Image-Based Pointing and Tracking for Inertially Stabilized Airborne Camera Platform

Zdeněk Hurák, *Member, IEEE*, and Martin Řezáč

***Abstract*—This paper describes a novel image-based pointing-tracking feedback control scheme for an inertially stabilized double-gimbal airborne camera platform combined with a computer vision system. The key idea is to enhance the intuitive decoupled controller structure with measurements of the camera inertial angular rate around its optical axis. The resulting controller can also compensate for the apparent translation between the camera and the observed object, but then the velocity of this mutual translation must be measured or estimated. Even though the proposed controller is more robust against longer sampling periods of the computer-vision system then the decoupled controller, a sketch of a simple compensation of this delay is also given. Numerical simulations are accompanied by laboratory experiments with a real benchmark system.**

***Index Terms*—Image-based visual servoing, inertial stabilization, line-of-sight stabilization.**

I. Introduction

A. Inertial Line-of-Sight Stabilization on Mobile Carriers

THE very basic control task for steerable cameras or antennas mounted on mobile carriers such as trucks, unmanned aircraft or ships, is to keep the commanded line of sight (optical axis) still even in presence of various disturbing phenomena like mass disbalance, aerodynamic (or wind-induced) torque and possible kinematic coupling between gimbal axes, see Fig. 1. Motivated also by defence technological needs, the topic of inertial stabilization was studied extensively in the past few decades. Several relevant papers from 1970s through 1990s were archived in the selection [1]. Dedication of a full issue of *IEEE Control System Magazine* (February 2008) featuring nice survey papers [2]–[4] confirms that the topic is still relevant for the engineering community. Another recent issue of the same journal brings a rigorous analysis of control problems related to a standard double gimbal system [5], though it is not directly applicable to inertial stabilization.

Manuscript received January 31, 2011; revised May 11, 2011; accepted July 06, 2011. Manuscript received in final form August 05, 2011. Date of publication September 12, 2011; date of current version June 28, 2012. Recommended by Associate Editor F. Caccavale. This work was supported in part by the Ministry of Education of the Czech Republic within the Centre for Applied Cybernetics (1M0567) and by the Ministry of Industry and Trade of the Czech Republic within the Project TIP FR-TI1/265.

The authors are with the Faculty of Electrical Engineering, Czech Technical University, Prague 16627, Czech Republic (e-mail: hurak@fel.cvut.cz; rezac.martin@fel.cvut.cz).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCST.2011.2164541

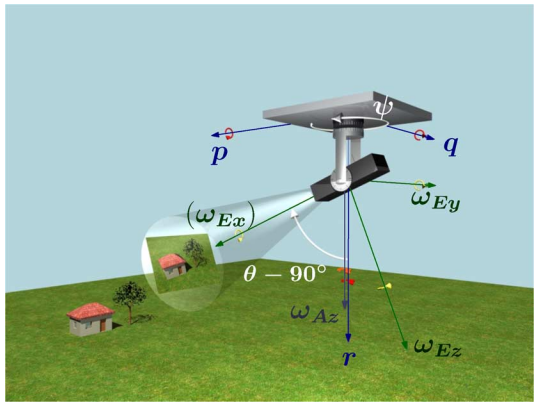

Fig. 1. Basic scenario for inertial line-of-sight stabilization. Depicted in green are the components $\omega_{Ex}, \omega_{Ey}, \omega_{Ez}$ of the vector of inertial angular rate of the elevation frame (as measured by MEMS gyros attached to the camera), blue vectors p, q, r denote the rate components of the base (UAV here). The ω_{Az} component is attached to the outer gimbal (the other two components are not shown). Two white arcs denote the two relative angles. The origins of the coordinate frames are assumed to coincide in the computations.

B. Automatic Visual Tracking on Mobile Carriers

All of the above cited works (including the references made therein) mostly focus on the task of inertial stabilization only. The issue of extending the inertial rate stabilizing feedback loop to visual tracking system is only dealt with at a rather simplistic level in [2] by suggesting the common cascaded control structure for every rotational degree of freedom: a single-input-single-output inner (inertial rate stabilization) loop is accepting commands from the output of the corresponding outer (visual tracking) loop. There are some pitfalls hidden in this decoupled approach, though. This paper will describe the troubles that are encountered when using the classical double-gimbal platform and offer a solution. To the best of the authors knowledge, this is the first formal treatment of visual pointing and tracking for inertially stabilized camera systems. Preliminary versions of this paper were presented at [6] and [7]; this paper includes corrections and some minor theoretical extensions but the most important extension is in supporting the theoretical analysis by reporting on laboratory experiments with a realistic benchmark platform.

C. Benchmark System

The configuration considered in this study is the common two-degree-of-freedom configuration: double gimbal system. The inner gimbal allows for elevation of the payload, the outer gimbal allows for a change in heading (or azimuth) angle. A benchmark system was designed and built within a project coordinated by Czech Air Force and Air Defence Technological

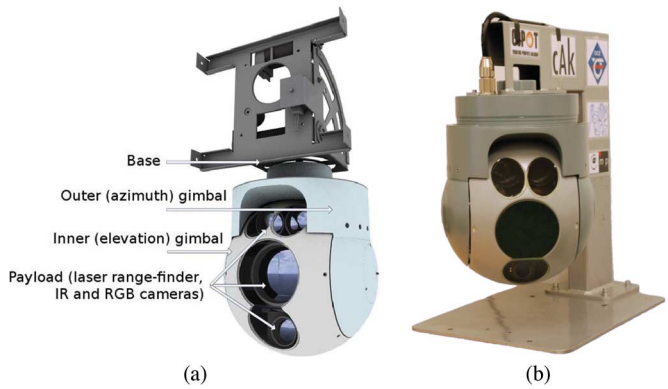

Fig. 2. Platform developed by Czech Air Force and Air Defense Technological Institute in collaboration with Czech Technical University in Prague and Essa company. (a) 3-D visualization of the platform: both the azimuth and the elevation angles can rotate $n \times 360°$. (b) Photo of the benchmark system. The supporting structure is only used in a lab.

Institute (Vojenský ústav letectva a PVO) in collaboration with Czech Technical University in Prague and ESSA company. The payload consists of a regular RGB camera, infrared camera and laser range-finder, see Fig. 2. Direct drive motors are used for the two axes and MEMS-based gyros (inertial rate sensors) are attached to the payload.

D. Notation for Coordinate Frames and Their Rotations

The paper relies on expressing rotations of coordinate frames with respect to some other coordinate frames. The right-handed orthogonal coordinate frames are represented by triads of vectors $\{x, y, z\}$ and for simplicity they all assume a common origin; this is certainly justifiable when faraway objects are tracked. The coordinate frames and their symbols used in subscripts and superscripts are: the reference coordinate frame $[R]$ aligned with the ground but translated to the center of gravity of the carrier, its z_R axis oriented towards the ground as is common in aerospace applications; the base coordinate frame fixed to the body of the carrier $[B]$ with its x_B axis heading forward and y_B to the starboard; the coordinate frame attached to the outer (azimuth) gimbal $[A]$, which can rotate with respect to the carrier around the $z_B = z_A$ axis; the coordinate frame attached to the inner (elevation) gimbal $[E]$, which can rotate with respect to the azimuth gimbal around the $y_A = y_E$ axis; and finally the coordinate frame attached to the camera $[C]$. Rotation of $[C]$ with respect to $[E]$ is fixed and is used just for the "esthetic" purpose of (re)denoting the camera optical axis as the z_C axis.

The sequence of the two key rotations expressing the pose of the inner gimbal (fixed to camera) with respect to the base (carrier) is visualized in Fig. 3 and for completeness it is given by

$$R_A^B = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

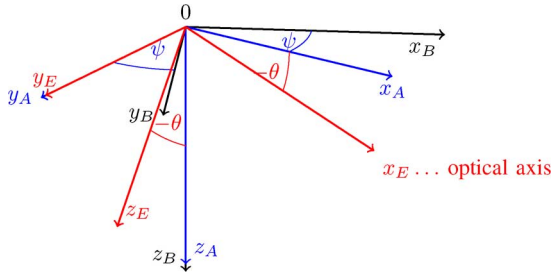

Fig. 3. Composition of rotation of coordinate frames attached to the carrier (base), outer (azimuth) gimbal, and inner (elevation) gimbal. The $[C]$ frame attached to the camera not visualized here, see Fig. 5.

and

$$R_E^A = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{2}$$

where the lower and upper indices are used here as "rotation matrix expressing the coordinate triade of the A frame within the B frame". Applying the right-hand rule, the (outer) azimuth gimbal rotates to right for the positive angle ψ and the (inner) elevation gimbal rotates up for a positive increment in the θ angle. Using the common shorthand notation like $c_\psi = \cos\psi$, the composition of the two rotations is given by the matrix product

$$R_E^B = \begin{bmatrix} c_\psi c_\theta & -s_\psi & -c_\psi s_\theta \\ s_\psi c_\theta & c_\psi & -s_\psi s_\theta \\ -s_\theta & 0 & -c_\theta \end{bmatrix}. \tag{3}$$

Finally, the fixed rotation between the camera frame and the elevation frame is given by

$$R_C^E = \begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}. \tag{4}$$

When specifying angular rates, the subscript/superscript scheme used here follows the common style, defined for instance in [8]: one needs to tell which coordinate frame is rotating with respect to which other coordinate frame, and in which coordinate frame is such a vector expressed. For example, $\omega_{A,E}^R$ stands for the angular rate of the Elevation gimbal with respect to the Azimuth gimbal, expressed in the Reference frame. Oftentimes, the notation is relaxed in the paper to avoid cluttering the formulas with indices. For instance, ω_A is a short notation for $\omega_{R,A}^A$, that is, the inertial angular rate of the A gimbal. Its z component is then ω_{Az}.

II. Inertial Angular Rate Stabilization

In order to make the line of sight insensitive to external disturbances, a simple controller structure can be used. Two decoupled single-input–single-output (SISO) inertial rate controllers suffice, one for each measured (component of the) inertial angular rate. Namely, the following:

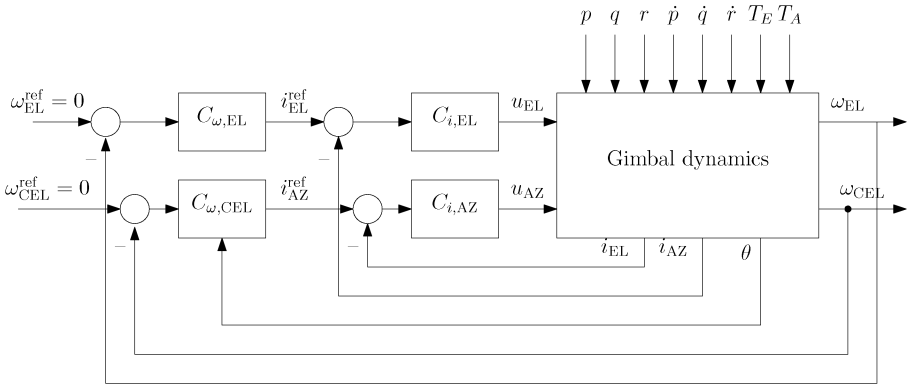

Fig. 4. Inertial stabilization. Two independent (decoupled) SISO feedback loops, one for each rate gyro attached to the body of camera, $\omega_{\rm EL}(=\omega_{Ey})$ and $\omega_{\rm CEL}(=\omega_{Ez})$. The cross-elevation stabilizing controller must contain secant gain correction $1/cos(\theta)$. The disturbing variables are the carrier roll, pitch, and yaw rates, p, q, r, respectively, their derivatives and external torques around the two motor axes. The innermost current loops are also depicted.

- the inertial angular rate ω_{Ey} (also denoted with the mnemotechnic $\omega_{\rm EL}$) of the payload about the axis of the elevation motor (camera elevation rate);
- and the inertial angular rate ω_{Ez} of the payload around its own vertical axis, also nicknamed camera cross-elevation rate (and denoted $\omega_{\rm CEL}$) since its axis is always orthogonal to the $\omega_{\rm EL}$ axis.

This is visualized in Fig. 1. The resulting decoupled controller configuration is in Fig. 4. It is clear from Fig. 1 that the cross-elevation controller must include a secant gain correction $1/cos(\theta)$, because the motor in the azimuth gimbal cannot directly affect $\omega_{\rm CEL}(=\omega_{Ez})$. It can only do so indirectly through ω_{Az}. It is only when the camera is pointing to the horizon, that is, when $\theta = 0$, that $\omega_{Az} = \omega_{\rm CEL}(=\omega_{Ez})$. See [3] for details.

Even though there is some gyroscopic coupling between the two axes (see [9], [10] for full models or [5] for the simplified version when the base is still), its influence is not worth designing a multiple-input–multiple-output (MIMO) rate controller. This neglected gyroscopic effect can be cast as yet another external disturbing torque and as such left to the rate controller to suppress.

III. Image Tracker Description

This section gives some details on the automatic image tracker, even though for the purpose of this paper it can be viewed as a black-box device. It serves the purpose of a "relative displacement sensor" and its communication with the rate loop is unidirectional.[1]

The image tracker is an algorithm (implemented in a software routine) which recognizes the target object in the input image frame sequence and returns its coordinates in every image frame of the acquired sequence. The target is usually selected by a human operator, who marks the target in the first image frame. The control system then steers the camera in order to get the (image of the) target into the center of the field of view (center of the image frame) and hold it there.

It appears that image tracking of a ground object for the purpose of airborne surveillance is one of the most difficult image tracking tasks [11], [12]. The following are the reasons.

- Weak visual differences between the tracked object and the background.
- Many localization results assume that the background is static with respect to the camera, which is certainly not the case for cameras carried by aircraft.
- The image of the target object is usually very small. Tracker useful for airborne surveillance must work with objects which projects into images as patches smaller than 10×10 pixels.
- The relative movement of the image of the target can be very fast (multiples of object's size from frame to frame) mainly due to rotation of camera, either intentional or unwanted.

There are many algorithms for image tracking in different conditions and tasks. The following families of algorithms can be considered.

- *Pattern matching algorithms* compare the captured image of the target with its model. Error function is defined that measures the similarity between the model (represented by an image patch) and the actual image in a specified position. The algorithm searches for a position where the error is at minimum. The representative of this class of algorithms are sum of square differences (SSD) [13] and the basic Kanade-Lucas (K-L) algorithm [14].
- *Feature tracking algorithms* are based on extraction of a small number features (important points) from the image frame and their tracking. The features should satisfactorily describe the object and be easily recognized in next image frame. There are many ways how to define and track the features. Mean-shift algorithm is based on a density analysis of feature space [15], but some more sophisticated algorithm were developed, the examples of which are Kanade-Lucas-Tomasi tracker (KLT) [16], scale invariant feature transform (SIFT) [17], maximally stable extremal regions (MSER) [18] and [19].
- *Object recognition trackers* aim at recognizing an object in the image frame. Machine-learning techniques (for example [20]) are applied on features which are extracted in a preprocessing stage. Target object is recognized and localized in each image. The learning stage of the procedure could be run offline, but then only a "learned" class

[1]So far the image tracker used in the project takes no advantage of availability of the inertial rate measurements.

of objects could be tracked. Online learning turns out more useful for the present application. An example of this class of algorithms is online boost [21].

These algorithms differ in tracking capabilities when it comes to different types of objects, required conditions, machine time consumption and many other operational characteristics. The experiments conducted with the benchmark system relied on the SSD algorithm. It is easy to implement and is capable of tracking small objects and recognizing similar objects. A major disadvantage of the SSD algorithm is its high machine time consumption.

From the control system viewpoint, the major complication that the image tracker brings into the feedback loop is the transportation delay. The delay consists of computational time for the actual tracker algorithm and the time for image capturing by the hardware. In the benchmark system, a standard PAL video camera with analog output was used, for which 5–20 ms were needed for image capturing. Then it takes another 40 ms (25 Hz frequency) to transmit an analog video signal from the camera. The analog video signal is then captured by a video grabber and after some 10 ms is transferred to the computer. Only then the tracking algorithm could be started. Therefore the minimum transportation delay is around 60 ms plus the computing time of tracker algorithm. It is only the latter that could be minimized by implementing more efficient algorithms.

IV. Modeling the Dynamics for Pointing and Tracking

Before starting discussions on ways to design and implement a feedback controller for the task of pointing and tracking, a model must be developed. At the initial treatment, the inertial angular rate (feedback) loops can be regarded as perfect within the appropriate frequency range and saturation bounds, that is, the commanded inertial angular rates $\omega_{\mathrm{EL}}^{\mathrm{ref}}$ and $\omega_{\mathrm{CEL}}^{\mathrm{ref}}$ can be regarded as perfectly followed by the inner loops. To develop a mathematical model for this idealized situation, a few basic concepts from the established domain of visual servoing will be given. The next few paragraphs are fully based on the two chapters from [8] dedicated to computer vision and vision-based control. They are given here just for a convenience of a reader nonacquainted with these concepts. Another comprehensive introductory material is [22].

A. Perspective Projection

The objects to be observed are located in the full 3-D world while the camera can only record their 2-D image. The coordinates of the object in the world (on the ground) expressed in the camera frame are given by $P = [x, y, z]^T$. Simplifying a bit the model of the optics, we make the so-called pinhole assumption, which defines the image coordinate frame as follows. At a focal distance λ from the origin of the camera coordinate frame, consider the image plane orthogonal to the optical axis of the camera. The coordinates of the point of intersection of the line connecting the object with the origin are $p = [u, w, \lambda]^T$. The vector $s = [u, w]^T$ thus gives the image coordinates. All this is visualized in Fig. 5. Thanks to the pinhole assumption

$$k[x \quad y \quad z]^T = [u \quad w \quad \lambda]^T \tag{5}$$

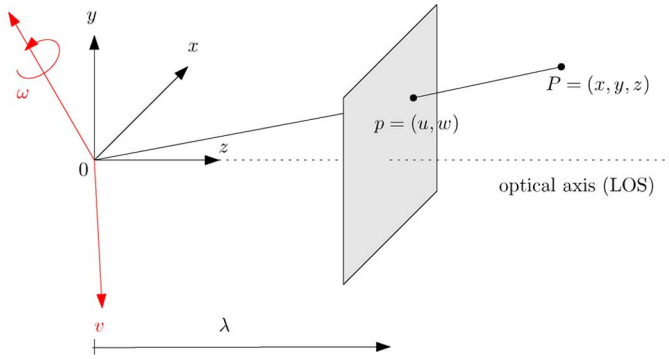

Fig. 5. Coordinates of the object on the ground expressed in the coordinate frame attached to the camera and (after projection) in the image plane. Rotation $\omega_{R,C}^{C}$ and translation v_C of the camera frame within with respect to the inertial frame is also illustrated (redrawn from [8]).

we have that

$$u = \lambda \frac{x}{z}, \quad w = \lambda \frac{y}{z}. \tag{6}$$

To make this story complete, the coordinates in the image plane should then be quantized and the origin should be moved to the lower left corner to obtain pixel coordinates $[r, c]^T$

$$-\frac{u}{s_x} = (r - o_r), \quad -\frac{w}{s_y} = (c - o_c) \tag{7}$$

where s_x and s_y are the pixel dimensions and o_r and o_c are half the width and height of the image frame in pixels.

Nonetheless, for the analysis in this paper we will stick to u and w variables to make the formulas less involved. In simulations and experiments presented here, the "pixelized" information will be considered centered (again, in the name of simplicity). That is, we will use

$$-\frac{u}{s_x} = r, \quad -\frac{w}{s_y} = c. \tag{8}$$

The computer vision system that processes the images captured by the camera can surely perform this centering before sending the data to the pointing-tracking controller.

B. Camera Motion and the Interaction Matrix

This subsection is again extracted from the nice introduction to image-based visual servoing in the textbook [8]. Consider the movement of the camera in the inertial space characterized by its linear and rotational velocities $v_C = [v_{Cx}, v_{Cy}, v_{Cz}]^T$ and $\omega_C = [\omega_{Cx}, \omega_{Cy}, \omega_{Cz}]^T$, both expressed in the camera frame. Stack them together to form a time-dependent vector $\xi(t) = [v_C(t), \omega_C(t)]^T \in \mathbb{R}^6$. To be rigorous, we should write $\omega_{R,C}^{C}$ to emphasize that it is an angular rate of the camera frame with respect to the reference (inertial) frame, expressed in the camera frame, and similarly $v_{o_C}^{C}$ to emphasize that it is a translational velocity of the origin o_C of the camera coordinate frame with respect to the inertial frame, also expressed in the camera frame. But this would yield the equations illegible.

The motion of the object as viewed by the camera is described by the so-called image feature velocity $\dot{s}(t)$, which can be obtained as a derivative of the image feature vector (in the simplest case it is just a position of some significant point). The nice thing

is that it is possible to relate ξ and $\dot{s}$ by a transform resembling the concept of Jacobian and denoted often an interaction matrix or image Jacobian

$$\dot{s}(t) = L(s, z, \lambda)\xi(t). \tag{9}$$

Next we consider the simplest case of a single-point feature and assume that the ground object does not move. Extension of the results stated here to the case of a moving ground target is feasible, but the resulting interaction matrix will be a function of the velocity of the ground object, which is unknown to the inertial stabilization system (but it may be worth exploring if at least rough estimate of the object velocities can be used). This matrix is derived in [8, p. 415, eq. (12.14)] as

$$\begin{bmatrix} \dot{u} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} -\frac{\lambda}{z} & 0 & \frac{u}{z} & \frac{uw}{\lambda} & -\frac{\lambda^2+u^2}{\lambda} & w \\ 0 & -\frac{\lambda}{z} & \frac{w}{z} & \frac{\lambda^2+w^2}{\lambda} & -\frac{uw}{\lambda} & -u \end{bmatrix} \begin{bmatrix} v_{Cx} \\ v_{Cy} \\ v_{Cz} \\ \omega_{Cx} \\ \omega_{Cy} \\ \omega_{Cz} \end{bmatrix}. \tag{10}$$

The procedure for the derivation is straightforward: first, express the position of a fixed (not moving) point on the ground in the coordinate frame of the moving (rotating and translating) camera, and then project these new coordinates to the image plane. (It is vital to keep in mind within which coordinate frame the velocity vectors are being expressed. This is quite tedious. In this case, both the translational and rotational velocities are indeed considered with respect to the inertial reference frame but are expressed in the camera frame).

It appears useful to highlight the structure in the interaction matrix by writing it as a composition of two parts

$$\dot{s} = L_v(u, w, z)v_C + L_\omega(u, w)\omega_C \tag{11}$$

because it turns out that only the part corresponding to the translation of the camera coordinate frame depends on the image depth (distance to the observed ground object) z. The rotational part is independent of z. The focal length λ is regarded as a fixed parameter.

The three components ω_{Cx}, ω_{Cy} and ω_{Cz} define the inertial angular rate vector $\omega_C = [\omega_{Cx}, \omega_{Cy}, \omega_{Cz}]^{\mathrm{T}}$ in the camera coordinate frame $[C]$, which is rotated with respect to the elevation gimbal frame $[E]$ using a fixed (constant) rotation matrix R_C^E

$$\omega_E = R_C^E \omega_C = \begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \omega_C. \tag{12}$$

While by using the two direct drive motors it is possible, at least partially, to affect the vector ω_E by commanding its two components $\omega_{Ey}(= \omega_{\mathrm{EL}})$ and $\omega_{Ez}(= \omega_{\mathrm{CEL}})$, it is rather unlikely that the translational velocity v_C will be commanded by the autopilot based on the needs of the pointing-tracking algorithm. (But some projects might allow it.)

Therefore, in order to develop some insight into the model, forget v_C for a moment (assume $v_C = 0$ temporarily, it can be treated as a disturbance later, either estimated or not). Using the transformation (12) and the rotation part of the interaction matrix (10) we get

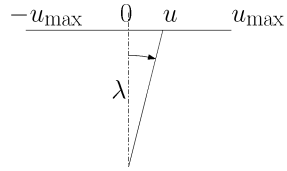

Fig. 6. Relationship between the image coordinate system and the corresponding angle.

$$\begin{bmatrix} \dot{u} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} -\frac{uw}{\lambda} & \frac{\lambda^2+u^2}{\lambda} - w\tan\theta \\ \frac{\lambda^2+w^2}{\lambda} & \frac{uw}{\lambda} + u\tan\theta \end{bmatrix} \begin{bmatrix} \omega_{\mathrm{EL}}^{\mathrm{ref}} \\ \omega_{\mathrm{CEL}}^{\mathrm{ref}} \end{bmatrix} \tag{13}$$

and the camera tilt angle θ evolves according to

$$\dot{\theta}(t) = \omega_{\mathrm{EL}}^{\mathrm{ref}}(t). \tag{14}$$

C. Linearization at Distinguished Operating Points

In order to develop an insight into the model (13), consider the situation when $\theta = 0$ (a wing-level flight and the camera pointing towards the horizon) and $w = 0$ (the observed object vertically centered on the screen). The dynamics is then constrained to one dimension and the equation simplifies to

$$\dot{u} = \frac{\lambda^2 + u^2}{\lambda} \omega_{\mathrm{CEL}}^{\mathrm{ref}}. \tag{15}$$

The term $(\lambda^2 + u^2)/\lambda$ expresses the nonlinear relationship between the angle and the line segment in the image plane. This is illustrated in Fig. 6.

The focal length λ for the system used by the authors ranges in [4.2, 42] mm. The width of the CCD camera chip is 3.2 mm. Hence, for the maximum zoom, the nonlinear term can be approximated by λ even for u approaching the maximum value, that is, the observed objects is initially located near the borders of the field of view (and the control goal is to bring it to the center). The linear dynamics is then

$$\dot{u} = \lambda \omega_{\mathrm{CEL}}^{\mathrm{ref}} \tag{16}$$

that is, the model of dynamics is represented by a pure integrator with a gain λ (given by the optics). For shorter focal lengths (approaching the lower limit of 4.2 mm) this approximation is only valid for correspondingly smaller u, that is, for tracking purposes only, not for (re)pointing over a large part of the image plane.

D. Analysis of Achievable Bandwidth for Pointing and Tracking

The computer vision system works at discrete time instants with the sampling period T_s ranging between something like 0.1 s and 2 s (depending on complexity and performance of the algorithm), which is relatively long compared to 250 Hz of the inner inertial rate loop. This introduces a total delay τ of about $1.5T_s$ into the feedback loop.

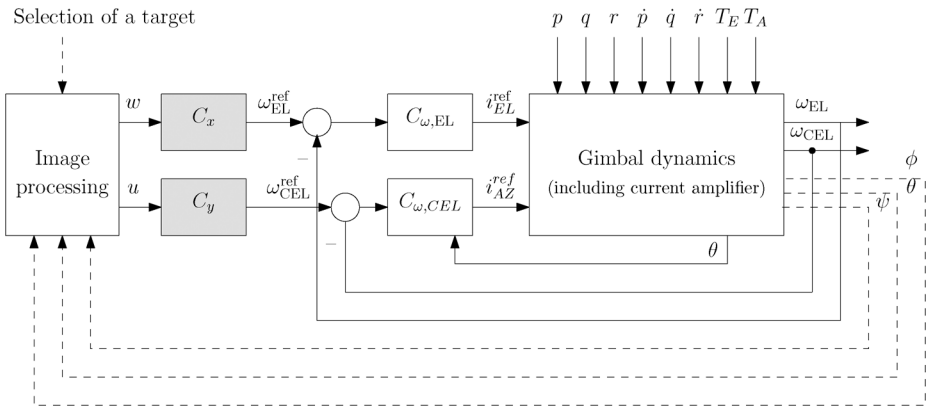

Fig. 7. Naive pointing-tracking system formed by two SISO loops closed around two inertial rate stabilization loops. The dashed lines are not signals truly fed back to the pointing-tracking controller. These are variables representing orientation of the camera which affects the position and orientation of objects in the image plane.

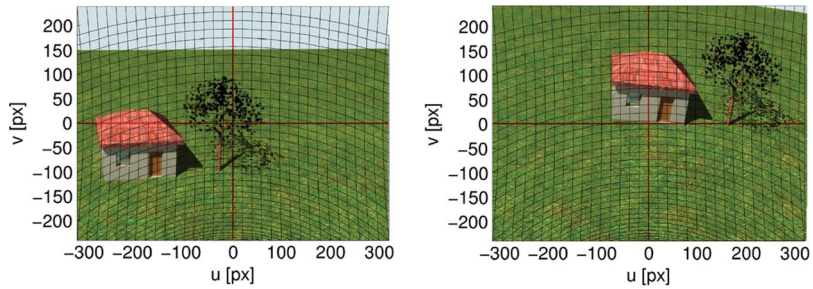

Fig. 8. Illustration of how in an attempt to steer the camera such that the image of the roof of the house gets back to the middle of the field of view using azimuth motor only, the introduced rotation of the camera around its optical axis makes the horizontal movement curved. Consequently, correction in vertical direction using the elevation motor is needed. Curvilinear coordinate system in the image plane corresponds to the initial elevation of camera by $\theta = -54^\circ$ with respect to the body of the aircraft.

It is known that the achievable bandwidth is limited by several properties of the system, delay being one of them. With the sampling period of the image tracker set to $T_s = 0.5$ s, the achievable bandwidth is approximately limited by

$$\omega_{\mathrm{BW}} < \frac{1}{\tau} = \frac{1}{1.5T_s} = 1.3 \text{ rad/s} = 0.2 \text{ Hz}. \tag{17}$$

It is derived in [23] from ideal closed-loop transfer functions achievable for systems with a delay τ. Ideally, $T(s) = 1e^{-\tau s}$, therefore $S(s) = 1 - e^{-\tau s}$. By Taylor series expansion $S(s) \approx \tau s$. Therefore $|S(j\omega)|$ crosses 0 dB at about $1/\tau$.

This suggests that the fastest possible pointing-tracking loop will work up to a fraction of 1 Hz if the information from the image tracker is provided twice per second and is delayed one sample period. This roughly corresponds to the classical rule-of-thumb rules [24] for selection of a sampling rate for undelayed systems as 10 to 20 times the closed-loop bandwidth.

V. Decoupled Pointing and Tracking

Proceeding one step further beyond the mere inertial stabilization, the question of the most suitable feedback control configuration for automatic visual tracking pops up. Shall we use the immediate extension which closes a SISO tracking loop around the corresponding SISO inertial rate loop?

The cascade approach is justified: whereas the inner (inertial rate) loop aims to attenuate the disturbances at middle and high frequencies, the outer (pointing) loop should be active at low frequencies. This straightforward but naive solution is in Fig. 7.

Insisting on decoupled controllers is plausible from an implementation viewpoint. There is a trick hidden here, though, as seen in Fig. 8. When the automatic computer vision tracker detects a regulation error in the horizontal direction in the image plane while seeing no error in vertical direction, the simple cascaded structure of Fig. 7 would command the azimuth motor only. This motor alone, however, cannot create a purely horizontal motion in the image plane when $\theta \neq 0$. A geometric explanation can be found in Fig. 1: to steer the camera such that the image of an object moves horizontally in an image plane, one would need to command the cross-elevation inertial rate ω_{CEL} (denoted as ω_{Ez} in the figure). However, the motor can only affect the component of the inertial rate in the direction of the azimuth motor axis, that is, ω_{Az}. As soon as there is some misalignment between the two, that is, when the camera is tilted up or down to the ground while the aircraft is in level flight ($\theta \neq 0$), the vector oriented in the azimuth motor axis of length ω_{Az} has some nonzero projection ω_{Ex} to the camera optical axis. Consequently, some unwanted rotation of the image as well as vertical displacement are introduced. Curvilinear coordinate mesh in Fig. 8 is generated by the nonlinear dynamics (13).

However, with sampling rate of the outer (image-based pointing-tracking) loop fast enough, the error introduced by the coupling between the two camera axes would be corrected in the very next step, when the regulation error in vertical direction

in the image plane is detected and a correcting command to the elevation motor would be sent. The currently implemented prototype system achieves sampling rate of 15 Hz, which seems enough to justify this naive approach. Having scanned the available literature, the authors can only suspect that some of the available commercial systems follow this approach too. The motivation for this paper is to improve this scheme, because a bit more advanced and computationally intensive computer vision algorithms can slow down the sampling rate of the outer loop to something like 1 or 2 s.

VI. Feedback Linearization-Based Visual Pointing and Tracking

The key idea for an improvement described in the rest of the paper is that the curvature of the coordinate axes as in Fig. 8 can be compensated for by measuring the third component of the inertial angular rate of the camera body, the one along its optical axis, the so far unused measurement ω_{Ex}. It is available at the sampling rate a few orders of magnitude faster than what the computer vision system provides. Using this information, exact feedback linearization can be implemented in the controller following standard techniques from image-based visual servoing introduced next.

The idea behind image-based visual servoing is that an error "sensed" in the image plane by the image tracker as

$$e(t) = s(t) - s^{\text{ref}} \tag{18}$$

can be eliminated by commanding a proper value of $\xi(t)$, which characterizes the velocity of the camera frame. Note that $s^{\text{ref}} = 0$ when the task is to bring the image of the object into the central position in the image plane by pushing $s(t) = [u(t), w(t)]^T$ to zero. How to find a proper ξ? Simply by inverting the interaction matrix L. In the case of a single-point feature, the matrix is 2×6, which suggests that such a solution will not be unique. Which one to pick? It will be shown shortly that there is one important constraint here which makes only one solution acceptable.

In contrast to common robotics tasks, here we cannot influence the translational position of the camera frame (unless there is a bidirectional communication between the UAV autopilot and the inertial stabilization & visual tracking system). Hence the linear velocity $v_C = [v_{Cx}, v_{Cy}, v_{Cz}]^T$ of the camera coordinate origin needs to be taken as given (enforced) from the outside. But then the task of determining ξ at a given time instant consists in solving the linear system (10) with the term corresponding to translation moved to the right-hand side

$$L_\omega(u,w)\omega_C = \dot{s} - L_v(u,w,z)v_C. \tag{19}$$

The 2×3 matrix L_ω has a 1-D nullspace parameterized by

$$\mathcal{N} = \left\{\omega_c = k[u \quad w \quad \lambda]^T\right\} \tag{20}$$

which can be interpreted quite intuitively: rotating the camera about the line connecting the observed point and the origin of the camera frame does not contribute to a change of the coordinates of the point in the image plane. With the right pseudoinverse of L_ω given by

$$L_\omega^\sharp = \begin{bmatrix} 0 & \frac{\lambda}{\lambda^2+u^2+w^2} \\ -\frac{\lambda}{\lambda^2+u^2+w^2} & 0 \\ \frac{w}{\lambda^2+u^2+w^2} & -\frac{u}{\lambda^2+u^2+w^2} \end{bmatrix} \tag{21}$$

all solutions are parametrized by a single constant k

$$\omega_C = L_\omega^\sharp \dot{s} - L_\omega^\sharp L_v v_C + k[u \quad w \quad \lambda]^T. \tag{22}$$

Substituting and abusing k since it is arbitrary we get

$$\omega_C = \frac{1}{z(\lambda^2+u^2+w^2)} \begin{bmatrix} \lambda^2 v_y - \lambda v_z w + \lambda \dot{w} z + ku \\ -\lambda^2 v_x + \lambda v_z u - \lambda \dot{u} z + kw \\ -\lambda u v_y + \lambda w v_x - u\dot{w}z + \dot{u}wz + k\lambda \end{bmatrix}. \tag{23}$$

What we have obtained so far is a procedure which for given velocities $\dot{u}$ and $\dot{w}$ of a point feature in the image plane computes the required angular velocity vector ω_C (it does not hurt now to use the full notation $\omega_{R,C}^C$) of the camera. A single arbitrary parameter k can be used to give some choice, which is the key idea to be exploited next.

A. Simple Proportional Image-Based Pointing and Tracking

In order to pull $s(t)$ to the vicinity of (0,0) in the image plane, a cascade control structure can be used: the pointing-tracking controller sets the reference rate vector $\dot{s}^{\text{ref}}(t)$ such that its actual value $s(t)$ goes to zero. One simple approach is to require exponential stability, that is, both $u(t)$ and $w(t)$ go to zero values according to

$$\dot{s}(t) = As(t) \tag{24}$$

where A has nonnegative eigenvalues. The simplest solution can be obtained by restricting A to a diagonal matrix $A = -\alpha I$ for some real positive α and then

$$\dot{s}(t) = -\alpha s(t). \tag{25}$$

The larger the α, the faster the error in the image plane goes to zero. Practical considerations of the choice of this parameter are discussed at the end of this section. Now, how can we force this error to evolve as in (25)? Noting that $\dot{s}(t)$ is related to the camera inertial angular velocities according to (23), we can conclude that asymptotically stable image error is guaranteed if the camera inertial velocities follow the reference value

$$\omega_C^{\text{ref}} = \frac{1}{z(\lambda^2+u^2+w^2)} \begin{bmatrix} \lambda^2 v_y - \lambda v_z w - \lambda\alpha w z + ku \\ -\lambda^2 v_x + \lambda v_z u + \lambda\alpha u z + kw \\ -\lambda u v_y + \lambda w v_x + k\lambda \end{bmatrix}. \tag{26}$$

It is not clear at this moment whether and how such a rotation rate of the camera can be established by the two motors. It is the free parameter k that can help to pick such a reference inertial velocity vector ω_C^{ref} of the camera that is realizable by the two motors.

B. Establishing the Camera Inertial Rate Using Two Motors

Once we know the required inertial angular rate of the camera, what remains is to express it via constant transformation R_C^E in the inner gimbal frame

$$\omega_E^{\text{ref}} = R_C^E \omega_C^{\text{ref}} = \frac{1}{z(\lambda^2+u^2+w^2)} \begin{bmatrix} -\lambda u v_y + \lambda w v_x + k\lambda \\ -\lambda^2 v_y + \lambda v_z w + \lambda\alpha w z - ku \\ \lambda^2 v_x - \lambda v_z u - \lambda\alpha u z - kw \end{bmatrix}. \quad (27)$$

The task for the inertial angular rate control system is to follow this velocity by commanding the two motors. It is important to keep track of the corresponding frames. The resulting ω_E^{ref} is fully labeled as $\omega_{R,E}^{E,\text{ref}}$ as it gives the required inertial rotation rate of the inner gimbal. Its true value is measured by the three-axis MEMS gyro fixed to the inner gimbal.

Now comes the key part. Having only two motors, it is not possible to set all the three components of the vector of inertial angular velocity independently. But the free scalar parameter k can be used to pick a specific triple that requires no change with respect to the current value of ω_{Ex} (inertial angular rate of the camera around its optical axis). The value of ω_{Ex} must be available to the controller then. Solving (27) for the value of k guaranteeing that the x component of the vector on the right is equal to the measured ω_{Ex} gives

$$k = -wv_x + uv_y + \frac{z(\lambda^2+u^2+w^2)}{\lambda}\omega_{Ex}. \quad (28)$$

Substituting this value back to the expressions for the other two components of the reference angular rate vector, the expressions for the controller outputs follow

$$\omega_{\text{EL}}^{\text{ref}} = \omega_{Ey}^{\text{ref}} = \frac{\alpha w\lambda}{\lambda^2+u^2+w^2} - \frac{\omega_{Ex}u}{\lambda} - \frac{\lambda^2 v_y - \lambda w v_z - uwv_x + u^2 v_y}{z(\lambda^2+u^2+w^2)} \quad (29)$$

$$\omega_{\text{CEL}}^{\text{ref}} = \omega_{Ez}^{\text{ref}} = -\frac{\alpha u\lambda}{\lambda^2+u^2+w^2} - \frac{\omega_{Ex}w}{\lambda} + \frac{\lambda^2 v_x - \lambda u v_z - wuv_y + w^2 v_x}{z(\lambda^2+u^2+w^2)}. \quad (30)$$

The expressions (29) and (30) for the controllers are structured such that three terms can be immediately recognized in each controller: a term corresponding to a regulation error in the corresponding axis as seen in the image plane, a term compensating for the rotation around the camera optical axis and finally a term attenuating the influence of mutual translational motion of the camera and the ground object.

In order to get an insight into this new controller and compare it with the originally proposed decoupled one, consider again the easy situation when the carrier is in level flight and the camera is pointing towards the horizon ($\theta = 0$). Neglect the translational velocities v_C. The observed object is vertically centered in the image plane, that is, $w = 0$. The expressions in (29) and (30) simplify to

$$\omega_{Ey}^{\text{ref}} = 0 \quad (31)$$

$$\omega_{Ez}^{\text{ref}} = -\alpha\frac{\lambda}{\lambda^2+u^2}u. \quad (32)$$

Compare this simplified controller and the model of the system (15) valid for the same conditions. Apparently, the nonlinear term $\lambda/(\lambda^2+u^2)$ serves just to invert the nonlinearity in the model. And this is what the controller does in general. It inverts the nonlinearity. In other words, it performs feedback linearization. Consideration of the inertial angular rate ω_{Ex} of the camera around its optical axis is another measure that the controller takes to invert the nonlinearity. For the maximum zoom ($\lambda = 42$ mm), the nonlinear term $\lambda/(\lambda^2+u^2)$ is sufficiently close to λ and therefore the controller's action is driven by

$$\omega_{Ez}^{\text{ref}} = -\alpha\frac{1}{\lambda}u. \quad (33)$$

The simplification can take place even in a more general situation $u, w \neq 0$ but small, and λ large (and v_C still neglected). The general expression for the controller output then reduces to

$$\omega_{Ey}^{\text{ref}} = \frac{\alpha w}{\lambda} - \frac{\omega_{Ex}u}{\lambda} \quad (34)$$

$$\omega_{Ez}^{\text{ref}} = -\frac{\alpha u}{\lambda} - \frac{\omega_{Ex}w}{\lambda}. \quad (35)$$

This reduced controller reveals the key enhancement with respect to the fully decoupled design: the controller output contains contribution from the angular rate of the camera around the optical axis!

C. Summary of Controller Structure for Pointing and Tracking

The feedback-linearizing pointing-tracking controller in (29) and (30) does not preserve the decoupled structure (no longer two separate pointing-tracking controllers). Each of the two controllers accepts not only both the "measured" position errors, that is, u and w, but also it follows that:

1) the x-component of the vector ω_E describing the inertial angular rate of the camera around the optical axis;
2) estimates of the aircraft translational velocity with respect to the ground, expressed in the camera coordinate frame (v_{Cz} describes how fast the camera is approaching the target);
3) an estimate of depth z of the image, that is, the distance from the camera to the ground target.

Moreover, the technical parameter that the controller must be aware of is the focal length λ. An upgrade of the naive scheme proposed in Fig. 7 can thus be seen in Fig. 9.

The key challenge in implementing this controller fully is in providing the controller with the extra measurements and/or estimates of the three components of the translational velocity v_C and the distance z to the object. These could be approached using inertial measurement unit (IMU) in combination with a laser range-finder and possibly also in combination with a computer vision system. For instance, the depth z and the "towards the object" velocity v_{Cz} is sometimes estimated from the apparent size of an object in the image (covering the image of the object by some polygon and computing its area, which is suggested in [8]). This technique can turn out of limited use here, though, because the images of observed objects can span just a few pixels and determination of v_{Cz} is then very inaccurate.

On the other hand, these new "complications" caused by the requirements of measuring the translational velocities are not really new and tied to the proposed control scheme. They are

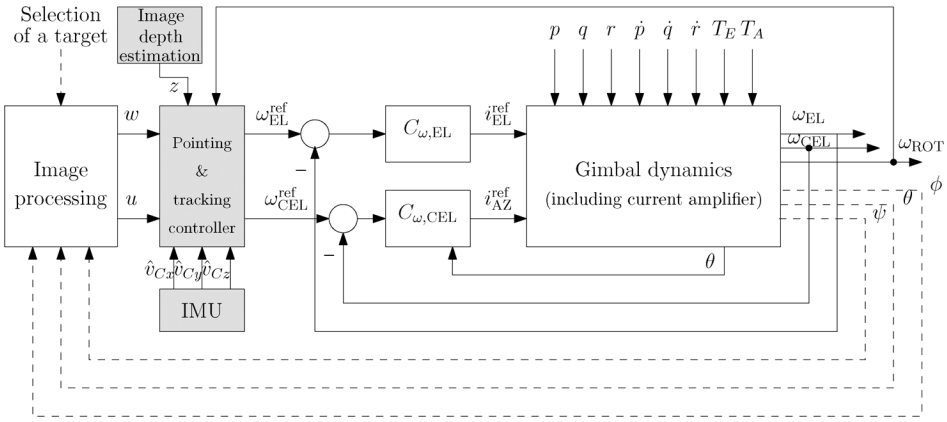

Fig. 9. Full feedback system with an image-based pointing controller aware of the angular rate about the optical axis and the translational motion. The pointing-tracking controller implements (29) and (30).

equally valid even with the (naive) decoupled control. Unless the translation velocity is known, one simply cannot tell whether the image is moving due to undamped aircraft oscillations or because the aircraft is approaching the object. But now, with the systematic analysis documented in this paper, the structure of the ideal controller is known. It is up to an engineer to decide whether or not to ignore the translational motion and regard its effects as unmeasured disturbance. Such disturbance is only significant at low frequencies and can be left for the image-based pointing loop to attenuate.

D. Practical Considerations for Setting the Image Dynamics

It is well known that the basic version of image-based visual servoing can be inefficient in terms of a requested manipulator movement needed to establish the requested image features trajectory. Several approaches have been proposed in the literature, see [25]–[27], that tackle the problem by separating the rotational and translational motion around and along the optical axis from the remaining controlled degrees of freedom. These issues have not been studied in the present situation since the only two controlled degrees of freedom are two rotations. The defective behavior of *camera retreat* is not present.

On the other hand, there is one practical aspect of setting the dynamics in the image plane that must be taken into consideration: placing the eigenvalues of the linearized system (25) too far in the left half plane makes the motional response of the system too fast for the image tracker. The image features then travel so fast in the image plane that the image tracker looses the grasp of the object (most image tracking algorithms such as [28] explore the nearest neighborhood only).

VII. Numerical Simulations

Closed-loop responses with the two controllers were simulated for the image of the observed object initially located out of the center of the image frame. The control goal is to bring the observed object into the center of the field of view.

Three simulations were run (always with both controllers): Two with the sampling rate 15 Hz for different initial locations of the image of the object. And one for the slower sampling at 2 Hz. Results are visualized in Figs. 10–12.

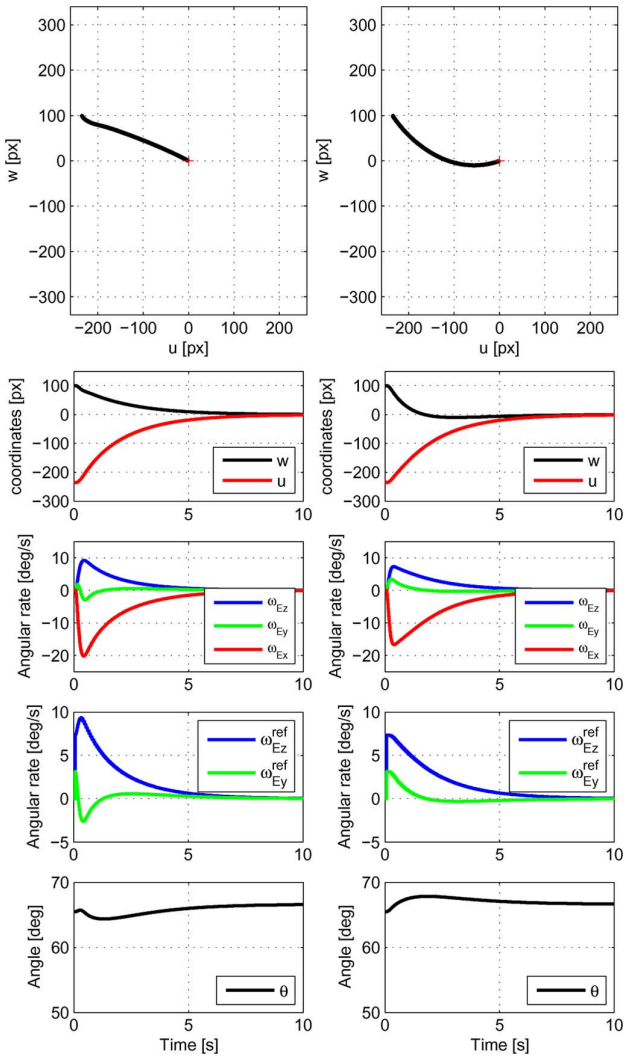

Fig. 10. Simulation 1: Responses of relevant quantities for $\theta(0) = 65.5°$, tracker sampling rate $f_{\rm sp} = 15$ Hz, $\alpha = 0.46$. Left: The proposed algorithm, right: the original decoupled approach.

Unlike in the design stage, in the simulations the inner (inertial rate) loops are not assumed to work ideally. That is, for the purpose of simulations their transfer functions are not iden-

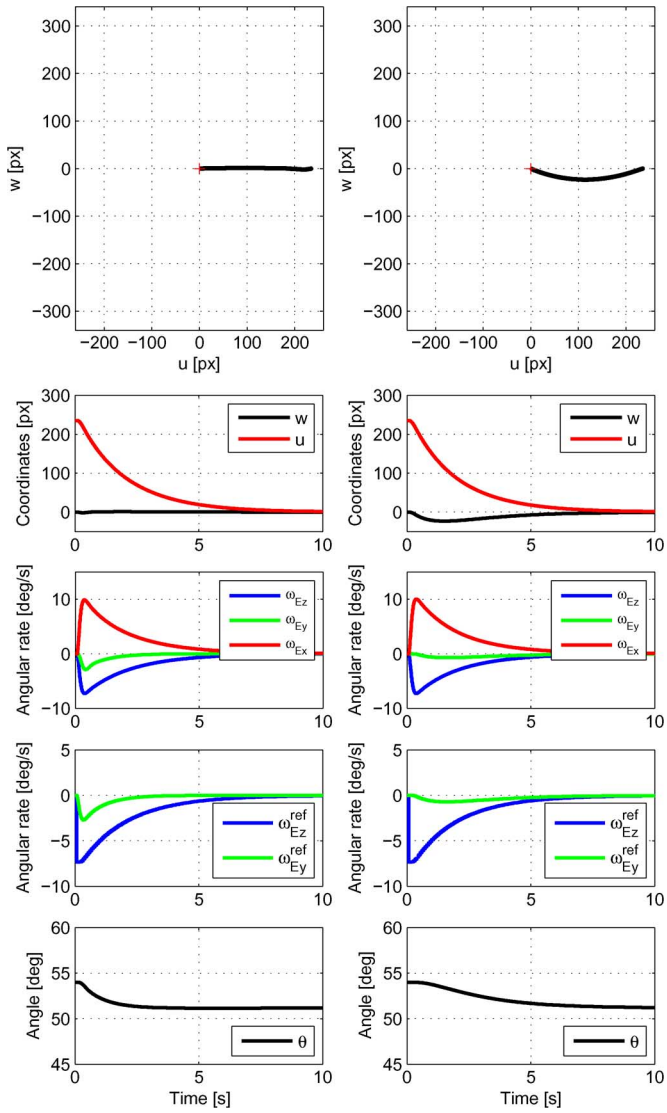

Fig. 11. Simulation 2: Responses of relevant quantities for $\theta(0) = 54°$, tracker sampling rate $f_{sp} = 15$ Hz, $\alpha = 0.46$. Left: The proposed algorithm, right: the original decoupled approach.

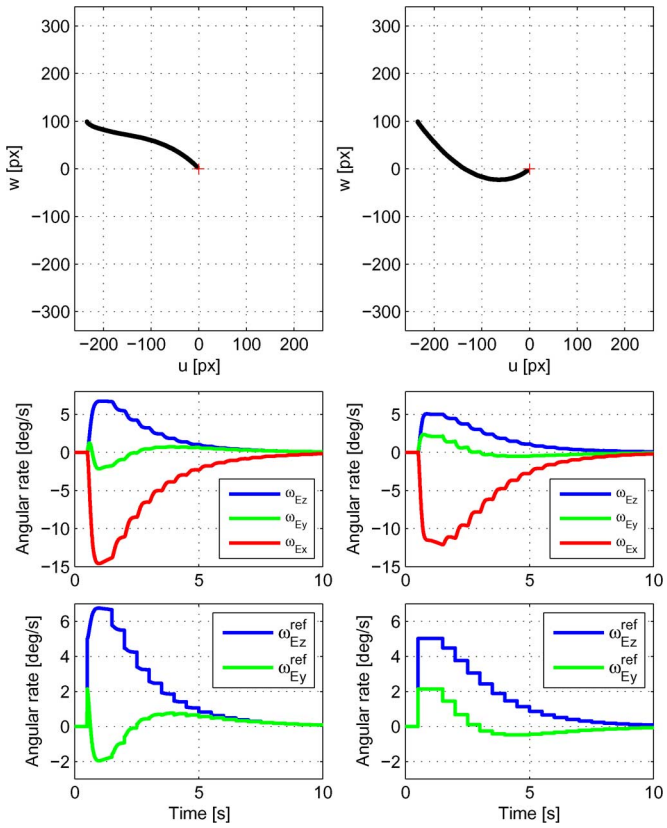

Fig. 12. Simulation 3: Responses of relevant quantities for $\theta(0) = 65.5°$, tracker sampling frequency $f_{sp} = 2$ Hz, $\alpha = 0.31$. Left: The proposed algorithm; right: the original decoupled approach.

tically equal to 1. The two SISO inertial rate loops are just standard feedback interconnections of a first-order system and P or PI regulator. Hence they can be modeled by low pass filters with a given bandwidth. However, the laboratory experiments described in the next section (see 15 and Figs. 16) reveal that this bandwith depends on the magnitude of the step in the reference inertial rate. This behavior is a consequence of a constraint on the amplitude of the control voltage inside the rate loops; a nonlinear model should be used to describe the rate loop more precisely. However, in order to obtain just a rough estimate of the system response, a low-order low pass filter seems satisfactory. The laboratory experiments with a real device then give a true assessment of the system performance.

No translation between the carrier and the observed object was assumed. The only physical parameters are those of the optics: focal length $\lambda = 42$ mm and resolution of the camera CCD chip is 640 × 480 pixels (u and w were scaled to pixels for visualization purposes).

As discussed in the section devoted to the image tracker, the process of extracting the pointing tracking error from the visual information devours some computational time, hence u and w are only updated with the sampling rate ranging from 0.5 to 15 Hz, depending on the complexity of the algorithm. For simulations we choose 15 and 2 Hz. In addition, the data is always available to the controller with a delay of one sampling period. The sampling rate of the inner loop is up to two orders of magnitude higher (250 Hz, the maximum provided by the inertial angular rate sensors used in the project).

The simulation results confirm that the new controller struggles to follow a linear path in the image plane. Not only is the linear behavior easier to analyze but also is more plausible for a human operator. Some deterioration of a perfectly linear path in the image plane is visible. This is due to the nonideal inertial rate loops that were considered in the simulations.

The third simulation with the slower image tracker sampling rate of 2 Hz demonstrates that the new proposed algorithm updates the reference values for camera inertial rates at a faster sampling rate than the original decoupled controller.

VIII. Compensating for the One-Period Delay in the Computer Vision System

The simulation results in the previous section differ from the linear expectations not only because of the nonideal inertial loops but also because of the delays introduced by the computer-vision system. The image tracker not only constrains the

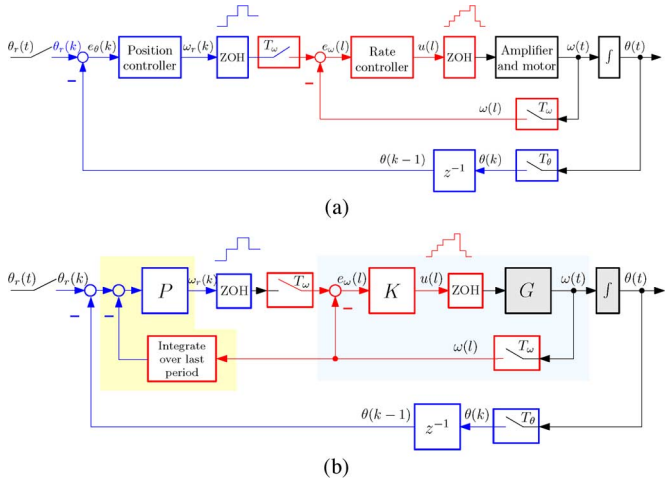

Fig. 13. (a) The inner (inertial rate) loop works at a fast sampling rate. The outer (image-based pointing) loop works at a slow sampling rate and suffers from a one-sampling-period delay. (b) Solution: integrate the rate ω over one *slow* sampling period, then subtract from the computed position error once the delayed data from the computer vision system arrive, and reset the integrator.

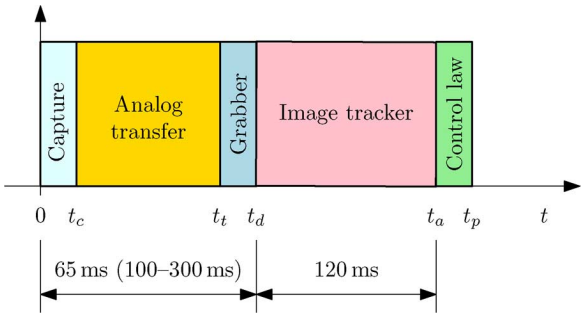

Fig. 14. Composition of the total delay in the visual pointing and tracking feedback loop: t_c is the time needed to capture the image (5–20 ms), which depends on the scene brightness; $t_t - t_c$ is the time for an analog transfer of the image (40 ms); $t_d - t_t$ is the time for digitization by a grabber. These three activities together can take about 65 ms when using the proprietary drivers provided with the camera, and 100–300 ms when using DirectX technology. The next tasks finishing at t_a is the actual visual tracking algorithm (120 ms on a common PC platform). The final task finishing at t_p is responsible for the pointing and tracking control law and consumes a negligible computational time. The sampling period is then given by the total delay. Alternatively, another image capturing can start right after t_d.

outer (pointing) loop to operate at a much slower sampling rate then the inner (inertial rate) loop but it also introduces one sampling period delay into the outer loop, see Fig. 13(a). This is a common problem encountered in visual servoing as discussed, for instance, in [29]. The cascade structure used in this particular system offers a simple solution in Fig. 13(b).

The composition of the total delay is visualized in Fig. 14. The sampling period is then given by this delay.

The simple solution consists in integrating the inertial angular rate ω as measured by the gyros over one *slow* sampling period and then subtracting from the computed position error once the delayed data from the vision system arrive. The discrete-integrator is then reset. This intuitive solution is investigated in [30] where the authors show that it is similar but not completely equivalent to a modified Smith predictor, which offers even better performance. The idea, being unrelated to the central topic, is not elaborated on further.

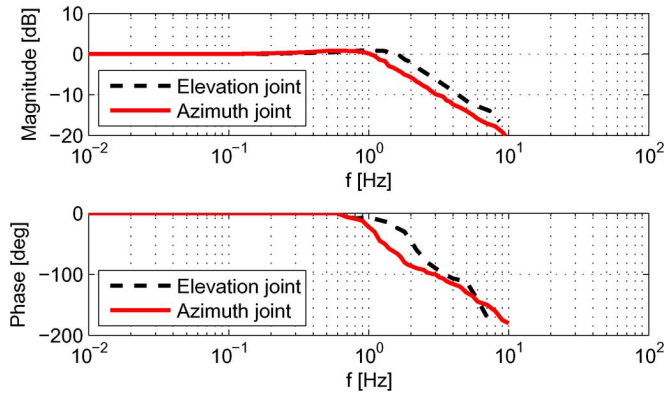

Fig. 15. Measured frequency responses for both inertial rate loops. The required inertial rate as an input and the true (measured) inertial rate as an output. The amplitude of the reference inertial rate was set up to 73°/s, which is 10× higher than in Fig. 16.

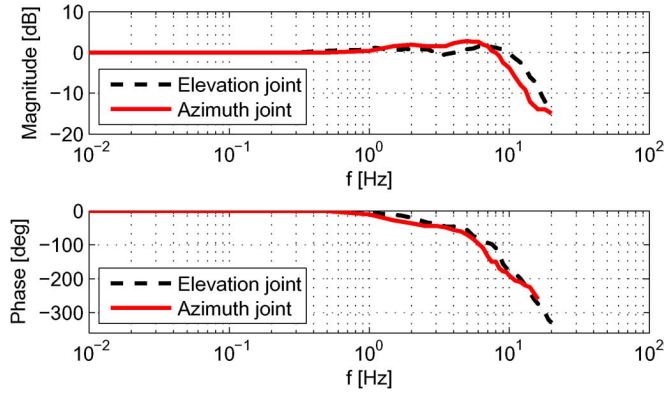

Fig. 16. Measured frequency responses for both inertial rate loops. The required inertial rate as an input and the true (measured) inertial rate as an output. The amplitude of the reference inertial rate was set up to 7.3°/s, which is 10× lower than in Fig. 15.

IX. Laboratory Experiments

The benchmark system introduced in the paper was used to validate the functionality of the proposed control scheme and compare its performance with the intuitive decoupled controller. The experimental test was conducted in an indoor lab while the camera platform was carried by a fixed laboratory stand. Therefore it was only the objects on the ceiling rather than on the floor that could be tracked conveniently. Both the new and the original (naive) decoupled controllers were tested only for the faster sampling rate of 15 Hz of the automatic image tracker. The controllers already included the simple heuristic delay compensation sketched above (although it was not much needed with this fast sampling rate).

The bandwidth of the inertial rate loop can be experimentally demonstrated to be at least 1 Hz, see Figs. 15 and 16. Therefore the assumption that the rate loop guarantees tracking of the required inertial rates up to a fraction of Hz is satisfied.

Three experiments were conducted and the measurements are visualized here in Figs. 17–19, the experimental data for the new algorithm always on the left and the data for the original decoupled scheme on the right.

The first two experiments were quite similar: the platform was in both cases sitting peacefully on the desk and the only differ-

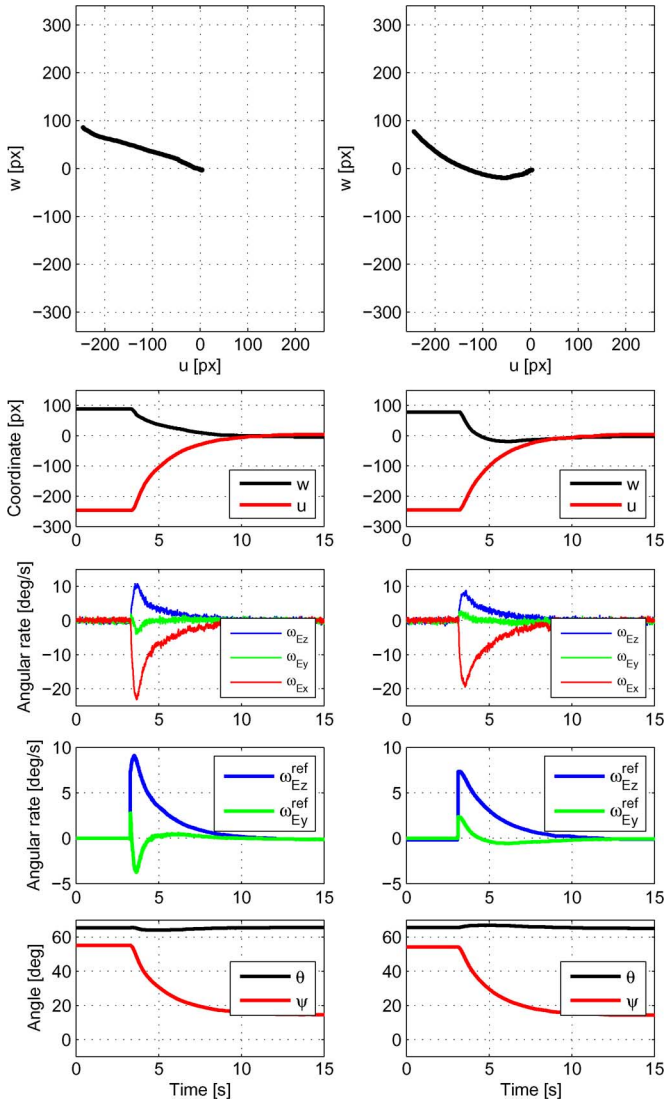

Fig. 17. Experiment 1: Responses of relevant quantities for $\theta(0) = 65.5°$, tracker sampling rate $f_{sp} = 15$ Hz, $\alpha = 0.46$. Left: The proposed algorithm; right: the original decoupled approach.

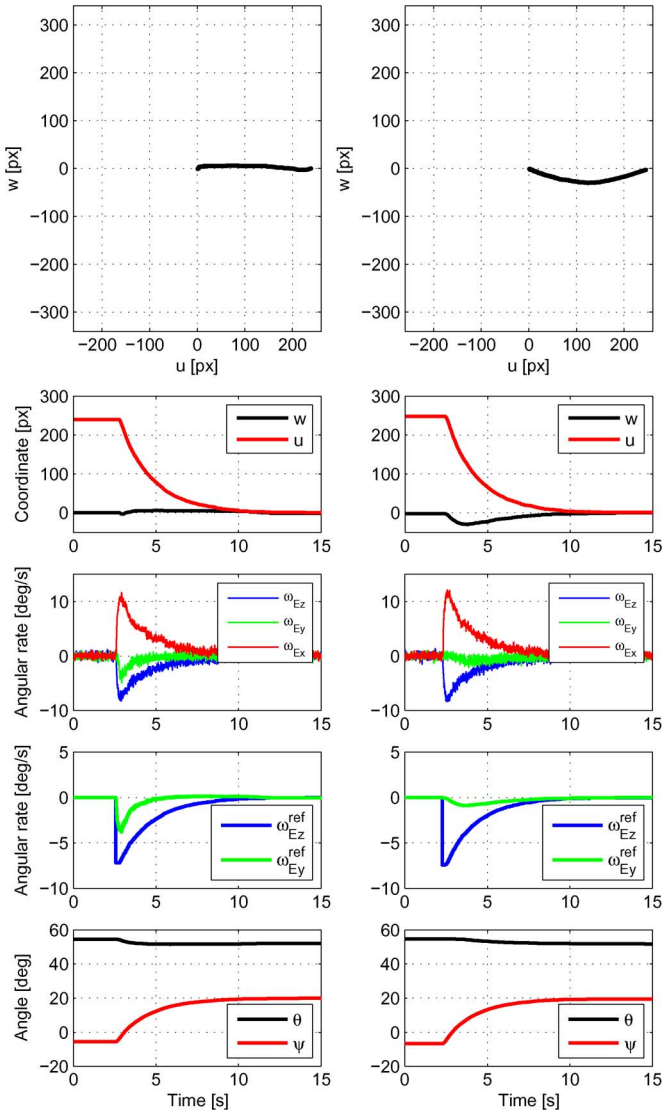

Fig. 18. Experiment 2: (left) The proposed algorithm; (right): the original decoupled approach. The initial elevation angle was set to $\theta(0) = 54°$. The image tracker sampling rate $f_{sp} = 15$ Hz. The controller parameter $\alpha = 0.46$.

ence was the initial elevation of the camera ($\theta(0) = 65.5°$ and $\theta(0) = 54°$ as in the simulations) and the position of the point to be tracked in the image plane. The measurements are in Figs. 17 and Fig. 18 and can be easily compared with the simulation results in Figs. 10 and Fig. 11. Apparently, the responses for both control methodologies are quite similar as for the time scale and control magnitude (but see the difference in one of the controller outputs ω^{ref}_{Ey} in Fig. 17). The key difference is that the new algorithm achieves a linear trajectory in the image plane as desired.

The third experiment validates the pointing and tracking performance even in presence of a disturbing rotational motion of the carrier. Namely, the optical axis of the camera was initially pointing to the ceiling with the elevation $\theta(0) = 70°$ and the laboratory stand was rotated manually around its vertical axis (orthogonal to the surface of the desk). The measured outcomes are in Fig. 19 both for the new algorithm and for the decoupled scheme, both of which use the same inner (inertial angular rate stabilization) feedback loop. The conclusion is that both algorithms exhibit the same characteristic behavior already known from the simulations and static experiments. Actually, the measurements related to the new controller appear a bit more disturbed but this is only due to the fact that the disturbing motion was induced manually, hence a bit differently in both cases.

X. Conclusion

This paper presented a systematic procedure for designing and implementing a pointing and tracking image-based controller for an airborne camera platform with an inertial line-of-sight stabilization already designed and implemented. The proposed scheme uses extra information from an inertial angular rate sensor; namely, the angular velocity of the payload (camera, laser) around its optical axis. This extra measurement is provided by a MEMS gyro at a much faster sampling rate than the pointing-tracking error produced by a computer vision system. Moreover, the proposed controller can take into consideration the measured or estimated translation velocity of the aircraft

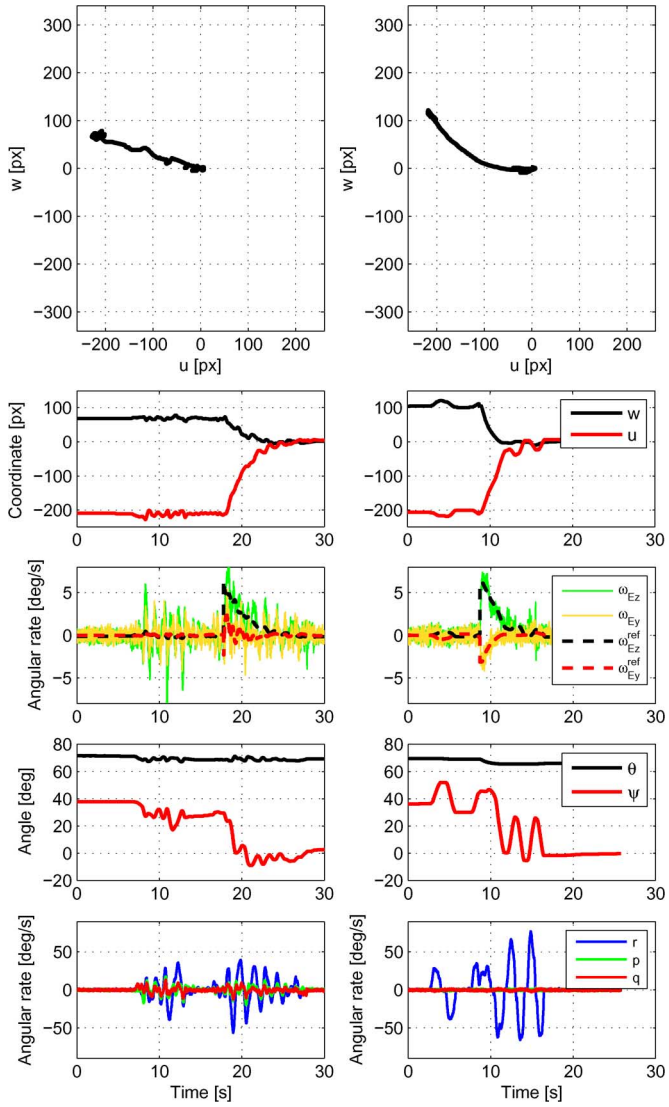

Fig. 19. Experiment 3: Responses of relevant quantities for the pointing exposed to external (recorded) disturbances p, q, r. $\theta(0) = 70°$, tracker sampling rate $f_{\rm sp} = 15$ Hz, $\alpha = 0.46$. Left: The proposed algorithm; right: the original decoupled approach.

with respect to the observed target (to compensate for the paralactic phenomenon).

The essence of the proposed design technique is that of a feedback linearization. The resulting controller enforces linear dynamics in the image plane. This not only makes the analysis and design systematic but putting it on the well-explored ground, but also makes the response of the system a bit more friendly for a human operator as the system follows linear paths in the image plane during (re)pointing.

The proposed scheme was thoroughly simulated and verified by practical laboratory experiments with a realistic benchmark system and compared against the more intuitive decoupled control scheme. Possible simplifications were discussed and practical pitfalls were highlighted.

ACKNOWLEDGMENT

The authors acknowledge consulting with J. Hilkert (Alpha-Theta Technologies) and D. Kostić (TU Eindhoven). The benchmark platform was co-developed by J. Žoha (CTU FEE), M. Bartoš (ESSA Prague Ltd.), and J. Nohýl (VTÚLaPVO). The credit for the implementation of the image tracker algorithm goes to P. Krsek (CTU FEE).

REFERENCES

[1] *Selected Papers on Precision Stabilization and Tracking Systems for Acquisition, Pointing and Control Applications*, ser. SPIE Milestone Series, M. Masten and L. Stockum, Eds.. Bellingham, WA: SPIE, 1996, vol. MS 123.

[2] M. Masten, "Inertially stabilized platforms for optical imaging systems: Tracking dynamic dynamic targets with mobile sensors," *IEEE Control Syst. Mag.*, vol. 28, no. 2, pp. 47–64, Feb. 2008.

[3] J. M. Hilkert, "Inertially stabilized platform technology: Concepts and principles," *IEEE Control Syst. Mag.*, vol. 28, no. 2, pp. 26–46, Feb. 2008.

[4] J. Debruin, "Control systems for mobile satcom antennas," *IEEE Control Syst. Mag.*, vol. 28, no. 1, pp. 86–101, Feb. 2008.

[5] J. Osborne, G. Hicks, and R. Fuentes, "Global analysis of the double-gimbal mechanism," *IEEE Control Syst. Mag.*, vol. 28, no. 4, pp. 44–64, Aug. 2008.

[6] Z. Hurák and M. Řezáč, "Combined line-of-sight inertial stabilization and visual tracking: Application to an airborne camera platform," presented at the 48th IEEE Conf. Decision Control, Shanghai, China, 2009.

[7] Z. Hurák and M. Řezáč, "Control design for image tracking with an inertially stabilized airborne camera platform," presented at the SPIE Deference, Security, Sensing, Orlando, FL, 2010.

[8] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. New York: Wiley, 2006.

[9] A. Rue, "Stabilization of precision electrooptical pointing and tracking systems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-5, no. 5, pp. 805–819, Sep. 1969.

[10] A. Rue, "Precision stabilization systems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-10, no. 1, pp. 34–42, Jan. 1974.

[11] R. Kumar, H. Sawhney, S. Samarasekera, S. Hsu, H. Tao, Y. Guo, K. Hanna, A. Pope, R. Wildes, D. Hirvonen, M. Hansen, and P. Burt, "Aerial video surveillance and exploitation," *Proc. IEEE*, vol. 89, no. 10, pp. 1518–1539, Oct. 2001.

[12] J. Nygårds, P. Skoglar, M. Ulvklo, and T. Högström, "Navigation aided image processing in UAV surveillance: Preliminary results and design of an airborne experimental system," *J. Robot. Syst.*, vol. 21, no. 2, pp. 63–72, Feb. 2004.

[13] Q. Zhu, K. Cheng, and H. Zhang, "SSD tracking using dynamic template and log-polar transformation," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, 2004, pp. 723–726.

[14] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. 7th Int. Joint Conf. Artif. Intell.*, 1981, pp. 674–679.

[15] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–575, May 2003.

[16] T. Kanade and C. Tomasi, "Detection and tracking of point features," Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-91-132, 1991.

[17] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE Int. Conf. Comput. Vision*, 1999, p. 1150.

[18] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image Vision Comput.*, vol. 22, no. 10, pp. 761–767, Sep. 2004.

[19] M. Donoser and H. Bischof, "Efficient maximally stable extremal region (MSER) tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Patt. Recog.*, 2006, vol. 1, pp. 553–560.

[20] P. Viola and M. Jones, "Robust real-time object detection," *Int. J. Comput. Vision*, vol. 57, no. 2, pp. 137–154, 2002.

[21] H. Grabner, J. Sochman, H. Bischof, and J. Matas, "Training sequential on-line boosting classifier for visual tracking," in *Proc. 19th Int. Conf. Patt. Recog. (ICPR)*, 2008, pp. 1–4.

[22] F. Chaumette and S. Hutchinson, "Visual servo control. i. Basic approaches," *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, Dec. 2006.

[23] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*, 2nd ed. New York: Wiley, 2005.

[24] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital Control of Dynamic Systems*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1997.

[25] K. Deguchi, "Optimal motion control for image-based visual servoing by decoupling translation and rotation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1998, pp. 705–711.
[26] E. Malis, F. Chaumette, and S. Boudet, "2-1/2 d visual servoing," *IEEE Trans. Robot. Autom.*, vol. 15, no. 2, pp. 238–250, Apr. 1999.
[27] P. Corke and S. Hutchinson, "A new partitioned approach to image-based visual servo control," *IEEE Trans. Robot. Autom.*, vol. 17, no. 4, pp. 507–515, Aug. 2001.
[28] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework," *Int. J. Comput. Vision*, vol. 56, no. 3, pp. 221–255, 2004.
[29] M. Nemani, T. Tsao, and S. Hutchinson, "Multi-rate analysis and design of visual feedback digital servo-control system," *J. Dyn. Syst., Meas., Control*, vol. 116, no. 1, pp. 45–55, Mar. 1994.
[30] Z. Hurák and M. Řezáč, "Delay compensation in a dual-rate cascade visual servomechanism," presented at the 49th Conf. Decision Control, Atlanta, GA, 2010.

Zdeněk Hurák (M'00) received the Ing. (M.Sc.) degree in aerospace electrical engineering (*summa cum laude*) from Military Academy, Brno, Czech Republic, in 1997. He was awarded Boeing Fellowship in 1999 to stay for three months at Iowa State University, Ames. He received the Ph.D. in cybernetics and robotics supervised by Prof. M. Šebek (thesis on ℓ_1-optimal control) from the Czech Technical University (CTU), Prague, Czech Republic, in 2004.

He was a visiting researcher with TU Eindhoven, Eindhoven, The Netherlands, in 2008. He is currently an Assistant Professor with the Department of Control Engineering, Faculty of Electrical Engineering, CTU. His research interests include optimal and robust control and applications to electromechanical systems such as inertially stabilized camera platforms, piezoelectric micro-manipulators, and non-contact micro-manipulation using dielectrophoresis.

Prof. Hurák chairs Control Systems Chapter of Czech-Slovak Section of IEEE and is a member of the editorial board of Kybernetika journal and Czech industry-oriented Automa journal.

Martin Řezáč received the Ing. degree (M.Sc.) in cybernetics and measurement with a major in control engineering (*summa cum laude*) from the Czech Technical University, Prague, Czech Republic, in 2008. He was one of the two key developers of the control system for the benchmark platform and his diploma thesis was awarded Dean's prize. He is a currently pursing the Ph.D. degree in the domain of inertial stabilization and estimation in combination with visual servoing.

He spent six months with Eindhoven University of Technology, Eindhoven, The Netherlands, thanks to the Erasmus Socrates Program.