Joint 48th IEEE Conference on Decision and Control and
28th Chinese Control Conference
Shanghai, P.R. China, December 16-18, 2009



# Combined line-of-sight inertial stabilization and visual tracking: application to an airborne camera platform

Zdeněk Hurák and Martin Řezáč

***Abstract*— The paper describes a novel pointing&tracking feedback algorithm for an inertially stabilized double gimbal airborne camera system equipped 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 translational motion must be measured or estimated. The proposed algorithm turns out to be more insensitive to longer sampling periods of the computer vision system.**

## I. INTRODUCTION

### A. *Inertial line-of-sight stabilization on mobile carriers*

The very basic control task for steerable cameras or antennas placed on mobile bases such as trucks, unmanned aircraft or ships, is to keep the commanded line of sight (optical axis) unchanged even in presence of various disturbing phenomena like mass disbalance, aerodynamic (or wind induced) torque and possible kinematic coupling between the gimbal axes. Driven also by military needs, the topic has been studied extensively in the past few decades. Several relevant papers from 1970s through 1990s were archived in a selection [1]. Confirmation, that the topic is still relevant for the engineering community can be found by noticing that the whole February 2008 issue of *IEEE Control System Magazine* was dedicated to the topic, bringing nice survey papers [2], [3] and [4]. 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.

### 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 [2] by suggesting the common cascaded control structure: each of the SISO inner (inertial rate stabilization) loops is accepting commands from the output of the corresponding outer (visual tracking) loop. This decoupled approach is sufficient as long as the gimbal system has (at least) three rotational degrees of freedom, that is, the motors can rotate the payload with complete freedom. As soon as the system offers less then that, design of a cascaded controller is not that straightforward. These troubles will be described and a possible solution to this problem will be offered in this paper. To avoid misunderstanding: no computer vision algorithms are dealt with in this paper, it is only the way in which this visual information enters the control system that is investigated here.

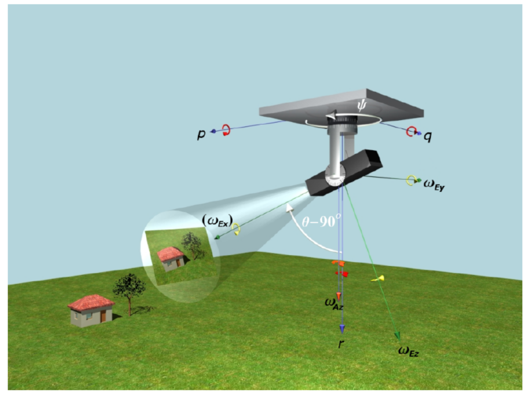



Fig. 1. Basic scenario for inertial line-of-sight stabilization (the ground objects may also move). Sketch of the basic configuration and roll, pitch and yaw rate vectors of the involved frames. Green vectors $\omega_{Ex}, \omega_{Ey}, \omega_{Ez}$ denote the rotation of the elevation frame (measured by MEMS gyros attached to the camera), blue vectors $p, q, r$ denote rotation of the base (UAV here). The vector $\omega_{Az}$ denotes inertial angular velocity of the outer gimbal (the other two components are not shown). Two white arcs denote the two angles caused by the two actuators. In computations, the origins of the coordinate frames are assumed to be in the same place.

### 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 change in heading (or azimuth) angle. Such a benchmark system was developed within a project coordinated by Czech Air Force Research 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. 3D descriptive visualization is in Fig.2(a), while a photo of a real system is in Fig.2(b). Direct drive motors are used and MEMS based gyros attached to the payload.

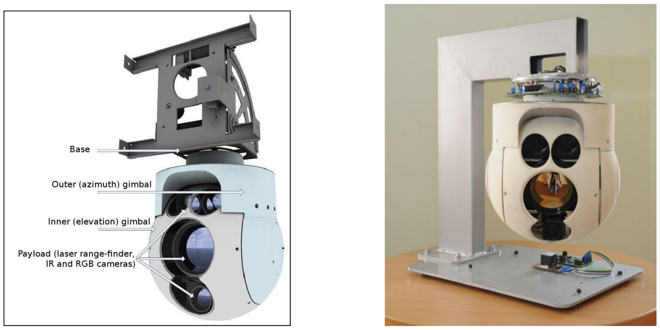
### D. *Notation*

This paper relies on expressing rotations of coordinate frames with respect to some other coordinate frames. The

This work was supported by the Ministry of Education of the Czech Republic within the project Centre for Applied Cybernetics (1M0567) and by Ministry of Industry and Trade of the Czech Republic within the project TIP FR-TI1/265.

Z. Hurák and M. Řezáč are with Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic. hurak@fel.cvut.cz

978-1-4244-3872-3/09/$25.00 ©2009 IEEE 




(a) 3D model of the platform: both the azimuth and the elevation angles can change $n \times 360^o$ thanks to sliprings.

(b) Photo of the resulting benchmark system.

Fig. 2. Stabilized platform developed by Czech Air Force Research Institute in collaboration with Czech Technical University in Prague and Essa company.

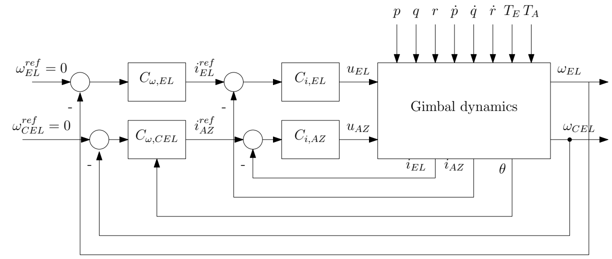



Fig. 3. Inertial stabilization. Two independent (decoupled) SISO feedback loops, one for each rate gyro attached to the body of camera, $\omega_{EL}(=\omega_{Ey})$ and $\omega_{CEL}(=\omega_{Ez})$. The cross-elevation stabilizing controller must contain secant gain correction $1/cos(\theta)$. The disturbing variables are roll, pitch and yaw rates, $p, q, r$, respectively, their derivatives and external torque around the two motor axes. The innermost current loops are also depicted.

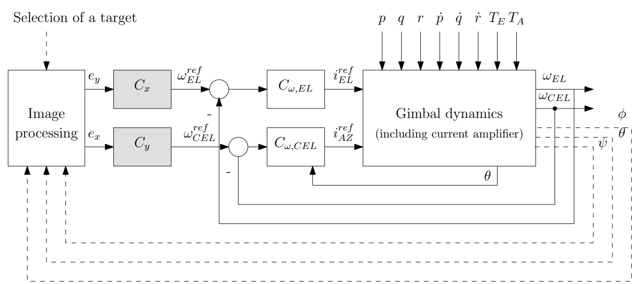



Fig. 4. Naive pointing&tracking system formed by two SISO loops closed around two inertial rate stabilization loops. The dashed lines are not signals truly fed back to the pointing&tracking controller. These are variables representing orientation of the camera which affects the position and orientation of objects in the image plane.

orthogonal coordinate frames used here and their symbols used in subscripts and superscripts are: the base coordinate frame attached to the body of the carrier [B], the ground coordinate frame [G], the reference coordinate frame (oriented as the ground frame but translated to the carrier) [R], the coordinate frame attached to the outer (azimuth) gimbal [A], the coordinate frame attached to the inner (elevation) gimbal [E] and the coordinate frame attached to the camera [C] (rotation of C with respect to E is fixed and is used just for convenience). The subscript/superscript scheme used here follows the common style, defined for instance in [6]: to specify the angular rate vectors, one needs to tell which coordinate frame is rotating with respect to which other coordinate frame and in which coordinate frame such a vector is expressed. For example, $\omega_{R,E}^{E}$ stands for inertial angular rate of the inner (Elevation) gimbal with respect to the Reference coordinate system, expressed in the coordinate frame attached to the elevation gimbal. Oftentimes, the notation is loosened in the paper to avoid cluttering the formulas with indices.

## 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 SISO inertial rate controllers suffice, one for each measured inertial angular rate, namely $\omega_{EL}$ for inertial angular rate of the payload about the axis of the elevation motor (pitch rate) and $\omega_{CEL}$ for inertial angular rate of the payload around its own vertical axis, also called cross-elevation rate since its axis is always orthogonal to the $\omega_{EL}$ axis. This is visualized in Fig.1. Note that in the figure the above mentioned angular rate vectors are denoted as $\omega_{Ey}$ and $\omega_{Ez}$ to emphasize that these are measured in the inner (Elevation) frame by gyros attached directly to the payload (body of the camera). The resulting decoupled controller configuration is in Fig.3. From Fig.1 it can be learnt that the cross-elevation stabilization controller must include a secant gain correction $1/cos(\theta)$, because it is only when the camera is pointing to the horizon ($\theta = 0$) that $\omega_{Az} = \omega_{CEL}(=\omega_{Ez})$. See [3].

Even though there is some gyroscopic coupling between the two axes [7] or [5] (for the simplified version when the base is still), its influence is not worth designing a 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. (NAIVE) DECOUPLED POINTING&TRACKING

Going one step further behind the mere inertial stabilization, the question of the right control configuration for automatic visual tracking pops up. Shall we use the immediate solution which simply builds a SISO tracking loop around the respective SISO (inertial) rate loop? This naive solution is depicted in Fig.4.

Keeping SISO controllers looks plausible from an implementation viewpoint. There is a trick hidden here, though, as seen in Fig.5. 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.4 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 $\omega_{CEL}$ (denoted as $\omega_{Ez}$ in the

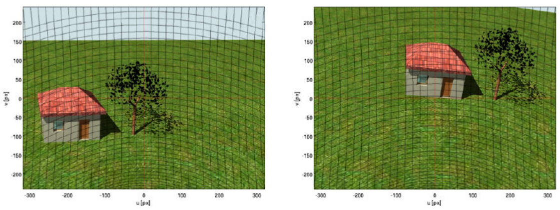


Fig. 5. Illustration of how in an attempt to steer the camera such that the image of the house gets back to the middle of the field of view using azimuth motor only, the introduced rotation of the camera around its optical axis makes the horizontal movement curved. Consequently, correction in vertical direction using the elevation motor is needed. Curvilinear coordinate system in the image plane corresponds to the elevation of camera by $\theta = -54^o$ with respect to the body of the aircraft.

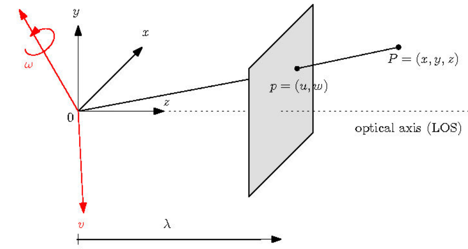



Fig. 6. Coordinates of the object on the ground expressed in the coordinate frame attached to the camera and (after projection) in the image plane. Rotation $\omega^C_{R,C}$ and translation $v_C$ of the camera frame within with respect to the inertial frame is also illustrated [6].

figure). However, the motor can only affect the component of the inertial rate in the direction of the azimuth motor axis, that is, $\omega_{Az}$. 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 $\omega_{Az}$ has some nonzero projection to the camera optical axis. Consequently, some unwanted rotation of the image as well as vertical displacement are introduced.

Curvilinear coordinate mesh in Fig.5 is generated by the nonlinear dynamics (1) for which detailed derivation is provided in the next section: combine (6) with the relationship between the inertial angular rates expressed in the inner frame (attached to the the camera body) and to the outer frame (attached to the azimuth motor), i.e., $\omega_{Ez}$ vs. $\omega_{Az}$. The key (reasonable) assumption of this model is that the inner (inertial rate) loop has already been designed properly and within its bandwidth (which is an order of magnitude larger then the image-driven pointing loop bandwidth) its behaviour can be modeled as a unity gain. The model described below also does not consider translational velocity of the aircraft and the observed object, these will be regarded as disturbances (see more in the next section).

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -\frac{uw}{\lambda} & \frac{\lambda^2+v^2}{\lambda} + w\tan(-\theta) \\ \frac{\lambda^2+w^2}{\lambda} & \frac{uw}{\lambda} - u\tan(-\theta) \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \omega^{ref}_{EL} \\ \omega^{ref}_{CEL} \end{bmatrix} \tag{1}$$

However, with fast enough sampling rate of the outer (image-based pointing&tracking) loop, the error introduced by the coupling between the two camera axes would be corrected in the very next step, when 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 10Hz, 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 easily lower the sampling rate of the outer loop to something like 2s.

## IV. ADVANCED POINTING&TRACKING

The key idea for improvement is that the curvature of the coordinate axes as in Fig.5 can be compensated for by measuring the inertial angular rate of the camera body around its optical axis. Such information is available at the rate a few orders of magnitude faster then what computer vision system offers. First, a few basic concepts from the established area of visual servoing will be given. This next few paragraphs are fully based on the two chapters from [6] dedicated to computer vision and vision-based control. They are given here just for a convenience of a reader nonacquainted with these concepts.

### A. Visual servoing

*1) Perspective projection:* The objects to be observed are located in the full 3D world while the camera can only record their 2D 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 defining the image coordinate frame as follows. At a focal distance $\lambda$ 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.6. Thanks to the pinhole assumption

$$k \begin{bmatrix} x & y & z \end{bmatrix}^T = \begin{bmatrix} u & w & \lambda \end{bmatrix}^T \tag{2}$$

we have that

$$u = \lambda \frac{x}{z}, \quad w = \lambda \frac{y}{z} \tag{3}$$

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

$$-\frac{u}{s_x} = (r - o_r), \quad -\frac{w}{s_y} = (c - o_c) \tag{4}$$

However, for control system analysis we will stick to the image coordinates just to make the notation less messy, visualization of simulation results being an exception.

*2) Camera motion and the interaction matrix:* This subsection is again more or less extracted from a nice introduction to image-based visual servoing in the textbook [6]. Consider the movement of the camera in the inertial space characterized by its linear and rotational velocities $v_C = [v_x, v_y, v_z]^T$ and $\omega_C = [\omega_x, \omega_y, \omega_z]^T$, both expressed in the camera frame. Put them together to form a vector $\xi = [v_C(t), \omega_C(t)]^T \in \mathbb{R}^6$. (To be rigorous, we should perhaps write $\omega_{R,C}^C$ to emphasize that it is rotation of the camera frame with respect to the fixed (inertial) frame aligned with the ground system but translated to the aircraft, and $v_{o_C}^C$ to emphasize that it is 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 the so-called image feature velocity $\dot{s}(t)$, which can be obtained as 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 $\xi$ and $\dot{s}$ by a linear transform resembling the concept of Jacobian and denoted often an interaction matrix or image Jacobian

$$\dot{s}(t) = L(s,q)\xi(t) \tag{5}$$

In the simplest case of a single point feature and assuming that the ground object does not move[1], this matrix is derived in [6], page 415, equation (12.14) as

$$\begin{bmatrix}\dot{u}\\ \dot{w}\end{bmatrix} = \begin{bmatrix} -\frac{\lambda}{z} & 0 & \frac{u}{z} & \frac{uw}{\lambda} & -\frac{\lambda^2+u^2}{\lambda} & w \\ 0 & -\frac{\lambda}{z} & \frac{w}{z} & \frac{\lambda^2+w^2}{\lambda} & -\frac{uw}{\lambda} & -u \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{6}$$

The procedure for the derivation was standard: 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 mind boggling. In this case, both translational and rotational velocities are indeed considered with respect to the ground 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

$$\dot{s} = L_v(u,w,z)v_C + L_\omega(u,w)\omega_C \tag{7}$$

because it turns out that only the part corresponding to translation of the camera coordinate frame depends on the depth $z$. The rotational part is independent of $z$.

[1]Extension of the results stated here to the case of a moving ground target is easily possible, but the resulting interaction matrix will then be a function of the velocity of the ground object, which is unknown to the inertial stabilization system. But perhaps it may be worth exploring if at least rough estimate of the object velocities can be used.

The basic idea behind image-based visual servoing is that an error evaluated by computer vision algorithms in the image plane $e(t) = s(t) - s_d$ (with $s_d = 0$ when the task is to bring the object into the central position in the image) can be eliminated by setting a proper value of $\xi(t)$, which characterizes the velocity of the camera frame. How to find a proper $\xi$? Simply by inverting the interaction matrix $L$. In the case of a single point feature, the matrix is $2 \times 6$, which suggests that such a solution will not be unique. Which one to pick? There is one constraint here.

In contrast with 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 $v_C = [v_x, v_y, v_z]^T$ of the camera coordinate origin needs to be taken as given (enforced) from the outside. But then the task of determining $\xi$ at a given time instant consists in solving the linear system (6) with the term corresponding to translation shifted to the right hand side as in (8).

$$L_\omega(u,w)\omega_C = \dot{s} - L_v(u,w,z)v_C \tag{8}$$

The $2 \times 3$ matrix $L_\omega$ has a 1-dimensional nullspace parametrized by

$$\mathcal{N} = \{\omega_c = k\begin{bmatrix}u & w & \lambda\end{bmatrix}^T\} \tag{9}$$

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 $L_\omega$ given by

$$L_\omega^\sharp = \begin{bmatrix} 0 & \frac{\lambda}{\lambda^2+u^2+w^2} \\ -\frac{\lambda}{\lambda^2+u^2+w^2} & 0 \\ \frac{w}{\lambda^2+u^2+w^2} & -\frac{u}{\lambda^2+u^2+w^2} \end{bmatrix} \tag{10}$$

all solutions are parametrized by a single constant $k$

$$\omega_C = L_\omega^\sharp\dot{s} - L_\omega^\sharp L_v v_C + k\begin{bmatrix}u & w & \lambda\end{bmatrix}^T \tag{11}$$

substituting and abusing $k$ since it is arbitrary

$$\omega_C = \frac{\lambda}{z\left(\lambda^2+u^2+w^2\right)} \begin{bmatrix} \dot{w}z - \lambda v_y + w v_z + ku \\ -\dot{u}z + \lambda v_x - u v_z + kw \\ -w\dot{u}z + \lambda w v_x + u\dot{w}z - \lambda u v_y + k\lambda \end{bmatrix} \tag{12}$$

What we have obtained so far is a procedure that for given velocities $\dot{u}$ and $\dot{w}$ of a feature (point) in the image plane computes the required angular velocity vector $\omega_C$ (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 later.

*3) Simple proportional image-based pointing and tracking:* As an output from the computer vision algorithm we get the ("measured") value $s(t) = [u(t), v(t)]^T$, which in the simplest case denotes position of a significant point (so called feature) in the image plane. The goal of the pointing&tracking controller is then to force this $s(t)$ as close to some demanded (reference) value $s_{ref}$. In our case

$s_{ref} = 0$, that is, the goal is to bring the observed object into the center of the image field as soon as possible with as low effort as possible. This is commonly implemented in a way resembling the cascade control structure in standard servosystems: the pointing&tracking controller is expected to set a reference rate vector $\dot{s}_{ref}(t)$, which guarantees that the error $e(t)$ (in the image plane) 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

$$\dot{s}(t) = As(t) \tag{13}$$

where $A$ has nonreal eigenvalues. The simplest solution can be obtained by restricting $A$ to $A = -\alpha I$ for some real nonpositive $\alpha$ and then

$$\dot{s}(t) = -\alpha s(t) \tag{14}$$

The larger the $\alpha$, the faster the error in the image plane goes to zero. Now, how can we make this error evolve as in (14)? Noting that $\dot{s}(t)$ is related to the camera inertial angular velocities according to (12), we can conclude that asymptotically stable image error is guaranteed if the camera inertial velocities follow the reference value

$$\omega_C^{ref} = \frac{\lambda}{z\left(\lambda^2 + u^2 + w^2\right)} \begin{bmatrix} -\alpha wz - \lambda v_y + wv_z + ku \\ \alpha uz + \lambda v_x - uv_z + kw \\ -\alpha wv_x + u\lambda v_y + k\lambda \end{bmatrix} \tag{15}$$

It is not clear at this moment whether and how such a rotation rate of the camera can be established by the two motors. Read on. The free parameter $k$ can help to pick such a desired inertial velocity vector 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 rotation rate of the camera, what remains is to express it via simple (constant) transformation $R_C^E$ into the inner gimbal frame, and then the task for the inertial angular rate control system is then to keep this velocity by commanding the two motors

$$\omega_E^{ref} = R_C^E \omega_C^{ref} = \frac{\lambda}{z\left(\lambda^2 + u^2 + w^2\right)} \begin{bmatrix} -\alpha wv_x + u\lambda v_y + k\lambda \\ \alpha wz + \lambda v_y - wv_z - ku \\ -\alpha uz - \lambda v_x + uv_z - kw \end{bmatrix} \tag{16}$$

It is important to keep track of the corresponding frames. The resulting $\omega_E^{ref}$ should be fully labeled as $\omega_{R,E}^{E,ref}$ because it gives required inertial rotation rate of the inner gimbal as measured in the inner gimbal frame by the three attached gyros.

Now comes the key part. Having just two motors, one can hardly have ambitions to command all the three inertial angular velocities. But the free scalar parameter $k$ can be used to pick a specific triple of inertial angular velocities that requires no change against the current value of $\omega_{Ex}$ (angular rate of the camera around its optical axis which is not directly controlled). The value of $\omega_{Ex}$ must be available to the controller, though. Solving (16) for the value of $k$ guaranteeing that the $x$ component of the vector on the right is equal to the measured $\omega_{Ex}$ gives (introducing back the explicit notation of dependence on time to make it clear what is a parameter and what is a time-dependent signal)

$$k(t) = \frac{\lambda(w(t)v_x(t) - u(t)v_y(t)) + \omega_{Ex}(t)z(t)\left(\lambda^2 + u^2(t) + w^2(t)\right)}{\lambda^2} \tag{17}$$

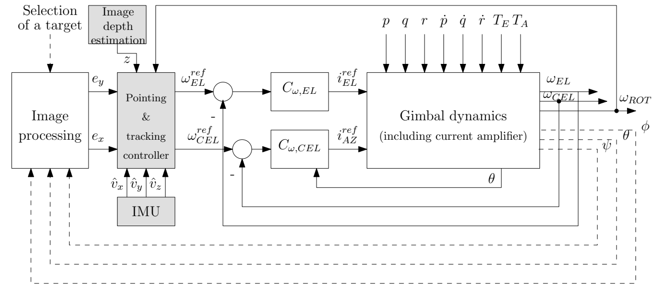



Fig. 7. Full feedback system with a computer vision tracker and pointing controller aware of the rotation about the optical axis and the translational motion.

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

$$\begin{aligned}\omega_{Ey}^{ref} &= \frac{\alpha w\lambda^2 + \omega_{Ex}(-u\lambda^2 - uw^2 - u^3)}{\lambda(\lambda^2 + u^2 + w^2)} \\ &\quad + \frac{\lambda v_y - \lambda wv_z - uwv_x + u^2 v_y}{z(\lambda^2 + u^2 + w^2)} \end{aligned} \tag{18}$$

$$\begin{aligned}\omega_{Ez}^{ref} &= -\frac{\alpha u\lambda^2 + \omega_{Ex}(w\lambda^2 + w^3 + wu^2)}{\lambda(\lambda^2 + u^2 + w^2)} \\ &\quad - \frac{\lambda^2 v_x - \lambda uv_z + w^2 v_x - wuv_y}{z(\lambda^2 + u^2 + w^2)} \end{aligned} \tag{19}$$

### C. Structure of the pointing&tracking controller

The pointing&tracking controller keeps the decoupled structure (two separate pointing&tracking controllers), which is plausible from a tuning point of view. Each of the two controllers accepts not only their respective "measured" position error, that is $u$ and $w$ but also:

1) the component of vector $\omega_E$ describing the 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 (hence $v_z$ described how fast the aircraft 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 $\lambda$. An upgrade of the naive scheme propose in Fig.4 can thus be seen in Fig.7. The key challenge in implementing this controller is in providing the controller with the extra measurement and/or estimates $v_x$, $v_y$, $v_z$ and $z$. These could be attacked using inertial measurement unit (IMU) in combination with a laser rangefinder and possibly also in combination with a computer vision system. For instance, the depth $z$ and the "towards the object" velocity $v_z$ can perhaps be 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 [6]).

## V. Comparison of the two controllers

The observed object is initially assumed to be found in the lower right corner of the monitor and responses of the two controllers were simulated. The control goal is to bring the observed object into the center of the field of view. The inner (inertial) loops are assumed to work perfectly, that is, the required (reference) values of the two inertial angular rates were assumed to be guaranteed by the corresponding rate controllers. No translation between the carrier and the observed object was assumed (this assumption could be omitted if the controller included both parts from (18) but then an inertial estimation system would have to be devised). The only physical parameters are those of the optics: focal length $\lambda = 42mm$ and resolution of the camera CCD chip is $640 \times 480$ pixels ($u$ and $w$ were scaled to pixels for visualization purposes). Two practical considerations were implemented in the simulations:

- The computer vision algorithm devours some computational time, hence the information about $v$ and $w$ is only updated with the rate from 0.5 to 10Hz, depending on the complexity of the algorithm. For simulation we choose 2Hz. In addition, the data is always available to the pointing&tracking controller with one sampling period delay. The sampling rate of the inner (inertial) loop is two order of magnitude higher (250Hz, the maximum provided by the inertial angular rate sensors).
- Both the elevation and the azimuth motors are constrained in their maximum angular rate to 60 deg/sec. Hence, the reference inertial angular rates produced by the pointing&tracking controller must be subject to saturation as well. The reference rate $\omega_{Ez}$ must be constrained more severely as the camera elevation $\theta$ is approaching $90^o$ through $\omega_{Ez} = cos(\theta)\omega_{Az}$.

Both practical consideration limit the achievable performance of both presented control schemes. The advanced algorithm takes neither of them into consideration but simulations hint that being provided with an extra information (the inertial angular rate of the camera around its optical axis), it copes with these practical issues better then the decoupled (naive) one. The simulations were run for two initial situations: the easy one $\theta(0) = -9^o$ and the tough one $\theta(0) = -72^o$. For the easy situation, it is possible to tune both controllers such that their response is nearly identical. Hence, only the simulations for the more tilted camera are visualized here only. Simulations were run both with the saturation included in the controllers and without any provision for the constraints on achievable inertial rates around the two camera body axis. Results are visualised in Fig.8 and 9. It is apparent that when the camera is tilted more towards the ground ($\theta$ approaching $-90^o$), the new controller takes much advantage of the extra information (inertial rate about the optical axis) provided by the gyro. This information is provided at the rate up to two orders of magnitude faster then pointing&tracking error provided by the computer vision system. The impact on shape of commanded rates between arrivals of two samples from the computer vision is easily noticeable in Fig.9.

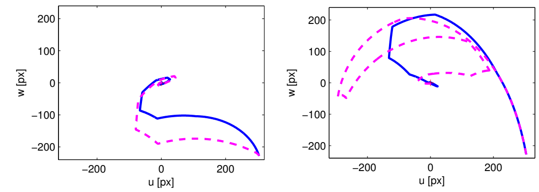



Fig. 8. Image features for the initial camera tilt $\theta(0) = -72^o$. The new proposed algorithm on the left, the decoupled (naive) control on the right. The full (blue) trajectory for a controller with a saturation, the dashed (magenta) trajectory for the (unrealistic) controller with no limits on its outputs. The coordinates in the image plane scaled to pixels, but not inverted and shifted as suggested by (4).

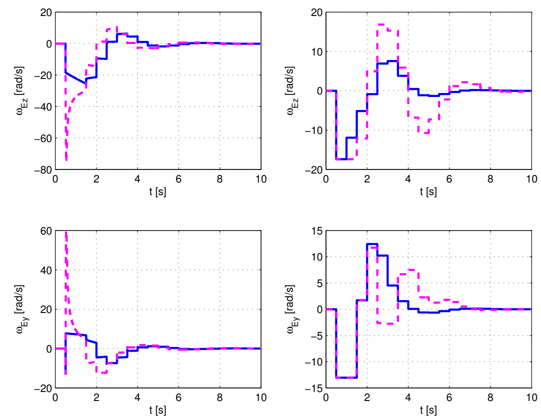



Fig. 9. Reference rates (the outputs of the pointing&tracking controller) for the conditions of Fig.8. The new proposed algorithm on the left, the decoupled (naive) control on the right.

## VI. Conclusions

The paper presented a rigorous analysis of how to implement a pointing&tracking controller into a cascaded structure with the inertial rate stabilization loop already implemented and closed. The proposed scheme has not yet been implemented on the benchmark system available to the authors.

## VII. Acknowledgments

The authors gratefully acknowledge fruitful discussions with J. M. Hilkert and D. Kostic.

## References

[1] M. Masten and L. Stockum, Eds., *Selected Papers on Precision Stabilization and Tracking Systems for Acquisition, Pointing and Control Applications*, ser. SPIE Milestone Series, vol. MS 123. SPIE, 1996.
[2] M. Masten, "Inertially stabilized platforms for optical imaging systems: Tracking dynamic dynamic targets with mobile sensors," *IEEE Control Systems Magazine*, vol. 28, pp. 47–64, Feb. 2008.
[3] J. M. Hilkert, "Inertially stabilized platform technology: concepts and principles," *IEEE Control Systems Magazine*, vol. 28, pp. 26–46, Feb. 2008.
[4] J. Debruin, "Control systems for mobile satcom antennas," *Control Systems Magazine, IEEE*, vol. 28, pp. 86–101, 2008.
[5] J. Osborne, G. Hicks, and R. Fuentes, "Global analysis of the double-gimbal mechanism," *Control Systems Magazine, IEEE*, vol. 28, pp. 44–64, 2008.
[6] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. Wiley, 2006.
[7] A. Rue, "Precision stabilization systems," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. AES-10, pp. 34–42, 1974.