

Dynamic Visual Servoing with Image Moments for a Quadrotor Using a Virtual Spring Approach

Ryuta Ozawa and François Chaumette,

Abstract—This paper presents an image-based visual servoing for controlling the position and orientation of a quadrotor using a fixed downward camera observing landmarks on the level ground. In the proposed method, the negative feedback to the image moments is used to control the vertical motion and rotation around the roll axis. On the other hand, the negative feedback cannot be used to control the horizontal motion due to under-actuation of a quadrotor. Thus, a novel control method is introduced to control the horizontal motion. Simulations are presented to validate the proposed method.

I. INTRODUCTION

Control of unmanned aerial vehicles (UAV) is an active research topic with civilian applications including search and rescue, wild fire monitoring, traffic monitoring, pipeline patrol and so on [1]. Attitude control [2] and pose control [3] of UAVs have become increasingly important to be applied to those applications, gyroscopes are used for attitude control and inertial sensors or GPSs are often used to sense the current pose. However, it is difficult to obtain accurate current position using those sensors [4]. A camera is a candidate to know the current state, and visual servoing is a powerful tool to control the pose using a camera. Visual servoing can be divided into two main classes [5]; position-based visual servoing (PBVS) and image-based visual servoing (IBVS). PBVS requires an accurate kinematic model to estimate

the pose and is sensitive to image measurement errors and kinematic model errors. On the other hand, IBVS uses image features directly, and is less sensitive to those errors than PBVS, though it has its own problems. It is generally more difficult to build controllers with IBVS than those with PBVS due to the complexity of kinematics between control variables and image features.

Much of the existing researches for visual servoing of UAVs have used PBVS [6], [7], [8]. An image-based PID controller using a stationary camera has been proposed to control the pose of a helicopter under the assumption the roll and pitch motion can be neglected [9]. The desired pose is calculated using the interaction matrix, and a PID controller is used to converge to this pose. An IBVS of an under-actuated system with an eye-in-hand camera has been designed with passivity-like properties in spherical image space and the measurement of translational velocity [10]. Based on the passivity-like properties, a new visual error term has been considered to design a dynamic controller [11]. Both the controllers stabilize the position with backstepping method, but an extra controller is needed to control the yaw angle. Bourquardez et al. [12] have compared several kinematic IBVS algorithms experimentally using a quadrotor. They found that perspective image moments are useful to design a IBVS translational control law, but the stability problem is still open. On the other hand, in the case of the spherical image moment, the stability problem has been solved but the behavior on the vertical axis is not good.

This paper proposes an image based visual servoing for controlling the pose of a quadrotor, which is a UAV with four fixed propellers and is often used for the UAV research [6], [8], [13], [11]. We assume that a camera is fixed to the quadrotor and is directed downward to observe landmarks on the level ground [14]. Perspective image moments are then computed from these landmarks. In the proposed method, the transposed Jacobian control [15] is used to control the pose of the quadrotor. It can be applied to control the translation along and the rotation around the vertical axis. However, it cannot be applied to control the translations in the horizontal plane due to the under-actuation of a quadrotor. To control these translations, positive image feature feedback method with virtual spring is proposed based on the kinematic properties of a quadrotor. Some benefits of the proposed approach are that it is compact, requires small amounts of computation and sensor information, does not need a supplementary method for controlling the yaw rotation, and does not neglect the rotational dynamics of the quadrotor. Previous approaches use the knowledge of the translational

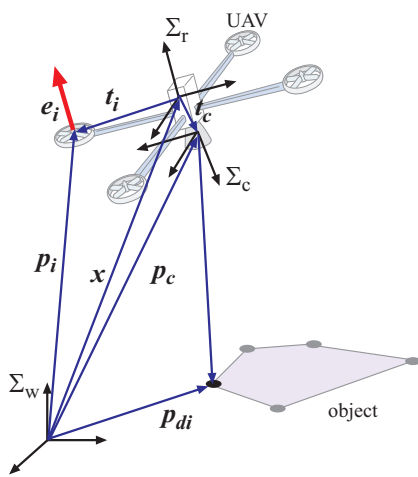


Fig. 1. Coordinates of a quadrotor.

R. Ozawa is with the Department of Robotics, Ritsumeikan University, Shiga, 525-8577 JAPAN e-mail: ryuta@se.ritsumei.ac.jp.

F. Chaumette is with INRIA Rennes-Bretagne Atlantique and IRISA, 35 042 Rennes, France e-mail: Francois.Chaumette@irisa.fr.

velocity [6]- [11] and the rotational matrix between the base and UAV frames, which includes rotation around yaw axis [6], [7], [10], [11]. On the other hand, the proposed approach only needs the knowledge of the angular velocity and the unit gravity vector, which does not contain the yaw rotation, in the body frame, and does not require any translational velocity.

Simulation results are finally presented to validate the effectiveness and robustness of the controller.

II. MODELING OF A QUADROTOR

First, we consider the kinematics of a quadrotor as shown in Fig. 1. Let $\mathbf{x} = (x, y, z)$ and $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$ be the position and the orientation of the quadrotor in a static reference frame Σ_w , and ${}^w\mathbf{R}_r(\boldsymbol{\theta})$ be the rotational matrix from Σ_w to Σ_r , referred to as roll, pitch, and yaw. Then, the velocity of the quadrotor ${}^r\mathbf{v}$ can be described as ${}^r\mathbf{v} = {}^r\mathbf{R}_w(\boldsymbol{\theta})\dot{\mathbf{x}}$, and the angular velocity of the quadrotor is given as

$$\begin{aligned} {}^r\boldsymbol{\Omega} &= [{}^r\boldsymbol{\omega}]_{\times} = {}^r\mathbf{R}_w(\boldsymbol{\theta}) {}^w\dot{\mathbf{R}}_r(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \\ &= -{}^r\dot{\mathbf{R}}_w(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) {}^r\mathbf{R}_w^T(\boldsymbol{\theta}). \end{aligned} \quad (1)$$

The vector expression of the angular velocity (1) is given as

$$\begin{aligned} {}^r\boldsymbol{\omega} &= \mathbf{G}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \\ \text{with } \mathbf{G}(\boldsymbol{\theta}) &= \begin{bmatrix} 1 & 0 & \sin\theta_2 \\ 0 & \cos\theta_1 & -\sin\theta_1 \cos\theta_2 \\ 0 & \sin\theta_1 & \cos\theta_1 \cos\theta_2 \end{bmatrix}. \end{aligned} \quad (2)$$

Thus,

$${}^r\dot{\mathbf{z}} = \begin{bmatrix} {}^r\mathbf{v} \\ {}^r\boldsymbol{\omega} \end{bmatrix} = \mathbf{V}\dot{\mathbf{z}}, \text{ where } \mathbf{V} = \begin{bmatrix} {}^r\mathbf{R}_w & \mathbf{0} \\ \mathbf{0} & \mathbf{G}(\boldsymbol{\theta}) \end{bmatrix}, \quad (3)$$

and $\dot{\mathbf{z}} = (\dot{\mathbf{x}}, \dot{\boldsymbol{\theta}})$. The Lagrangian of the quadrotor is given as follows:

$$L = \frac{1}{2}m\|{}^r\mathbf{v}\|^2 + \frac{1}{2}{}^r\boldsymbol{\omega}^T \hat{\mathbf{I}} {}^r\boldsymbol{\omega} + m\mathbf{g}^T \mathbf{x}. \quad (4)$$

where m is the mass of the quadrotor, $\hat{\mathbf{I}}$ is the inertia moment, $\mathbf{g} = (0, 0, -g_r)$ is the gravitational vector. Then, by solving the variational problem using Eq. (4) [16] and by adding damping terms, the dynamical equation of the quadrotor is given as follows:

$$\begin{aligned} \begin{bmatrix} m\mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{H}(\boldsymbol{\theta}) \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{x}} \\ \ddot{\boldsymbol{\theta}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \frac{1}{2}\frac{d}{dt}\mathbf{H}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} + \mathbf{S}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} \end{bmatrix} \\ - \begin{bmatrix} m\mathbf{g} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix} &= \mathbf{V}^T \mathbf{A}^T \mathbf{u}, \end{aligned} \quad (5)$$

where $\mathbf{H}(\boldsymbol{\theta}) = \mathbf{G}^T(\boldsymbol{\theta})\hat{\mathbf{I}}\mathbf{G}(\boldsymbol{\theta})$, \mathbf{u} is the force generated by the propellers, $\mathbf{S}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ is the skew-symmetric matrix, \mathbf{b}_1 and \mathbf{b}_2 are the translational and rotational friction components, which are modeled as

$$\mathbf{b}_1 = d_1 \begin{bmatrix} \|\dot{x}\|\dot{x} \\ \|\dot{y}\|\dot{y} \\ \|\dot{z}\|\dot{z} \end{bmatrix}, \text{ and } \mathbf{b}_2 = d_2 \begin{bmatrix} \|\dot{\theta}_1\|\dot{\theta}_1 \\ \|\dot{\theta}_2\|\dot{\theta}_2 \\ \|\dot{\theta}_3\|\dot{\theta}_3 \end{bmatrix}, \quad (6)$$

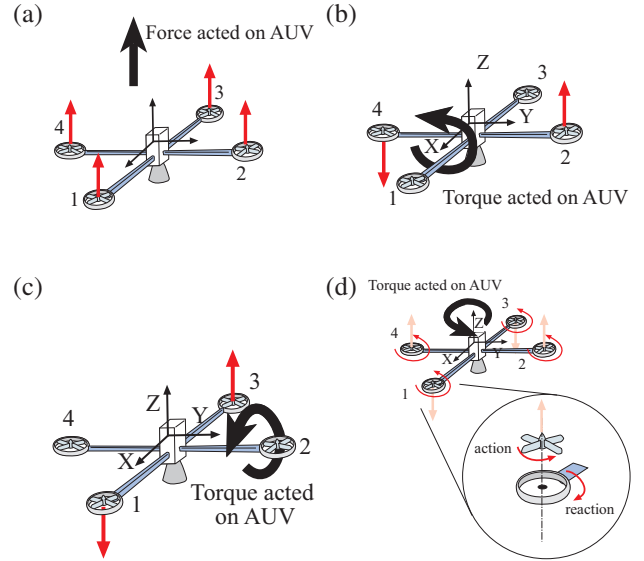


Fig. 2. The effects of the propellers on the motion of the quadrotor. The red arrows indicate increment of force/torque generated by the propellers and acted on the quadrotor from the equilibrium. The black thick arrows indicate the net force/torque acted on the quadrotor. The internal forces are eliminated in this figure. (a) Translational motion in z direction. (b) Rotational motion around x axis. (c) Rotational motion around y axis. (d) Rotational motion around z axis. As shown in the circle, the action to rotate the propeller generates the reaction torque, which is used to rotate the quadrotor around z axis.

d_1 and d_2 are the damping coefficients, \mathbf{A} is the transmission matrix defined by

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 & -\ell & \kappa \\ 0 & 0 & 1 & \ell & 0 & -\kappa \\ 0 & 0 & 1 & 0 & \ell & \kappa \\ 0 & 0 & 1 & -\ell & 0 & -\kappa \end{bmatrix}, \quad (7)$$

ℓ is the length of the moment arm, and κ is a ratio between the translational force to the rotational torque. The rank of the matrix \mathbf{A} is four so that the system is under-actuated.

It is important to understand the feature of the driving force for designing a controller for the quadrotor. Fig. 2 shows the explanation of the net force/torque acted on the quadrotor. The translational movement in the z axis is generated as shown in Fig. 2 (a). The summation of all the propellers is equivalent to the translational force. The rotational motions around the x , y and z axes are shown in Fig. 2 (b), (c) and (d) respectively. The rotation around the x and y axes are generated by the difference between two propellers. Each propeller generates the reaction torque to the quadrotor, and the summation of those reactive torque generates the rotation around the z axis.

Thus, the dynamics of quadrotor given by (5) can be rewritten as follows (see, e.g., [11]),

$$m\ddot{\mathbf{x}} + \mathbf{b}_1 - m\mathbf{g} = {}^w\mathbf{R}_r {}^r\hat{\mathbf{e}}\tau_1 \quad (8)$$

$$\mathbf{H}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \left\{ \frac{1}{2}\dot{\mathbf{H}}(\boldsymbol{\theta}) + \mathbf{S}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \right\} \dot{\boldsymbol{\theta}} + \mathbf{b}_2 = \mathbf{G}^T(\boldsymbol{\theta})\boldsymbol{\tau}_2, \quad (9)$$

where ${}^r\hat{\mathbf{e}} = (0, 0, 1)$ is the unit vector of the direction of the translational driving force, τ_1 is the translational force

for quadrotor and τ_2 is the rotational torque w.r.t. Σ_r .

From Eq. (2), note that the dynamical equations (8) and (9) are effective in the case that θ_2 does not cross $\pm\pi/2$ (rad). The quaternion representation [2] is useful to avoid this singularity. However, in the case that $\theta_2 = \pm\pi/2$, the quadrotor lies in the vertical plane and cannot be controlled anymore. Thus, in this paper, we treat the behavior of the quadrotor only in the region where $-\pi/2 < \theta_2 < \pi/2$.

III. IMAGE SPACE

First we consider the general framework of image errors, and interaction matrices [5]. Let s_i and s_i^* be the current and the desired image feature of the i th landmark, $\Delta s_i = s_i - s_i^*$ be the image error, ${}^c\dot{z} = ({}^c\mathbf{v}, {}^c\boldsymbol{\omega})$ be the camera velocity in the camera coordinates Σ_c . Then,

$$\Delta\dot{s}_i = L_{s_i} {}^c\dot{z} = [L_{s_i1} \quad L_{s_i2}] {}^c\dot{z}, \quad (10)$$

where L_{s_i} is the interaction matrix. The camera velocity is given as

$${}^c\dot{z} = {}^cS_r \mathbf{V}\dot{z}, \quad (11)$$

where

$${}^cS_r = \begin{bmatrix} {}^cR_r & -{}^cR_r [{}^r t_c]_{\times} \\ \mathbf{0} & {}^cR_r \end{bmatrix}, \quad (12)$$

${}^cR_r = \text{diag}(1, -1, -1)$ and ${}^r t_c$ is the position vector of the camera from the origin of Σ_r w.r.t. Σ_r . Thus, from Eqs. (10), (11) and (12),

$$\Delta\dot{s}_i = P_i \mathbf{V}\dot{z} = [P_{i1} \quad P_{i2}] \mathbf{V}\dot{z} \quad (13)$$

where

$$P_{i1} = L_{s_i1} {}^cR_r, \text{ and } P_{i2} = -L_{s_i1} {}^cR_r [{}^r t_c]_{\times} + L_{s_i2} {}^cR_r.$$

We now give the interaction matrices of perspective image moments as shown in [14]. As will be described later, the translational parts of these matrices are independent each others, and this property is useful to design our new controller. Let n points form a landmark on the level ground. Then, the moments are defined by:

$$m_{ij} = \sum_{k=1}^n x_k^i y_k^j, \quad (14)$$

where (x_i, y_i) is the position of the i th landmark in image space. Let (x_g, y_g) be the center of gravity of the landmarks in the image. The centered moments are defined by:

$$\mu_{ij} = \sum_{k=1}^n (x_k - x_g)^i (y_k - y_g)^j, \quad (15)$$

where $x_g = m_{10}/n$ and $y_g = m_{01}/n$, $m_{00} = n$. Let a be defined by

$$a = \mu_{20} + \mu_{02}. \quad (16)$$

As the visual features to control the translation, we choose $s_n = (x_n, y_n, a_n)$ which are given by

$$x_n = a_n x_g, \quad y_n = a_n y_g, \quad a_n = Z^* \sqrt{\frac{a^*}{a}}, \quad (17)$$

where a^* and Z^* are the value of a and the depth when the quadrotor is in the desired configuration. Then,

$$\Delta\dot{s}_n = [P_{s_n1} \quad P_{s_n2}] \mathbf{V}\dot{z}. \quad (18)$$

where, when the points are coplanar and parallel to the image plane [14],

$$P_{s_n1} = -{}^cR_r \text{ and } P_{s_n2} = {}^cR_r [{}^r t_c]_{\times} + L_{s_n2} {}^cR_r,$$

$$L_{s_n2} = \begin{bmatrix} L_{x_n2} \\ L_{y_n2} \\ L_{a_n2} \end{bmatrix} = \begin{bmatrix} a_n \epsilon_{11} & -a_n(1 + \epsilon_{12}) & y_n \\ a_n(1 + \epsilon_{21}) & -a_n \epsilon_{22} & -x_n \\ -a_n \epsilon_{31} & a_n \epsilon_{32} & 0 \end{bmatrix},$$

with

$$\begin{cases} \epsilon_{11} &= n_{11} + x_g(y_g - \epsilon_{31}), \\ \epsilon_{12} &= n_{20} + x_g(x_g - \epsilon_{32}), \\ \epsilon_{21} &= n_{02} + y_g(y_g - \epsilon_{31}), \\ \epsilon_{22} &= n_{11} + y_g(x_g - \epsilon_{32}), \\ \epsilon_{31} &= y_g + (y_g \mu_{02} + x_g \mu_{11} + \mu_{21} + \mu_{03})/a, \\ \epsilon_{32} &= x_g + (x_g \mu_{20} + y_g \mu_{11} + \mu_{12} + \mu_{30})/a, \end{cases}$$

and $n_{ij} = \mu_{ij}/m_{00}$. ${}^cR_r = \text{diag}(1, -1, -1)$, thus we can see that these translational motions are independent each other. It will be used to control the translations in the horizontal plane.

The third element of Δs_n , Δa_n , is important to control the translation on the vertical axis. For Δa_n , the interaction matrix is given as follows,

$$P_{a_n} = [P_{a_n1} \quad P_{a_n2}] = \begin{bmatrix} {}^r \hat{e}^T & {}^r \hat{e}^T [{}^r t_c]_{\times} + L_{a_n2} {}^cR_r \end{bmatrix}. \quad (19)$$

Let α be the object orientation angle defined as

$$\alpha = \frac{1}{2} \tan^{-1} \left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right). \quad (20)$$

The time derivative of $\Delta\alpha = \alpha - \alpha^*$ is

$$\Delta\dot{\alpha} = P_{\alpha} \mathbf{V}\dot{z}, \quad (21)$$

when the points are coplanar and parallel to the image plane, where,

$$P_{\alpha} = [P_{\alpha1} \quad P_{\alpha2}] = [\mathbf{0} \quad L_{\alpha2} {}^cR_r], \quad (22)$$

with

$$L_{\alpha2} = [\alpha_{wx} \quad \alpha_{wy} \quad -1], \quad (23)$$

and

$$\begin{aligned} \alpha_{wx} &= (\beta[\mu_{12}(\mu_{20} - \mu_{02}) + \mu_{11}(\mu_{03} - \mu_{21})] \\ &\quad + \gamma x_g[\mu_{02}(\mu_{20} - \mu_{02}) - 2\mu_{11}^2] + \gamma y_g \mu_{11}[\mu_{20} - \mu_{02}])/d, \\ \alpha_{wy} &= (\beta[\mu_{21}(\mu_{02} - \mu_{20}) + \mu_{11}(\mu_{30} - \mu_{12})] \\ &\quad + \gamma x_g \mu_{11}[\mu_{20} + \mu_{02}] + \gamma[\mu_{20}(\mu_{02} - \mu_{20}) - 2\mu_{11}^2])/d, \\ d &= (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2, \end{aligned}$$

with $\beta = 4$ and $\gamma = 2$. $P_{\alpha1}$ is the zero matrix, and we can see that $\dot{\alpha}$ is independent from the translational motion. In the next section, we design a controller based on the visual features selected, that is s_n and α .

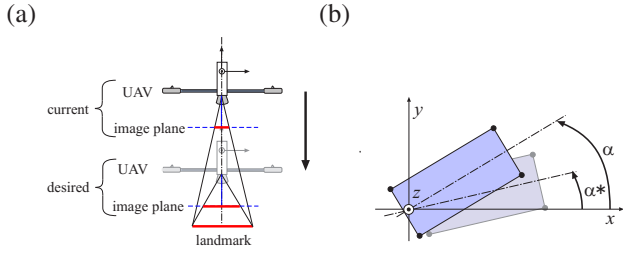


Fig. 3. Translational and rotational motions of the quadrotor in the vertical plane.(a) translational motion, (b) rotational motion.

IV. CONTROLLER DESIGN FOR A QUADROTOR

The control objective is to make a quadrotor stabilize the point such that $(s_n, \alpha, \theta_1, \theta_2) = (s_n^*, \alpha^*, 0, 0)$, where θ_1 and θ_2 are roll and pitch angles of the quadrotor. The desired visual features s_n^* and α^* can be chosen to reach any particular configuration of the quadrotor such that it is parallel to the level ground.

To control the quadrotor, we need to sense the unit vector of the gravity, which contains the roll and pitch angles of the quadrotor, while previous approaches need to sense ${}^w R_r$ [6]-[11], which also requires the yaw angles. It is defined as

$${}^r \mathbf{y}_{\text{sen}} = {}^r \mathbf{R}_w \mathbf{e}_g, \quad (24)$$

where $\mathbf{e}_g = (0, 0, -1)$ is the gravitational unit vector w.r.t. Σ_w . The time derivative is calculated using Eqs. (1) and (2) as

$${}^r \dot{\mathbf{y}}_{\text{sen}} = {}^r \mathbf{y}_{\text{sen}} \times \mathbf{G}(\boldsymbol{\theta}) \dot{\boldsymbol{\theta}}. \quad (25)$$

To control the translational and rotational motions in and around the vertical axis in Fig. 3, we consider the case where the quadrotor is parallel to the level ground above the target. We will see in the simulation results that the controller is also effective when it is not the case.

Here, we design a controller based on a task-space PD feedback with gravity compensation [15] to control the translational and rotational motion on the z axis using the interaction matrices (19) and (22) as follows,

$$\tau_1 = -\frac{mg_r}{{}^r \mathbf{e}^T {}^r \mathbf{y}_{\text{sen}}} - \gamma_1 \Delta a_n, \quad (26)$$

$$\tau_2 = \mathbf{v} - \gamma_1 \mathbf{P}_{\alpha_2}^T \Delta a_n - \gamma_2 \mathbf{P}_{\alpha_2}^T \Delta \alpha - \gamma_3 \mathbf{G}(\boldsymbol{\theta}) \dot{\boldsymbol{\theta}}. \quad (27)$$

The first term in τ_1 is for the gravitational compensation. The second terms of τ_1 and τ_2 are the visual feedback terms to eliminate the vertical error Δa_n . As shown in Eq. (18), Δa_n affects only in the z direction of the translational motion. The third term in τ_2 is the image feedback of $\Delta \alpha$. It does not affect τ_1 because the first three columns of (22) are zeros. The last term in τ_2 is the rotational damping. \mathbf{v} in τ_2 is an additional control input, and we will discuss it later.

Next, we consider the translational motion in the horizontal plane. The sideway motion cannot be generated using the

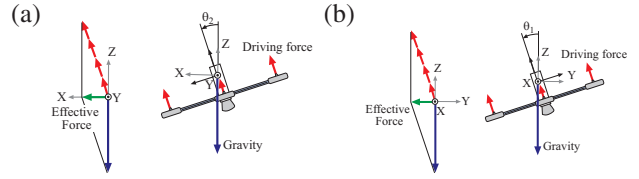


Fig. 4. The effect of Rotation for translational motion in horizontal plane. (a) x direction, (b) y direction

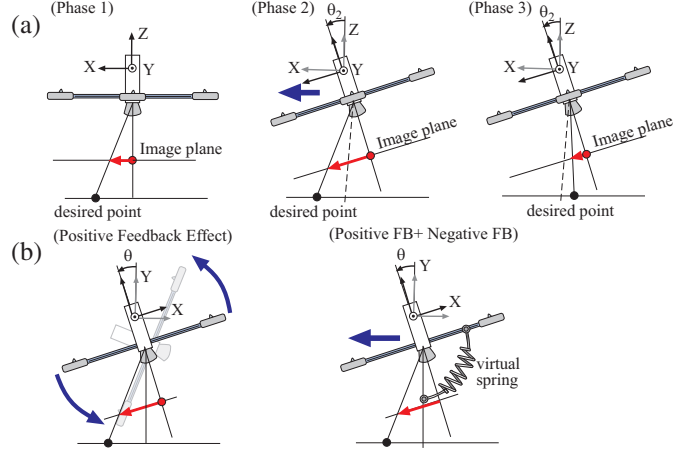


Fig. 5. Effect of Feedback for translational motion in horizontal plane. (a) Image error and the orientation of the quadrotor. The error can be eliminated by moving on the left, and the quadrotor must be inclined to move on the left. (b) Positive visual feedback effect. It provides translational force, but the quadrotor over-rotates and the system become unstable. A virtual spring approach prevents from turning the UAV over, and is effective to eliminate the position errors.

image feedback directly due to under-actuation of a quadrotor. As shown in Fig. 4 (a), when the quadrotor inclines around the y axis and generates the gravity compensation force, the vertical force is canceled and the translational force in the x direction is generated implicitly. Then, the quadrotor can move in the x direction. (For y translational force, see Fig. 4 (b).) We now consider the relationship between the image error and the translational motion. As shown in Fig. 5(a), the quadrotor would like to move to the left for eliminating the image error (Phase 1). The quadrotor inclines to generate the positive force in x direction, and the image error becomes larger (Phase 2). Then, the quadrotor moves till the image error is eliminated (Phase 3). Thus, to generate the translational force in the x direction, the image error must be fed back positively as shown in Fig. 5 (b) left. However, the positive visual feedback makes the system unstable, and we must prevent the quadrotor from giving the overthrow so that a virtual spring is added, as shown in Fig. 5 (b) right. Thus,

$$\mathbf{v} = \mathbf{v}_1 + \mathbf{v}_2 \quad (28)$$

is composed of the virtual spring force \mathbf{v}_1 and the positive feedback \mathbf{v}_2 .

To realize the virtual spring effect, we consider the following artificial potential.

$$U_k = \frac{1}{2} \Delta {}^r \mathbf{y}_{\text{sen}}^T \mathbf{K} \Delta {}^r \mathbf{y}_{\text{sen}}, \quad (29)$$

where

$$\Delta {}^r \mathbf{y}_{\text{sen}} = {}^r \mathbf{y}_{\text{sen}} - \mathbf{e}_g. \quad (30)$$

Then, from Eq. (25),

$$\mathbf{v}_1 = -\frac{\partial U_k}{\partial \boldsymbol{\theta}} = \mathbf{G}(\boldsymbol{\theta})^T [{}^r \mathbf{y}_{\text{sen}}]_{\times} \mathbf{K} \Delta {}^r \mathbf{y}_{\text{sen}}. \quad (31)$$

To design the positive feedback of the image error, we consider the static equilibrium in Eq. (8) as follows,

$$m\mathbf{g} - {}^w \mathbf{R}_r {}^r \hat{\mathbf{e}} \frac{mg_r}{{}^r \hat{\mathbf{e}}^T {}^r \mathbf{y}_{\text{sen}}} - \gamma_1 {}^w \mathbf{R}_r {}^r \hat{\mathbf{e}} \Delta a_n \quad (32)$$

$$= \mathbf{f}_g - \gamma_1 \Delta a_n {}^w \mathbf{R}_r \mathbf{P}_{a_n 1}^T = \mathbf{0}, \quad (33)$$

where \mathbf{f}_g is the effective force in Fig. 4 and is defined as

$$\mathbf{f}_g = mg_r {}^w \mathbf{R}_r \left\{ {}^r \mathbf{y}_{\text{sen}} - {}^r \hat{\mathbf{e}} \frac{1}{{}^r \hat{\mathbf{e}}^T {}^r \mathbf{y}_{\text{sen}}} \right\}. \quad (34)$$

If \mathbf{f}_g is generated by a virtual force $\mathbf{h}_v = \gamma_0 \Delta \mathbf{s}_n$ in the image space as a result of the image error $\Delta \mathbf{s}_n$, then it could be expressed using the transpose of the interaction matrix as

$$\begin{bmatrix} \mathbf{f}_g \\ \mathbf{G}^T(\boldsymbol{\theta}) \mathbf{v}_{21} \end{bmatrix} = \mathbf{V}^T \mathbf{P}_{s_n}^T \gamma_0 \mathbf{h}_v = \begin{bmatrix} {}^w \mathbf{R}_r \mathbf{P}_{s_n 1}^T \\ \mathbf{G}(\boldsymbol{\theta})^T \mathbf{P}_{s_n 2}^T \end{bmatrix} \mathbf{h}_v, \quad (35)$$

where \mathbf{v}_{21} is a component of \mathbf{v}_2 . However, \mathbf{f}_g cannot be generated as a result of \mathbf{h}_v due to under-actuation of the quadrotor.

Next we consider the relationship between the rotation and the image error as shown in Fig. 5. The quadrotor inclines by the image error, and we can find the torque \mathbf{v}_{22} which is equivalent to \mathbf{h}_v . If the translational motion of the quadrotor can be neglected, then from the principle of the virtual work,

$$\delta \boldsymbol{\theta}^T \mathbf{G}^T \mathbf{v}_{22} - \delta(\Delta \mathbf{s}_n)^T \mathbf{h}_v = 0. \quad (36)$$

From the first raw of Eq. (35), \mathbf{h}_v can be calculated as

$$\begin{aligned} \mathbf{h}_v &= ({}^w \mathbf{R}_r \mathbf{P}_{s_n 1}^T)^{-1} \mathbf{f}_g = -{}^c \mathbf{R}_r {}^r \mathbf{R}_w \mathbf{f}_g \\ &= -mg_r {}^c \mathbf{R}_r \left\{ {}^r \mathbf{y}_{\text{sen}} - {}^r \hat{\mathbf{e}} \frac{1}{{}^r \hat{\mathbf{e}}^T {}^r \mathbf{y}_{\text{sen}}} \right\}. \end{aligned} \quad (37)$$

From $\mathbf{h}_v = \gamma_0 \Delta \mathbf{s}_n$, the virtual displacement of $\Delta \mathbf{s}_n$ can be calculated as

$$\delta(\Delta \mathbf{s}_n) = \delta \frac{1}{\gamma_0} \mathbf{h}_v = \frac{1}{\gamma_0} \frac{\partial \mathbf{h}_v}{\partial \boldsymbol{\theta}} \delta \boldsymbol{\theta} = \frac{1}{\gamma_0} \mathbf{H}_v \mathbf{G} \delta \boldsymbol{\theta} \quad (38)$$

where

$$\mathbf{H}_v = -mg_r {}^c \mathbf{R}_r \left\{ \mathbf{I}_3 + \frac{{}^r \hat{\mathbf{e}} {}^r \hat{\mathbf{e}}^T}{({}^r \hat{\mathbf{e}}^T {}^r \mathbf{y}_{\text{sen}})^2} \right\} [{}^r \mathbf{y}_{\text{sen}}]_{\times}. \quad (39)$$

Then, Eq. (36) can be calculated using Eqs. (37)-(39) as

$$\begin{aligned} \delta \boldsymbol{\theta}^T \mathbf{G}^T \mathbf{v}_{22} - \delta(\Delta \mathbf{s}_n)^T \mathbf{h}_v &= \delta \boldsymbol{\theta}^T \mathbf{v}_{22} - \delta \left(\frac{1}{\gamma_0} \mathbf{h}_v \right)^T \gamma_0 \Delta \mathbf{s}_n \\ &= \delta \boldsymbol{\theta}^T \mathbf{G}^T \{ \mathbf{v}_{22} - \mathbf{H}_v^T \Delta \mathbf{s}_n \} = \mathbf{0}. \end{aligned}$$

The virtual displacement $\delta \boldsymbol{\theta}$ is an arbitrary vector, and we can see that

$$\mathbf{v}_{22} = \mathbf{H}_v^T \Delta \mathbf{s}_n. \quad (40)$$

Substituting Eqs. (35) and (40) into Eq. (28), we can get

$$\mathbf{v}_2 = \mathbf{P}_{s_n}^T \mathbf{h}_v + \mathbf{H}_v^T \Delta \mathbf{s}_n. \quad (41)$$

By substituting Eq. (41) into Eq. (27), the control input for the rotational part can be calculated as follows,

$$\begin{aligned} \boldsymbol{\tau}_2 &= [{}^r \mathbf{y}_{\text{sen}}]_{\times} \mathbf{K} \Delta {}^r \mathbf{y}_{\text{sen}} + \mathbf{H}_v^T ({}^r \mathbf{y}_{\text{sen}}) \Delta \mathbf{s}_n \\ &\quad + \mathbf{P}_{s_n 2}^T \mathbf{h}_v - \gamma_1 \mathbf{P}_{a_n 2}^T \Delta a_n - \gamma_2 \mathbf{P}_{\alpha 2}^T \Delta \alpha - \gamma_3 \mathbf{G} \dot{\boldsymbol{\theta}}. \end{aligned} \quad (42)$$

The first term of $\boldsymbol{\tau}_2$ is the virtual spring term, the second and the third terms are for the positive feedback, the fourth and the fifth terms are negative feedback of Δa_n and $\Delta \alpha$, and the last term is the damping of the angular velocity, respectively. The damping term can be omitted when the quadrotor is enough damped by the natural damping $\mathbf{b}_i (i = 1, 2)$. Note that the controllers (26) and (42) use only the sensing of ${}^r \mathbf{y}_{\text{sen}}$, the landmarks and the angular velocity $\boldsymbol{\theta}$. \mathbf{G} is function of $\boldsymbol{\theta}$, but uses only the roll and the pitch angles (see Eq. (2)), which can be calculated from ${}^r \mathbf{y}_{\text{sen}}$.

We omit the stability analysis due to space limitation, but we can prove that the equilibrium point $(\Delta \mathbf{s}_n, \Delta \alpha, \theta_1, \theta_2, \dot{\mathbf{z}}) = \mathbf{0}$ of the system (8) and (9) controlled by (26) and (27) is locally asymptotically stable by the Lyapunov's direct method in the vicinity of the equilibrium point where the UAV is parallel to the image plane.

Note that the interaction matrices (18), (21), and (22) are obtained under the assumption that the visual feature points lie in the plane parallel to the image plane. The robustness of these approximations will be confirmed in the simulations.

V. SIMULATION

In this section, five selected simulations are presented to validate the effectiveness of the proposed controller using the conditions given in TABLE I. The following parameters are used for the simulations; $m = 0.90$ (kg), $\ell = 0.30$ (m), $\hat{I} = (1.20, 1.20, 2.00) \times 10^{-3}$ (kgm²), $\ell = 0.30$ (m), and $\kappa = 0.0158$ (m). The visual data is updated every 33 (ms) except the case 4, while ${}^r \mathbf{y}_{\text{sen}}$ and $\dot{\boldsymbol{\theta}}$ are measured and the control input is updated every 1 (ms). We use four feature points, which are distributed around the origin of the base frame. As shown in TABLE II, in Cases 1 to 4, the initial configuration is assigned so that the quadrotor goes from the vicinity to the above of the base frame. In Case 5, the initial configuration is assigned so that the quadrotor goes away from the above of the base frame. The differences in each configuration are shown in the image sampling time, the feedback gain γ_3 and the coefficient of the damping terms (6). Figs. 6-10 show the translation and orientation errors of each case as well as the image features error and the image points trajectory. In Case 1, a good damping is used, and the errors converge to zero around 15(s). In Case 2, less damping in the translational directions is used. Then, the system is under-damped and the convergence is retarded compared to Case 1. In Case 3, less damping in the rotational directions is used. Then, residual errors remain and the motion becomes vibrationally as in a limit cycle. This is due to the sampling time of the visual data. This motion is improved using 1(ms) image sampling time as shown in Case 4, as well as using

TABLE I
FEEDBACK GAINS USED IN THE SIMULATOR

parameter	symbol	value
feedback gain of Δa_n	γ_1	1.00
feedback gain of $\Delta \alpha$	γ_2	1.00
spring coefficient	k	40.0

TABLE II
CONTROLLER PARAMETERS

Symbol	initial Point	final Point	damping	gain	image sample time
Unit			d_1, d_2	γ_3	
			(kg/m), $\left(\frac{\text{kgm}^2}{\text{rad}^2}\right)$	$\left(\frac{\text{kgm}^2}{\text{rad}\cdot\text{s}}\right)$	(ms)
Case 1	B [†]	A	(100, 0.5)	1.5	33
Case 2	B	A	(10.0, 0.5)	1.5	33
Case 3	B	A	(100, 0.3)	0.0	
Case 4	A	B	(100, 0.3)	0.0	1
Case 5	A	B	(100, 0.5)	1.5	33

[†] Point A is that $x = (0, 0, 0.5)$ (m) and $\theta_3 = 0$ (rad), and Point B is that $x = (0.5, 0.5, 5.5)$ (m) and $\theta_3 = 0.2\pi$ (rad).

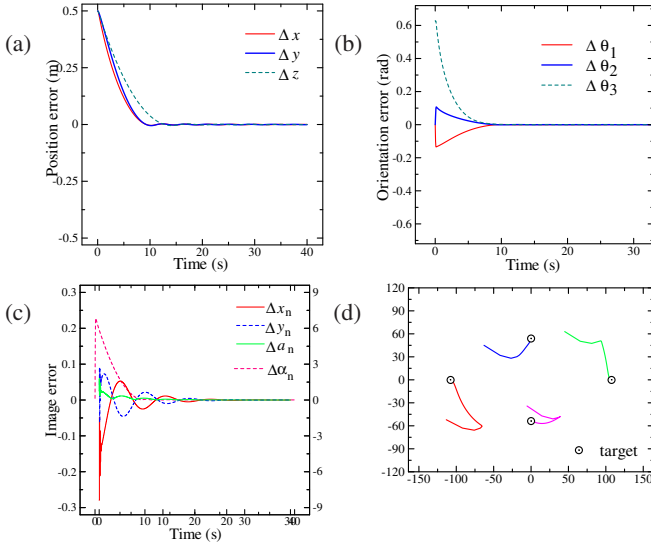


Fig. 6. Simulation results in Case 1. (a) Position error, (b) orientation error, (c) image error, and (d) image trajectories.

larger rotational damping as shown in Case 1. In Case 5, the convergence is a little bit slower than in Case 1 because of the difference of the desired configuration. We can get the better performance when the landmarks are just below the quadrotor at a desired configuration. Indeed, the interaction matrix plays role of the moment arm of the image error, and the desired configuration in Case 5 becomes more sensitive to the image error than in Case 1-4.

VI. CONCLUSION

This paper proposed an image-based visual servoing for controlling the position and the orientation of a quadrotor with a fixed camera, which points the downward direction. The transpose Jacobian control from image moments is used

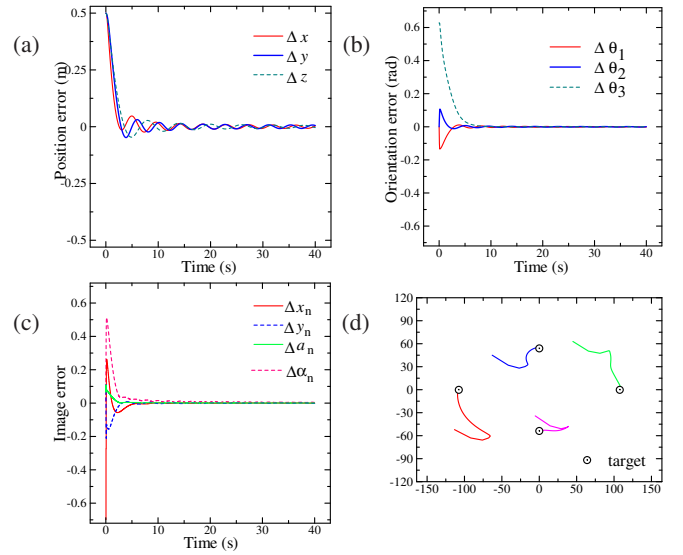


Fig. 7. Simulation results in Case 2, configured as Fig.6.

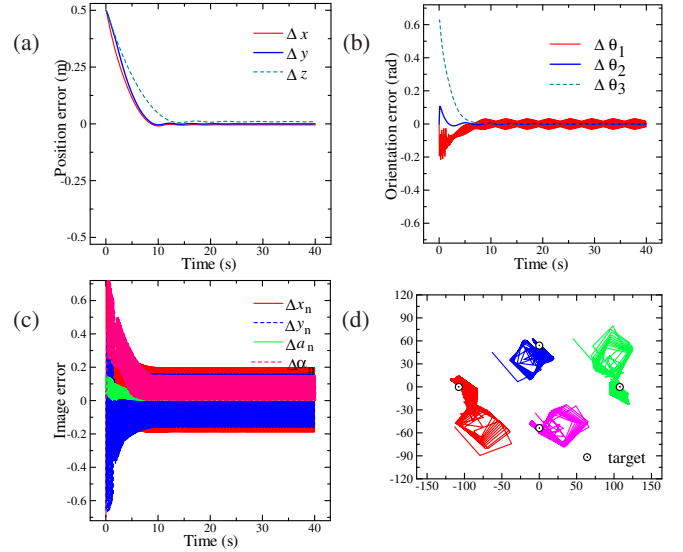


Fig. 8. Simulation results in Case 3, configured as Fig.6.

to control the translational and the rotational motions in the vertical axis. On the other hand, the sideways motion cannot be generated with this method due to the under-actuation of quadrotors. Positive feedback structure of the image errors to the sideways motion was revealed, and a virtual spring was introduced to stabilize the pose of the quadrotor. Simulations showed that this method is effective to control the pose of a quadrotor, even if the low sampling rate of the visual data may cause some residual errors. We show also that it is better to put the landmarks just below the desired pose of the quadrotor.

Future works will be devoted to realize experiments on a real platform. The current controller depends on the perspective image moment, and the controller is also undergoing to be generalized to algorithms, which can use any other image features, and camera configuration.

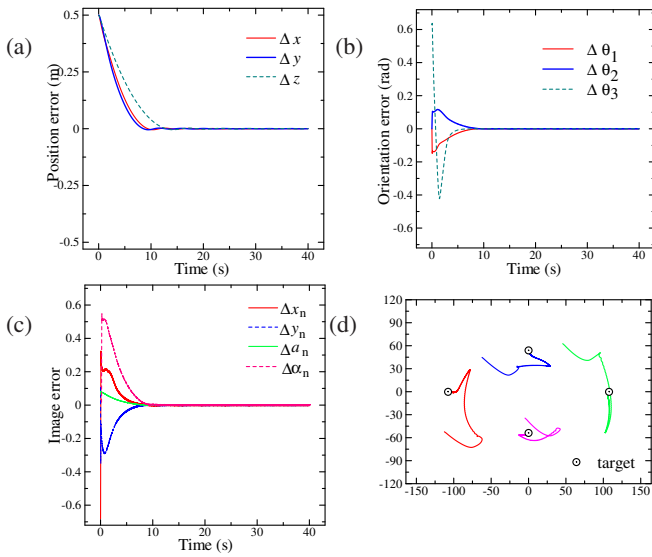


Fig. 9. Simulation results in Case 4, configured as Fig.6.

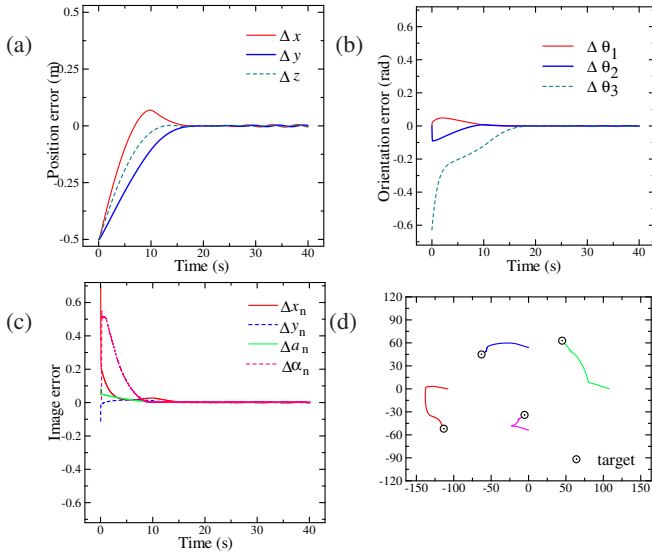


Fig. 10. Simulation results in Case 5, configured as Fig.6.

ACKNOWLEDGMENT

We would like to thank Dr. Nicolas Guenard from CEA/LIST for his valuable information of the parameters used in the simulation.

REFERENCES

- [1] M. T. DeGarmo. Issues concerning integration of unmanned aerial vehicles in civil airspace. Technical report, MITRE, Center for Advanced Aviation System Development, McLean, Virginia, 2004.
- [2] A. Tayebi and S. McGilvray. Attitude stabilization of a VTOL quadrotor aircraft. *IEEE Trans. Control Syst. Technol.*, Vol. 14, No. 3, pp. 562–571, 2006.
- [3] T. Hamel, R. Mahony, R. Lozano, and J. Ostrowski. Dynamic modelling and configuration stabilization for an X4-flyer. In *IFAC World Congress*, Barcelona, Spain, 2002.
- [4] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT press, 2006.

- [5] F. Chaumette and S. Hutchinson. Visual servo control part I: Basic approaches. *IEEE Robot. Automat. Mag.*, Vol. 18, No. 4, pp. 82–90, December 2006.
- [6] E. Altug, J. P. Ostrowski, and R. Mahony. Control of a quadrotor helicopter using visual feedback. In *Proc. of IEEE Int. Conf. on Robot. and Autom.*, pp. 72–77, Washington, DC, May 2002.
- [7] L. Mejias, S. Saripalli, P. Campoy, and G. S. Sukhatme. Visual servoing of an autonomous helicopter in urban areas using feature tracking. *J. of Field Robotics*, Vol. 23, No. 3/4, pp. 185–199, 2006.
- [8] S. Azrad, F. Kendoul, and K. Nonami. Visual servoing of quadrotor micro-air vehicle using color-based tracking algorithm. *J. of System, Design and Dynamics*, Vol. 4, No. 2, pp. 255–268, 2010.
- [9] K. Watanabe, Y. Yoshihata, Y. Iwatani, and K. Hashimoto. Image-based visual PID control of a micro helicopter using a stationary camera. *Advanced Robotics*, Vol. 22, pp. 381–393, 2008.
- [10] T. Hamel and R. Mahony. Visual servoing of an under-actuated dynamic rigid-body system: An image-based approach. *IEEE Trans. on Robotics and Automation*, Vol. 18, No. 2, pp. 187–198, 2002.
- [11] N. Guenard, T. Hamel, and R. Mahony. A practical visual servo control for an unmanned aerial vehicle. *IEEE Trans. on Robotics*, Vol. 24, No. 2, pp. 331–340, 2008.
- [12] O. Bourquardez, R. Mahony, N. Guenard, F. Chaumette, T. Hamel, and L. Eck. Image-based visual servo control of the translation kinematics of a quadrotor aerial vehicle. *IEEE Trans. on Robotics*, Vol. 25, , 2009.
- [13] S. Bouabdallah, P. Murrieri, and R. Siegwart. Design and control of an indoor micro quadrotor. In *Proc. of IEEE Int. Conf. on Robot. and Autom.*, pp. 4393–4398, New Orleans, LA, April 2004.
- [14] O. Tahri and F. Chaumette. Point-based and region-based image moments for visual servoing of planar objects. *IEEE Trans. on Robotics*, Vol. 21, No. 6, pp. 1116–1127, 2005.
- [15] M. Takegaki and S. Arimoto. A new feedback method for dynamic control of manipulators. *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 102, pp. 119–125, 1981.
- [16] H. Goldstein, C. P. Poole, and J. L. Safko. *Classical Mechanics*. Pearson Education, 3rd edition, 2001.