise, a small inclination might suffice to capture a well-textured ground image. As for the flight scenario discussed earlier, it was found that an inclination angle of −5 deg gives adequate performance.

Once the number of feature points, FOV size, and inclination angle were set, the algorithm proceeded as follows. First, seven feature points were selected at random. Then, the appropriate transformation matrices were used to convert the inertial location of the feature points to their corresponding projections in camera axes, that is, onto the image plane.

By construction, the initial feature points projections lay within the FOV limits. Because the locations of the point projections change as a function of the aircraft dynamics, eventually some point will exceed the FOV limits; at this stage, a new feature point is randomly selected, and the process continues. In real applications, the problem of selecting new feature points can be complicated from an image processing viewpoint. This difficulty is not addressed here and should be taken care of by the image processing block.

With the feature point projections at hand, the implicit constraint was calculated together with the matrices relevant for the IEKF algorithms. The recursive computation proceeds to yield an estimation of the observable state variables. The observability issue will be discussed later. The nominal test scenario assumed a tracking noise level of 0.5 m rad at 1σ for both the vertical and horizontal feature point projections. This characteristic is typical of imaging sensors.

The overall simulation structure is shown in Fig. 2. The flowchart in Fig. 2 illustrates the various stages that constitute the algorithm of state estimation from visual information.

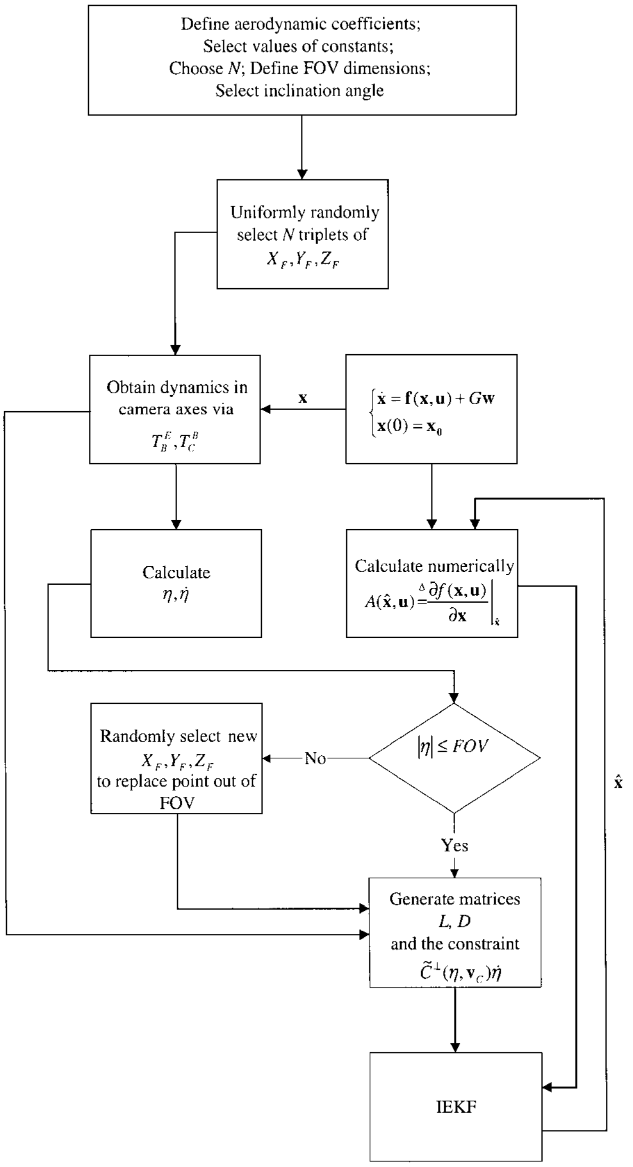
Define aerodynamic coefficients;
Select values of constants;
Choose N; Define FOV dimensions;
Select inclination angle
Uniformly randomly select N triplets of X_F, Y_F, Z_F
Obtain dynamics in camera axes via T_B^F, T_C^B
x
$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{u}) + Gw$
$\mathbf{x}(0) = \mathbf{x}_0$
Calculate $\eta, \dot{\eta}$
Calculate numerically $A(\hat{\mathbf{x}},\mathbf{u}) \triangleq \frac{\partial f(\mathbf{x},\mathbf{u})}{\partial \mathbf{x}}\Big|_{\hat{\mathbf{x}}}$
Randomly select new X_F, Y_F, Z_F to replace point out of FOV
No
$|\eta| \leq FOV$
Yes
$\hat{\mathbf{x}}$
Generate matrices L, D and the constraint $\tilde{C}^{\perp}(\eta, \mathbf{v}_C)\dot{\eta}$
IEKF

Fig. 2 Simulation flowchart.

B. Numerical Results

This section contains some numerical results obtained from the simulation described in the preceding paragraphs. The evaluation includes two parts: a deterministic evaluation and a statistical test using Monte Carlo techniques. Several robustness issues, such as sensitivity to tracking noise level and unknown wind gusts, will be also discussed. The section concludes with a brief observability analysis.

1. *Deterministic Performance*

The numerical values given in Appendix C, seven feature points, an FOV of 18 × 13.5 deg, and an inclination angle of −5 deg were used to generate a scenario for a deterministic testing of the algorithm.

The tuning parameters of the IEKF were as follows:

$$\tilde{Q} = \text{diag}(20{,}000, 20{,}000, 20{,}000, 100, 100, 100, 100, 100, 100, 100, 100, 100)$$

$$R_\eta = 0.1 \cdot I_{12\times 12} \qquad (36)$$

Figure 3 shows a comparison between the actual and the estimated body velocity components. The sluggish estimation behavior results from the ambiguity principle discussed earlier. Only the relations between body velocity components may be estimated. Thus, functions of these relations, such as the angle of attack α and the angle of sideslip β, are expected to be estimated correctly. This is verified by the examination of Fig. 4, which depicts a comparison between the actual and the estimated values of α and β. It is evident that after about 3 s the estimation converges.

As shown in Fig. 5, the algorithm also provides a good tool for the estimation of body angular rates. Figure 5 depicts the actual and estimated values of p, q, and r. After about 6 s, the estimation converges, yielding small errors. A complete quantification of the estimation errors will be appear in the next subsection.

The other state variables (i.e., the Euler body angles and the inertial location) are unobservable from the visual information. Yet, the success of estimating α, β, p, q, and r yields an efficient and simple way to generate valuable information useful both for navigation and body control purposes. Moreover, these variables can be utilized to design a complete three-axis autopilot for aircraft stabilization.

The behavior of the estimation error covariance P should be also considered when examining the overall performance of the algorithm because it reflects important aspects of the estimation process. Consequently, Fig. 6 presents the estimation error covariances of p, q, and r, which are denoted P_p, P_q, and P_r, respectively. Notice that the covariance components decrease, which implies that the estimation of these state variables converges.

2. *Statistical Performance*

In addition to the deterministic scenario discussed, a statistical examination of the estimation algorithm was performed using a Monte Carlo method. The simulation was run 300 times, with a different measurement noise seed chosen for each run, but with the same standard deviation, 0.5 mrad. The Monte Carlo results are depicted in Figs. 7 and 8.

Figure 7 presents the mean of the estimation errors of α (top) and β (bottom), together with a mean $\pm\sigma$ envelope. The mean values of the estimation errors converge to zero after about 3.5 s. After this time, the standard deviation of the estimation error reaches a value of 0.01 rad for both α and β.

Figure 8 depicts the mean values of the estimation errors of p (top), q (middle), and r (bottom), together with a mean $\pm\sigma$ envelope. The mean of the estimation error of p converge to zero after about 8.5 s. This relatively long duration stems from the low observability of the roll rate from the visual flow of images, especially when the roll rate is small. However, the mean values of the estimation errors of q and r converge to zero faster, after about 6 s. The standard deviations of the p, q, and r estimation errors at the steady state are 0.05, 0.01, and 0.04 rad, respectively.

Practically speaking, the Monte Carlo method is a most useful technique for obtaining the estimation error covariance matrix when dealing with a nonlinear estimation scheme such as an IEKF. This analysis has proved that the estimation of α, β, p, q, and r is consistent, unbiased, as well as robust to various noise regimes used to simulate the corruption of the feature point projections measurements.

C. Algorithm Robustness

Recall that both the deterministic and statistic simulations assumed the following: no wind gusts present, fixed level of Gaussian measurement noise ($1\sigma = 0.5$ mrad), and fixed number of feature points ($N = 7$). However, in practice, all of these assumptions may be violated. For example, wind gusts are present during any reasonable flight. Moreover, the measurement noise intensity might vary as a function of illumination, the quality of the electro-optical sensor, and weather conditions. The algorithm performance might change as a result. This motivates the need for a robustness analysis, which is the subject of this subsection.

1. *Sensitivity to Unknown Random Wind Gusts*

Several gust models exist in the literature. A particularly well-known model is that of Dryden (see Ref. 21), where the wind gust vector velocity field $[u_g, v_g, w_g]^T$ is formulated as an

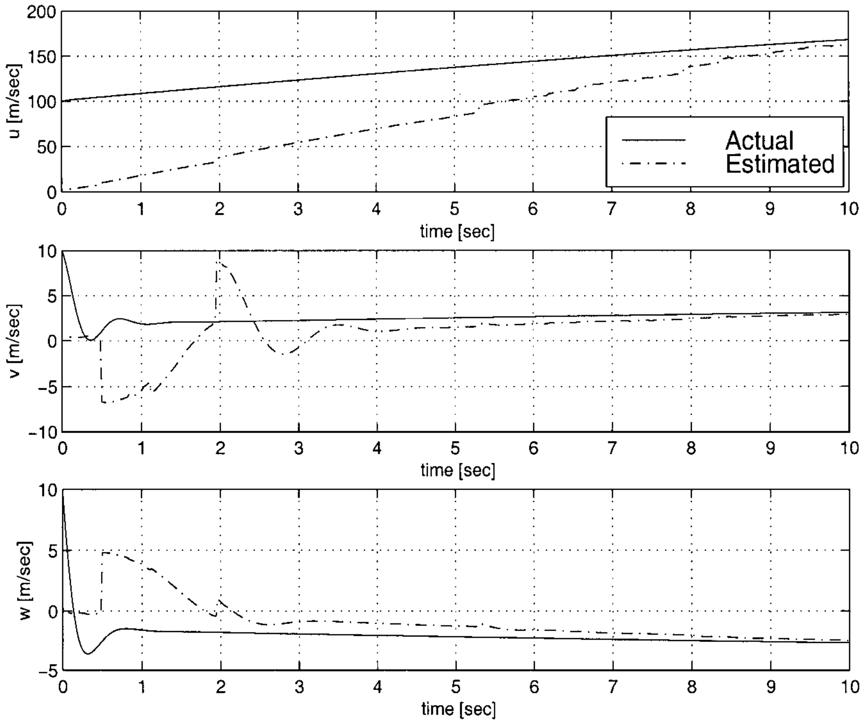

Fig. 3 Actual and estimated body velocity components; sluggish behavior results from the ambiguity principle.

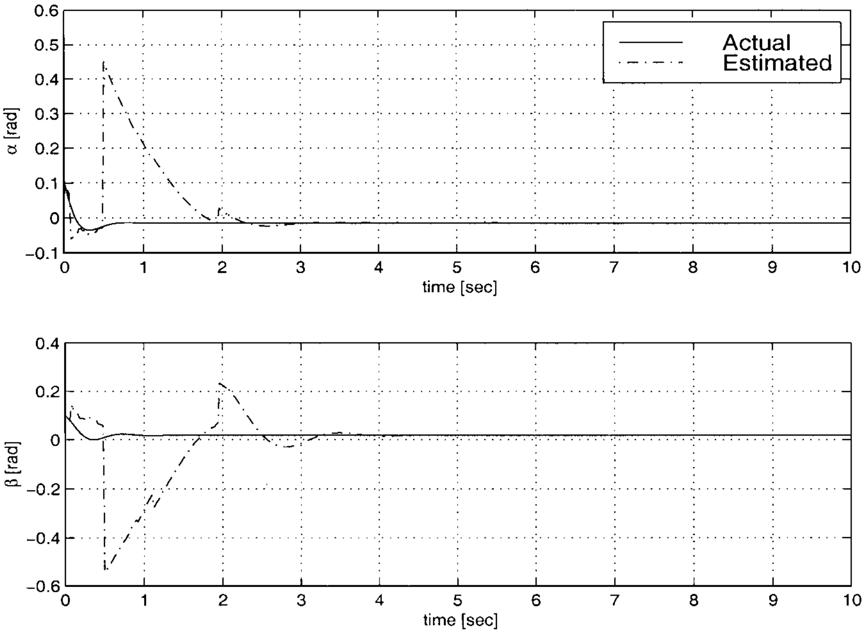

Fig. 4 Actual and estimated angle of attack and angle of sideslip; estimation converges after 3 s.

airspeed- and altitude-dependentfirst-order Gauss–Markov process with an unknown mean value. However, to better understandthe sensitivity of the estimation algorithm to random gusts, we adopt a simpler model, which represents gust components as random Gaussian constants:

$$\dot{u}_g = 0,\ E[u_g(0)] = 0, \sigma[u_g(0)] = \sigma_{u_g}$$

$$\dot{v}_g = 0,\ E[w_g(0)] = 0, \sigma[v_g(0)] = \sigma_{v_g}$$

$$\dot{w}_g = 0,\ E[v_g(0)] = 0, \sigma[w_g(0)] = \sigma_{w_g} \tag{37}$$

where

$$\sigma_{u_g} = \sigma_{v_g} = \sigma_{w_g} = 15 \text{ m/s} \tag{38}$$

Thus, every simulation run produces a different deterministic wind gust vector. The gust values taken in the estimator model were all set to zero. With 300 Monte Carlo runs, the performance of the algorithm was investigated with gust model (37). The influence of the unknown wind gust vector on the estimation quality of α and β is describedgraphicallyin Fig. 9. As expected,the estimationerrors remain unbiased, but the standard deviations are increased. However, this effect is negligible, as seen when comparing Fig. 9 to Fig. 7.

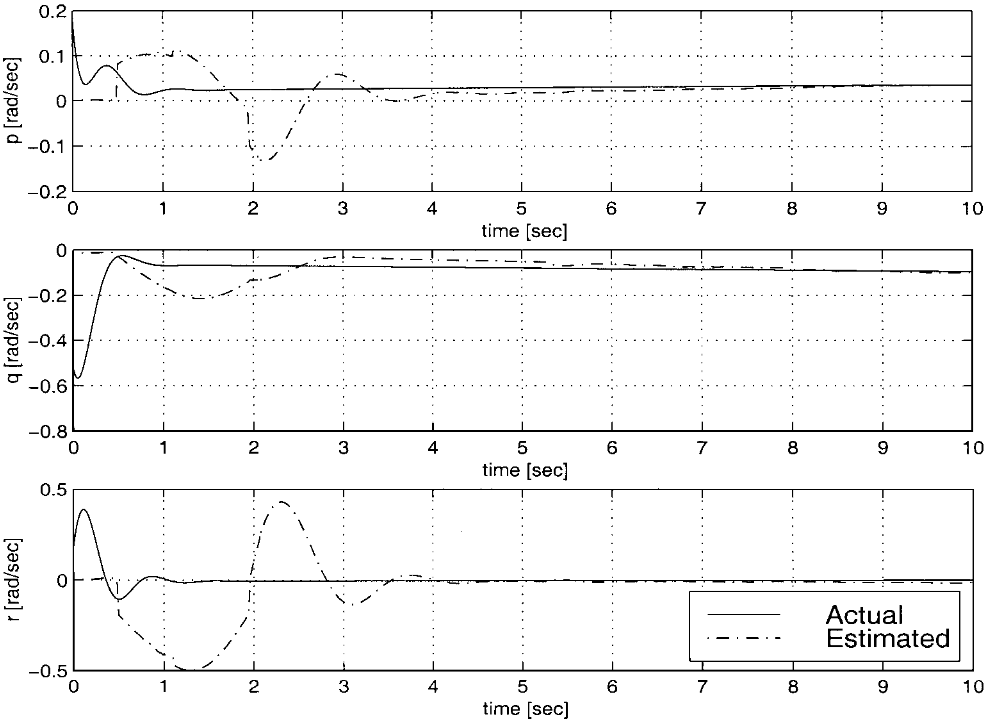

Fig. 5 Actual and estimated body rates; estimation converges after 6 s.

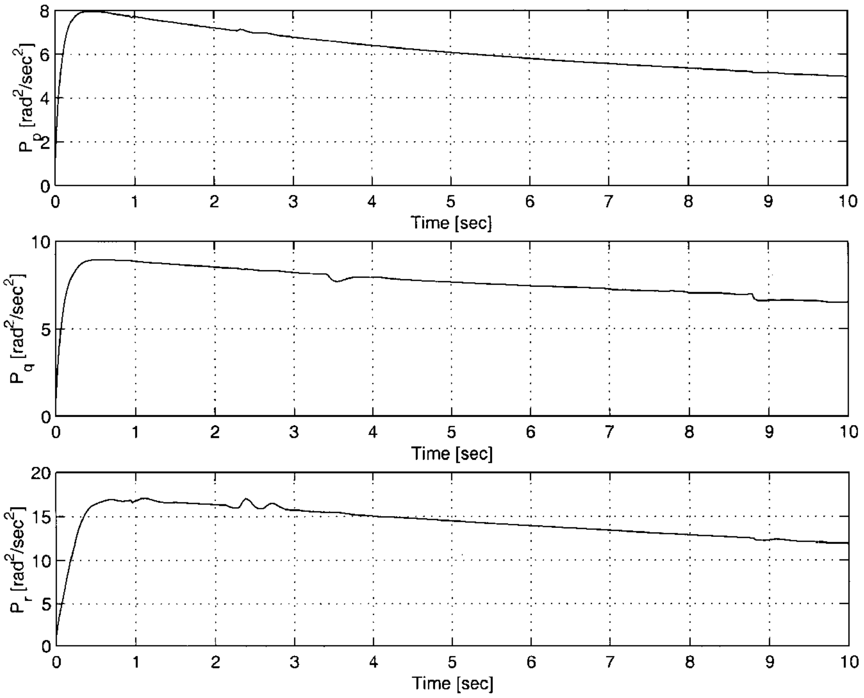

Fig. 6 Estimation error covariances of the body angular rate imply estimation convergence.

The influence of unknown gusts on the angular body rate vector is insignificant as well.

2. Sensitivity to Tracking Noise Level

When the measurement noise level changes, a proper retuning of the IEKF can reduce the estimation error covariance at the expense of a smaller convergence rate. However, because feature point velocity on the image plane was computed by taking the difference between two consecutive images, the performance of the algorithm deteriorates rapidly for increasing noise levels. In fact, simulations show that a noise level above 1.5 mrad (1σ) may cause the estimator to diverge.

D. Observability and Fusion with Additional Measurements

It was mentioned that not all 12 state variables could be estimated using a subspace-constraint estimation algorithm, which implies partial state observability. Indeed, when examining the local observability matrix, that is,

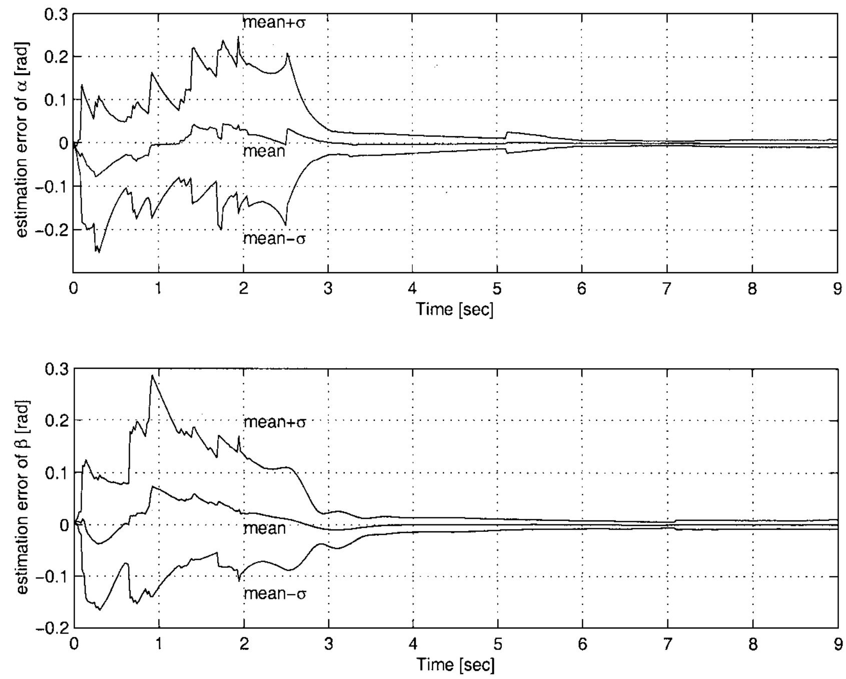

Fig. 7 Estimation errors of the angle of attack and the angle of sideslip; mean value converges to zero after 3.5 s.

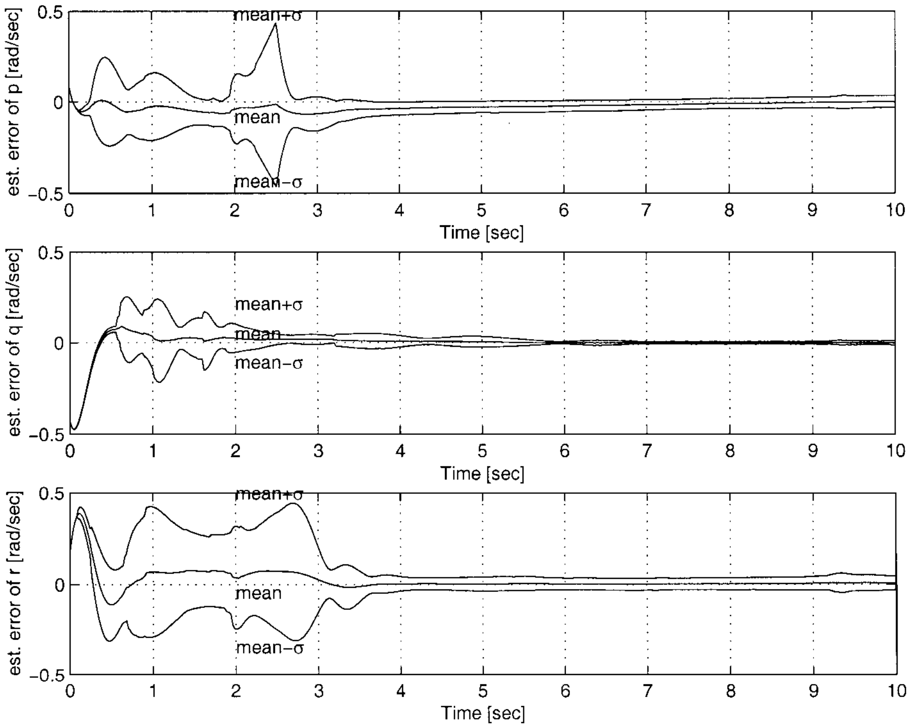

Fig. 8 Estimation errors of the body angular rates; mean values of the pitch rate and yaw rate converge to zero faster than the mean of the roll rate.

$$\mathcal{O} \triangleq \begin{bmatrix} L \\ LA \\ \vdots \\ LA^{11} \end{bmatrix} \tag{39}$$

one gets

$$\text{rank}(\mathcal{O}) = 8 < 12 \tag{40}$$

It is well known that for time-varying systems, frozen time nonobservability does not necessarily imply nonobservabilityof the original system. However, the rank deficiency shows that observability will be at best trajectory dependent. The unobservable states associated with $\mathcal{O}$ are the three Euler angles and the inertial location. This reflects a concrete physical reality: The camera senses only variations of orientation and position, due to the nature of the electro-optical tracking. Thus, inertial quantities should be impossible to obtain from merely a body-fixed electro-optical sensor.

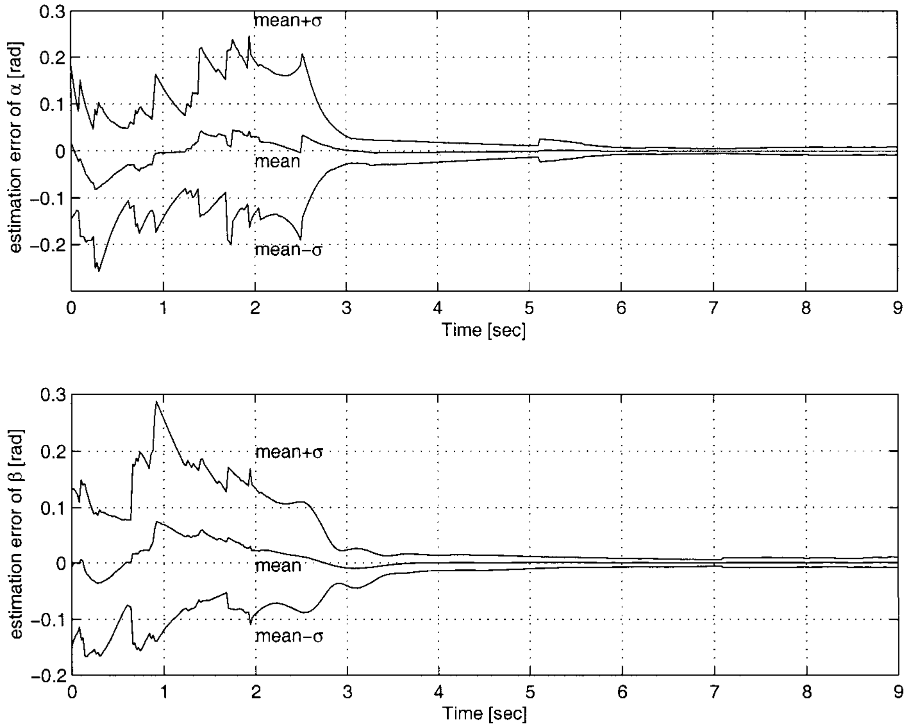

Fig. 9 Estimation errors of the angle of attack and the angle of sideslip with unknown random wind gusts; effect on the estimation is negligible.

VI. Conclusions

This paper addressed the estimation of aircraft motion based on a dynamic model for the aircraft and the information obtained from a downward-looking body-fixed camera. The basic idea is to track feature points on the image plane and use the constraints arising from a rigid-scene assumption as the measurement of an IEKF. The feasibility of the approach was verified by the performance of extensive simulations of the resulting algorithm. The most important feature of the algorithm is the ability to estimate the angle of attack, the angle of sideslip, and the angular body pitch, yaw, and roll rates. During the simulations, it was found that the estimation algorithm might diverge if measurement noise level is relatively high. For practically reasonable noise levels, however, the performance of the algorithm is satisfactory. Robustness issues were also considered. In particular, it was shown that the effect of unknown random constant Gaussian wind gusts on estimation accuracy was negligible.

Both the ambiguity problem and the frozen-time observability analysis show that a body-fixed camera cannot be used as the unique input for an autonomous navigation system. Note that establishing the opposite case was not the motivation of the present work. Rather, we believe that this paper suggests that an improvement in navigation error can be expected whenever the camera measurements are fused with an inertial measurement unit or other source of navigation measurements. This implies that a body-fixed camera and low-grade inertial instruments may provide an adequate solution for applications, which would otherwise require better, and more expensive, inertial instruments.

Appendix A: Aircraft Equations of Motion

We adopt the following six-degree-of-freedom model for the aircraft motion[17]:

1) The dynamic translatory equations in the body axes are

$$\dot u = -Q(S/m)(C_D\cos\alpha\cos\beta + C_Y\cos\alpha\sin\beta - C_L\sin\alpha) + T/m - g\sin\theta - qw + rv \tag{A1}$$

$$\dot v = -Q(S/m)\,(C_D\sin\beta - C_Y\cos\beta) + g\sin\phi\cos\theta - ru + pw \tag{A2}$$

$$\dot w = -Q(S/m)(C_D\sin\alpha\cos\beta + C_Y\sin\alpha\sin\beta + C_L\cos\alpha) - g\cos\theta\sin\phi - pv + qu \tag{A3}$$

2) The dynamic angular equations in the body axes are

$$\dot p = 1\Big/\left(I_xI_z - I_{xz}^2\right)\{I_x[QSbC_l + (I_y - I_z)qr] + I_{xz}[QSbC_n + (I_x - I_y + I_z)pq - I_{xz}qr]\} \tag{A4}$$

$$\dot q = (1/I_y)\left[QScC_m + I_{xz}(r^2 - p^2) + (I_z - I_x)rp\right] \tag{A5}$$

$$\dot r = 1\Big/\left(I_xI_z - I_{xz}^2\right)\{I_z[QSbC_n - (I_y - I_z)qr] + I_{xz}[QSbC_l + (I_y - I_z - I_z)qr - I_{xz}pq]\} \tag{A6}$$

3) The transformation of angular body rates to inertial Euler angles rates is

$$\dot\phi = p + \tan\theta(q\sin\phi + r\cos\phi) \tag{A7}$$

$$\dot\theta = q\cos\phi - r\sin\phi \tag{A8}$$

$$\dot\psi = q(\sin\phi/\cos\theta) + r(\cos\phi/\cos\theta) \tag{A9}$$

4) The transformation of ground speed from body to Earth coordinates is

$$\dot Z_E = -\dot h = -u\sin\theta + v\sin\phi\cos\theta + w\cos\phi\cos\theta \tag{A10}$$

$$\dot X_E = u\cos\theta\cos\psi + v(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi) + w(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) \tag{A11}$$

$$\dot Y_E = u\cos\theta\sin\psi + v(\sin\phi\sin\theta\sin\psi - \cos\phi\cos\psi) + w(\cos\phi\sin\theta\sin\psi + \sin\phi\cos\psi) \tag{A12}$$

5) The aerodynamic coefficients are, for the force coefficients,

$$C_L = C_{L_0} + C_{L_\alpha}\alpha + C_{L_\beta}\beta + (c/2V_a)C_{L_q}q + C_{L_{\delta_E}}\delta_e \tag{A13}$$

$$C_Y = C_{Y_\alpha}\alpha + C_{Y_\beta}\beta + (b/2V_a)\left(C_{Y_p}p + C_{Y_r}r\right) + C_{Y_{\delta a}}\delta_a + C_{Y_{\delta r}}\delta_r + C_{Y_{\delta e}}\delta_e \tag{A14}$$

$$C_D = C_{D_0} + C_k C_L^2 \tag{A15}$$

and for the moment coefficients,

$$C_m = C_{m_0} + C_{m_\alpha}\alpha + C_{m_\beta}\beta + (c/2V_a)\left(C_{m_{\dot\alpha}}\dot\alpha + C_{m_q}q\right) + C_{m_{\delta e}}\delta_e \tag{A16}$$

$$C_l = C_{l_\alpha}\alpha + C_{l_\beta}\beta + (b/2V_a)\left(C_{lp}p + C_{l_r}r\right) + C_{l_{\delta a}}\delta_a + C_{l_{\delta r}}\delta_r + C_{l_{\delta e}}\delta_e \tag{A17}$$

$$C_n = C_{n_\alpha}\alpha + C_{n_\beta}\beta + (b/2V_a)\left(C_{np}p + C_{n_r}r\right) + C_{n_{\delta a}}\delta_a + C_{n_{\delta r}}\delta_r + C_{n_{\delta e}}\delta_e \tag{A18}$$

6) The total airspeed V_a, angle of attack α, and angle of sideslip are given by

$$V_a = \sqrt{(u-u_g)^2 + (v-v_g)^2 + (w-w_g)^2} \tag{A19}$$

$$\alpha = \tan^{-1}[(w-w_g)/(u-ug)] \tag{A20}$$

$$\beta = \sin^{-1}[(v-v_g)/V_a] \tag{A21}$$

where u_g, v_g, and w_g are wing gust components along the body axes x_B, y_B, and z_B.

7) The relationship between air density ρ and altitude h is given by

$$\rho = 1.225(1 - 2.18\cdot 10^{-5}\cdot h)^{4.2586} \quad [\mathrm{kg/m^3}] \tag{A22}$$

8) Finally, the dynamic pressure is

$$Q = \tfrac{1}{2}\rho V_a^2 \tag{A23}$$

Appendix B : Linearization of the Implicit Measurement Equation

In Appendix B we discuss in detail the linearization of the implicit measurement equation, which stems from the subspace constraint. This equation is given by

$$\tilde C^\perp(\boldsymbol{\eta}, \boldsymbol{v}_C)\dot{\boldsymbol{\eta}} = \tilde{\boldsymbol{n}} \tag{B1}$$

where

$$\tilde C^\perp = (I - \tilde C\tilde C^T) \tag{B2}$$

Notice that in the case of a standard EKF, where the measurement is an explicit function of the state, a local linearization around a nominal estimated state is sufficient for filter realization. However, in case of implicit measurements, local linearization should be performed simultaneously around the estimated state and the measurement. Thus, denote

$$\zeta = \tilde C^\perp(\boldsymbol{\eta}, \boldsymbol{v}_C)\dot{\boldsymbol{\eta}} - \tilde{\boldsymbol{n}} \tag{B3}$$

and notice that

$$\zeta \approx \left.\frac{\partial \tilde C^\perp(\boldsymbol{\eta}, \boldsymbol{v}_C)\dot{\boldsymbol{\eta}}}{\partial \boldsymbol{v}_C}\right|_{\eta,\dot\eta,\hat v_C} \boldsymbol{v}_C + \left.\frac{\partial \tilde C^\perp(\boldsymbol{\eta}, \boldsymbol{v}_C)\dot{\boldsymbol{\eta}}}{\partial[\eta\ \dot\eta]}\right|_{\eta,\dot\eta,\hat v_C}\begin{bmatrix}\boldsymbol{\eta}\\ \dot{\boldsymbol{\eta}}\end{bmatrix} \triangleq \tilde L\boldsymbol{v}_C + D\begin{bmatrix}\eta\\ \dot\eta\end{bmatrix} \tag{B4}$$

or, equivalently,

$$\zeta \approx \left[\tilde L\cdot T_C^B \,\middle|\, 0_{2N\times 9}\right]\boldsymbol{x} + D\begin{bmatrix}\boldsymbol{\eta}\\ \dot{\boldsymbol{\eta}}\end{bmatrix} \triangleq L\boldsymbol{v}_C + D\begin{bmatrix}\boldsymbol{\eta}\\ \dot{\boldsymbol{\eta}}\end{bmatrix} \tag{B5}$$

Next, denote

$$R_\eta = \operatorname{cov}\begin{bmatrix}\eta\\ \dot\eta\end{bmatrix} \tag{B6}$$

to obtain the familiar formulation of an EKF linearized measurement equation.

A procedure for calculating the matrices $\tilde L$ (which immediately yields L) and D proceeds as follows. Denote by γ a scalar parameter, to be either ε_V, ε_H, u_C, v_C, or w_C, and define

$$\tilde C_\gamma \triangleq \frac{\partial \tilde C}{\partial \gamma} \tag{B7}$$

It can be shown[4] that

$$\tilde C_\gamma^\perp = -\left\lfloor \tilde C^\perp \tilde C_\gamma\, \tilde C^+ + \left(\tilde C^\perp \tilde C_\gamma\, \tilde C^+\right)^T \right\rfloor \tag{B8}$$

Now calculate $\tilde C_\gamma$ for $\gamma = \varepsilon_V, \varepsilon_H, u_C, v_C$, or w_C. To this end, recall the definition of $\tilde C$ given in Eq. (B8). This definition directly leads to

$$\tilde C_{u_C} = \left[\begin{array}{ccc|cccc} \begin{bmatrix}\varepsilon_V\\ \varepsilon_H\end{bmatrix}_1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & \ddots & 0 & \vdots & \vdots & \vdots & \vdots\\ 0 & 0 & \begin{bmatrix}\varepsilon_V\\ \varepsilon_H\end{bmatrix}_N & 0 & 0 & 0 & 0 \end{array}\right] \in \Re^{2N\times N+3} \tag{B9}$$

$$\tilde C_{v_C} = \left[\begin{array}{ccc|cccc} \begin{bmatrix}0\\ 1\end{bmatrix}_1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & \ddots & 0 & \vdots & \vdots & \vdots & \vdots\\ 0 & 0 & \begin{bmatrix}0\\ 1\end{bmatrix}_N & 0 & 0 & 0 & 0 \end{array}\right] \in \Re^{2N\times N+3} \tag{B10}$$

$$\tilde C_{w_C} = \left[\begin{array}{ccc|cccc} \begin{bmatrix}-1\\ 0\end{bmatrix}_1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & \ddots & 0 & \vdots & \vdots & \vdots & \vdots\\ 0 & 0 & \begin{bmatrix}-1\\ 0\end{bmatrix}_N & 0 & 0 & 0 & 0 \end{array}\right] \in \Re^{2N\times N+3} \tag{B11}$$

The matrix $\tilde L$ can now be obtained by calculating $\tilde C_{u_C}^\perp$, $\tilde C_{v_C}^\perp$, and $\tilde C_{w_C}^\perp$ via Eq. (B8) and stacking the results as follows:

$$\tilde L = \left[\begin{array}{c|c|c} \tilde C_{u_C}^\perp\cdot\begin{bmatrix}\begin{bmatrix}\dot\varepsilon_V\\ \dot\varepsilon_H\end{bmatrix}_1\\ \vdots\\ \begin{bmatrix}\dot\varepsilon_V\\ \dot\varepsilon_H\end{bmatrix}_N\end{bmatrix} & \tilde C_{v_C}^\perp\cdot\begin{bmatrix}\begin{bmatrix}\dot\varepsilon_V\\ \dot\varepsilon_H\end{bmatrix}_1\\ \vdots\\ \begin{bmatrix}\dot\varepsilon_V\\ \dot\varepsilon_H\end{bmatrix}_N\end{bmatrix} & \tilde C_{w_C}^\perp\cdot\begin{bmatrix}\begin{bmatrix}\dot\varepsilon_V\\ \dot\varepsilon_H\end{bmatrix}_1\\ \vdots\\ \begin{bmatrix}\dot\varepsilon_V\\ \dot\varepsilon_H\end{bmatrix}_N\end{bmatrix} \end{array}\right] \tag{B12}$$

Similarly, from Eq. (20),

$$\left(\tilde C_{\varepsilon_V}\right)_1 = \left[\begin{array}{cccc|c} \begin{bmatrix}-u_C\\ 0\end{bmatrix} & 0 & \cdots & 0 & \begin{bmatrix}0 & 2\varepsilon_V & \varepsilon_H\\ 1 & \varepsilon_H & 0\end{bmatrix}\\ 0 & 0 & \cdots & 0 & 0\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & \ldots & 0 & 0 \end{array}\right]$$

$$\left(\tilde C_{\varepsilon_H}\right)_1 = \left[\begin{array}{cccc|c} \begin{bmatrix}0\\ -u_C\end{bmatrix} & 0 & \cdots & 0 & \begin{bmatrix}-1 & 0 & \varepsilon_V\\ 0 & \varepsilon_V & 2\varepsilon_H\end{bmatrix}\\ 0 & 0 & \cdots & 0 & 0\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & \ldots & 0 & 0 \end{array}\right] \tag{B13}$$

$$
\left(\tilde{C}_{\varepsilon_V}\right)_2 = \left[\begin{array}{cccc|c}
0 & 0 & \cdots & 0 & 0 \\
0 & \begin{bmatrix} -u_C \\ 0 \end{bmatrix} & \cdots & 0 & \begin{bmatrix} 0 & 2\varepsilon_V & \varepsilon_H \\ 1 & \varepsilon_H & 0 \end{bmatrix} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \ldots & 0 & 0
\end{array}\right]
$$

$$
\left(\tilde{C}_{\varepsilon_H}\right)_2 = \left[\begin{array}{cccc|c}
0 & 0 & \cdots & 0 & 0 \\
0 & \begin{bmatrix} 0 \\ -u_C \end{bmatrix} & \cdots & 0 & \begin{bmatrix} -1 & 0 & \varepsilon_V \\ 0 & \varepsilon_V & 2\varepsilon_H \end{bmatrix} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \ldots & 0 & 0
\end{array}\right]
\tag{B14}
$$

and by this scheme

$$
\left(\tilde{C}_{\varepsilon_V}\right)_N = \left[\begin{array}{cccc|c}
0 & 0 & \cdots & 0 & 0 \\
0 & 0 & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \ldots & \begin{bmatrix} -u_C \\ 0 \end{bmatrix} & \begin{bmatrix} 0 & 2\varepsilon_V & \varepsilon_H \\ 1 & \varepsilon_H & 0 \end{bmatrix}
\end{array}\right]
$$

$$
\left(\tilde{C}_{\varepsilon_H}\right)_N = \left[\begin{array}{cccc|c}
0 & 0 & \cdots & 0 & 0 \\
0 & 0 & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \ldots & \begin{bmatrix} 0 \\ -u_C \end{bmatrix} & \begin{bmatrix} -1 & 0 & \varepsilon_V \\ 0 & \varepsilon_V & 2\varepsilon_H \end{bmatrix}
\end{array}\right]
\tag{B15}
$$

The matrices D can now be obtained by calculating $(\tilde{C}^{\perp}_{\varepsilon_V})_{1,\ldots,N}$, $(\tilde{C}^{\perp}_{\varepsilon_H})_{1,\ldots,N}$ via Eq. (B8) and stacking the results as follows:

$$
D = \left[\begin{array}{cccccc|c}
\left(\tilde{C}^{\perp}_{\varepsilon_V}\right)_1 \cdot & \begin{bmatrix} \begin{bmatrix} \dot{\varepsilon}_V \\ \dot{\varepsilon}_{\mathrm{H}} \end{bmatrix}_1 \\ \vdots \\ \begin{bmatrix} \dot{\varepsilon}_V \\ \dot{\varepsilon}_{\mathrm{H}} \end{bmatrix}_N \end{bmatrix} & \left(\tilde{C}^{\perp}_{\varepsilon_H}\right)_1 \cdot & \begin{bmatrix} \begin{bmatrix} \dot{\varepsilon}_V \\ \dot{\varepsilon}_{\mathrm{H}} \end{bmatrix}_1 \\ \vdots \\ \begin{bmatrix} \dot{\varepsilon}_V \\ \dot{\varepsilon}_{\mathrm{H}} \end{bmatrix}_N \end{bmatrix} & \cdots \left(\tilde{C}^{\perp}_{\varepsilon_H}\right)_N \cdot & \begin{bmatrix} \begin{bmatrix} \dot{\varepsilon}_V \\ \dot{\varepsilon}_{\mathrm{H}} \end{bmatrix}_1 \\ \vdots \\ \begin{bmatrix} \dot{\varepsilon}_V \\ \dot{\varepsilon}_{\mathrm{H}} \end{bmatrix}_N \end{bmatrix} & \tilde{C}^{\perp}
\end{array}\right]
\tag{B16}
$$

When the matrices L and D are evaluated during the recursive computation of the IEKF, the noisy measurements of ε_V and ε_H should be plugged in the suitable places, and the estimated velocity components, that is, $\hat{u}_C$, $\hat{v}_C$, and $\hat{w}_C$, should be used where appropriate.

Appendix C: Numerical Data

The simulation was carried out assuming a model of a general aviation airplane called Navion.[20] The values of the various aerodynamic coefficients are given in Eq. (C1), and the configuration data are described in Eq. (C2). The fin deflections constitute inputs to the three maneuver channels and are given in Eq. (C3). Finally, Eq. (C4) gives the initial conditions chosen for the simulation.

1) The aerodynamic coefficients (per radian) are

$$
\begin{gathered}
C_{L0} = 0, \qquad C_{L\alpha} = 4.44, \qquad C_{L\beta} = 0, \qquad C_{Lq} = 3.8, \\
C_{L_{\delta e}} = 0.355, \qquad C_{m0} = 0, \qquad C_{m\alpha} = -0.683, \\
C_{m\beta} = 0, \qquad C_{m\dot{\alpha}} = -4.36, \qquad C_{mq} = -9.96, \\
C_{m_{\delta e}} = -0.923, \qquad C_{Y\alpha} = 0, \qquad C_{Y\beta} = -0.564, \\
C_{Yp} = 0, \qquad C_{Yr} = 0, \qquad C_{Y_{\delta a}} = 0, \qquad C_{Y_{\delta r}} = 0.157, \\
C_{Y_{\delta e}} = 0, \qquad C_{l\alpha} = 0, \qquad C_{l\beta} = -0.074,
\end{gathered}
$$

$$
\begin{gathered}
C_{lp} = -0.41, \qquad C_{lr} = 0.107, \qquad C_{l_{\delta a}} = -0.134, \\
C_{l_{\delta r}} = 0.107, \qquad C_{l_{\delta e}} = 0, \qquad C_{n\alpha} = 0, \\
C_{n\beta} = 0.071, \qquad C_{np} = -0.0575, \qquad C_{nr} = -0.125, \\
C_{n_{\delta a}} = -0.0035, \qquad C_{n_{\delta r}} = -0.072, \qquad C_{n_{\delta e}} = 0 \\
C_{D0} = 0.05, \qquad C_K = 0.33
\end{gathered}
\tag{C1}
$$

2) The configuration data are

$$
\begin{gathered}
S = 20\ \mathrm{m}^2, \qquad c = 1.9\ \mathrm{m}, \qquad T = 15,000\ \mathrm{Nt} \\
m = 1245.8\ \mathrm{kg}, \qquad I_x = 1420.9\ \mathrm{kgm}^2, \\
I_y = 2.86 I_{xx}, \qquad I_z = 3.36 I_{xx}
\end{gathered}
\tag{C2}
$$

3) The fin deflections are

$$
\delta_e = 1\ \mathrm{deg}, \qquad \delta_a = 0, \qquad \delta_r = 1\ \mathrm{deg}
\tag{C3}
$$

4) The initial conditions are

$$
\begin{gathered}
u_0 = 100\ \mathrm{m/s}, \qquad v_0 = 10\ \mathrm{m/s}, \qquad w_0 = 10\ \mathrm{m/s} \\
p_0 = 10\ \mathrm{deg/s}, \qquad q_0 = -30\ \mathrm{deg/s}, \qquad r_0 = 10\ \mathrm{deg/s} \\
\phi_0 = \theta_0 = \psi_0 = 0\ \mathrm{deg}, \qquad h_0 = 1000\ \mathrm{m}, \qquad X_{E0} = Y_{E0} = 0
\end{gathered}
\tag{C4}
$$

References

[1] Greenway, P., "Recovery of Motion Parameters Using Optical Flow," *Transactions of SPIE, Mobile Robots II*, Vol. 852, 1987, pp. 214–225.

[2] Soatto, S., Frezza, R., and Perona, P., "Motion Estimation via Dynamic Vision," *IEEE Transactions on Automatic Control*, Vol. 41, No. 3, 1996, pp. 393–412.

[3] Weng, J., Narendra, A., and Huang, T. S., "Optimal Motion and Structure Estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 9, 1993, pp. 864–883.

[4] Soatto, S., and Perona, P., "Recursive 3-D Visual Motion Estimation Using Subspace Constraints," *International Journal of Computer Vision*, Vol. 22, No. 3, 1997, pp. 235–259.

[5] Dickmanns, E. D., and Christians, T., "Relative 3-D State Estimation for Autonomous Visual Guidance of Road Vehicles," *Proceedings of Intelligent Autonomous Systems*, Amsterdam, 1989.

[6] Merhav, S. J., and Bresler, Y., "On-Line Vehicle Motion Estimation from Visual Terrain Information Part 1: Recursive Image Registration," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-22, No. 5, 1986, pp. 583–587.

[7] Merhav, S. J., and Bresler, Y., "On-Line Vehicle Motion Estimation from Visual Terrain Information Part 2: Ground Velocity and Position Estimation," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-22, No. 5, 1986, pp. 588–604.

[8] Chaumette, F., and Santos, A., "Tracking a Moving Object by Visual Serving," *Proceedings of the 12th IFAC World Congress*, Vol. 9, 1993, pp. 409–414.

[9] Curwen, R., Blake, A., and Zisserman, A., "Real-Time Visual Tracking for Surveillance and Path-Planning," *Proceedings of the ECCV*, 1992.

[10] Adam, A., Rivlin, E., and Rotstein, H., "Fusion of Fixation and Odometry for Vehicle Navigation," *IEEE Transaction on Systems, Man, and Cybernetics* (to be published).

[11] Faugeras, O. D., *Three Dimensional Vision, A Geometric Viewpoint*, MIT Press, Cambridge, MA, 1993, pp. 226–262.

[12] Grunwald, A. J., and Kohn, S., "Flight-Path Estimation in Passive Low-Altitude Flight by Visual Cues," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 2, 1993, pp. 363–370.

[13] Koifman, K., and Merhav, S. J., "Autonomously Aided Strapdown Attitude Reference Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 6, 1991, pp. 1164–1168.

[14] Durham, W. C., Lutze, F. H., Barlas, R. M., and Munro, B. C., "Nonlinear Model-Following Control Application to Airplane Control," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 3, 1994, pp. 570–577.

[15] Robinett, R. D., and Eisler, R. G., "Trajectory Matching Flight-Path Optimization of Aerospace Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 5, 1997, pp. 903–908.

[16] Iliff, K. W., "Parameter Estimation for Flight Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 12, No. 5, 1989, pp. 602–622.

[17] Merhav, S. J., *Aerospace Sensor Systems and Applications*, McGraw–Hill, New York, 1992, pp. 372–380.

[18] Soatto, S., "3-D Structure from Visual Motion: Modeling, Representation and Observability," *Automatica*, Vol. 33, No. 7, 1997, pp. 1287–1312.

[19] Gelb, A., *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1974, pp. 102–142.

[20] Nelson, R. C., *Flight Stability and Automatic Control*, McGraw–Hill, New York, 1989, pp. 70–85.

[21] Etkin, B., *Dynamics of Atmospheric Flight*, Wiley, New York, 1972, pp. 122–142.