UNSYS digital

Journal of Unmanned System Technology

A Nonlinear Camera Gimbal Visual Servoing using Command Filtered Backstepping

Fadjar Rahino Triputra[†], Bambang Riyanto Trilaksono[‡], Trio Adiono[‡], Rianto Adhy Sasongko[§], Mohamad Dahsyat[†]

[†]Agency for the Assessment and Application of Technology, BPPT, Indonesia

[‡]School of Electrical Engineering and Informatics, Bandung Institute of Technology, Indonesia

[§]Faculty of Mechanical and Aerospace Engineering, Bandung Institute of Technology, Indonesia

Abstract—This paper presents a method for controlling a camera gimbal mechanism that is usually mounted on a fixed-wing unmanned aerial vehicle (UAV). An image-based visual servoing (IBVS) is employed to govern the camera direction always pointing at a target object in the ground or sea using pan-tilt gimbal mechanism while the fixed-wing UAV flying over the object. A command filtered backstepping (CFBS) control law is designed for IBVS and compared to integral backstepping (IBS) control law design to show the design easiness of CFBS and its performance. Software and hardware in the loop simulation (SILS / HILS) results show the effectiveness of the pan-tilt gimbal IBVS using command filtered backstepping to maintain the image feature position always at the center of the image even in radical movements of the fixed-wing UAV that is carrying a camera attached to the pan-tilt gimbal.

Keywords—Image based visual servoing, command filtered backstepping, nonlinear camera gimbal modeling.

I. INTRODUCTION

FIXED-WING platform of unmanned aerial vehicle (UAV) is more prominent for long endurance and long range surveillance mission than a rotary-wing UAV. However, due to its nonholonomic nature, fixed-wing UAV cannot move freely within 6-degree of freedom compared to the rotary-wing UAV. Therefore, a camera must be attached to a pan-tilt gimbal mechanism that governs the camera direction pointing to the desired target object freely in a sustainable manner. Image features of a suspicious unknown target object are obtained in the long-term tracking for governing camera direction to point to the target object. Kalal et al. [1] proposes a tracking framework which decomposes a long-term tracking task into tracking, learning, and detection (TLD). In this paper, this features extraction method is then used to extract a feature point from the selected unknown target object. The feature point is assumed to always be available on the camera image plane during this proposed control system works. If the extracted feature point in image plane is at the center of the image plane, it is indicated that the camera direction points to the target object.

Hutchinson et al. [2] and Chaumette et al. [3-4] have explained about an image-based visual servoing (IBVS) method that governs the camera movement pointing to the target object using a model of feature points motion and its camera movements. Hamel et al. [5] and Bourquardez et al. [6] utilize this IBVS method for rotary-wing UAV with camera fixed on its fuselage to control the UAV position and attitude relative to the target object. Le Bras et al. [7-8] also utilize this IBVS method for fixed-wing UAV landing guidance and circular trajectories with camera fixed on its fuselage too. However, the utilization of camera gimbal mechanism for fixed-wing UAV to track the target object that employs IBVS method is recently developed [9] using integral backstepping (IBS) control law.

Khalil [10] describes nonlinear control law design including the using of the nonlinear model in the form of affine strict feedback system. Krstic et al. [11] then explains an adaptive nonlinear IBS control law design using affine strict feedback system form. The prominent advantages of backstepping method are useful nonlinearities, flexible, recursive and constructive. However, IBS control law design needs analytical time-differential derivation of nonlinear model parts that makes it difficult to develop a control design for complicated camera gimbal movement model that involves radical gimbal carrier movements. Furthermore, Farrel et al. [12-13] introduces a command filtered backstepping (CFBS) control law design that eliminates the requirements of analytical time-differential derivation.

This paper describes a nonlinear camera gimbal visual servoing using CFBS. The technical approach employs the nonlinear dynamics of the image feature motion model and the gimbal pan-tilt mechanism movement in the form of affine strict feedback system. Both IBS and CFBS control law designs are developed and compared using nonlinear camera gimbal model without involving the gimbal carrier movements. Furthermore, this paper shows that more complex nonlinear camera gimbal model involving UAV movements can be designed easily using the CFBS control law. The results of the software and hardware in the loop simulation (SILS/HILS) prove the effectiveness of the proposed control designs using CFBS to converge the image feature residual between the current feature position and the camera image plane center by governing the gimbal mechanism to change the camera attitude pointing to the target object.

Corresponding author: Fadjar R. Triputra (e-mail: fadjar.rahino@bppt.go.id).
This paper was submitted on May 16, 2015; revised on September 20, 2015; and accepted on October 10, 2015.

The contributions of this paper are: (1) a nonlinear modeling of camera gimbal movements related to camera image feature of target object; (2) IBS and CFBS control law design comparison for camera gimbal IBVS; (3) CFBS control law design for camera gimbal IBVS involving the radical gimbal carrier movements of fixed-wing UAV; and (4) both simulation results of SILS and HILS which prove the effectiveness of the proposed control design. These contributions provide great advantages to design an autonomous visual based fixed-wing UAV controller using IBVS involving nonlinear UAV dynamics modeling.

II. Nonlinear Modelling

This section presents nonlinear IBVS modeling that correlates the extracted feature motion from the projected target object in image plane of camera sensor with the movement of pan-tilt camera gimbal mechanism. Regarding to commanded feature position at the center of camera image plane, this model also produces a relative attitude of camera to the gimbal carrier by using pan-tilt angles to make the camera aiming to the target object.

A. Image Feature Motion

In order to make an estimation of camera attitude based on the current feature position in image plane of the camera sensor, a relational model between the feature motion and the camera movement is introduced. **Figure 1** shows the relational model between a target object in 3D world space and its feature in camera image plane. Two 3D coordinate frames (i.e. inertia frame $\langle o_e, x_e, y_e, z_e \rangle$ and camera frame $\langle o_c, x_c, y_c, z_c \rangle$) and a 2D coordinate frame (image frame $\langle o_p, x_p, y_p \rangle$) are utilized to represent coordinates of each point respectively. The target object point $\mathbf{P}_1$ is then defined as ${}^{c}\mathbf{P}_1 = [x_{c_1} \quad y_{c_1} \quad z_{c_1}]^\intercal$ in camera frame and the feature point $\mathbf{p}_1$ is defined as ${}^{i}\mathbf{p}_1 = [p_{x_1} \quad p_{y_1}]^\intercal$ in image frame that can be represented as follows:

$$\begin{bmatrix} {}^{i}\mathbf{p}_1^\intercal & \lambda \end{bmatrix}^\intercal = k\ {}^{c}\mathbf{P}_1 \tag{1}$$

where λ is a focal length of the camera and k is a projection scale.

By substituting $k = \lambda / z_{c_1}$, the feature position in image frame is rewritten as follows

$${}^{i}\mathbf{p}_1 = \frac{\lambda}{z_{c_1}} [x_{c_1} \quad y_{c_1}]^\intercal \tag{2}$$

Then, the target object point $\mathbf{P}_1$ can also be represented as ${}^{e}\mathbf{P}_1 = [x_{e_1} \quad y_{e_1} \quad z_{e_1}]^\intercal$ in inertia frame. Corke [14] writes the correlation of the object target point $\mathbf{P}_1$ between camera and inertia frame in homogoneous form as follows:

$$\begin{bmatrix} {}^{e}\mathbf{P}_1 \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{e}\mathbf{R}_c & {}^{e}\mathbf{T}_c \\ \mathbf{O}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} {}^{c}\mathbf{P}_1 \\ 1 \end{bmatrix} \tag{3}$$

where ${}^{e}\mathbf{R}_c$ and ${}^{e}\mathbf{T}_c$ respectively are rotation matrix and translation vector from camera frame to inertia frame.

Thus, the target object point $\mathbf{P}_1$ in camera frame can be expressed in the homogeneous form as follows:

$$\begin{bmatrix} {}^{c}\mathbf{P}_1 \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{e}\mathbf{R}_c^\intercal & -{}^{e}\mathbf{R}_c^\intercal\ {}^{e}\mathbf{T}_c \\ \mathbf{O}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} {}^{e}\mathbf{P}_1 \\ 1 \end{bmatrix} \tag{4}$$

Furthermore, the Equation (4) of the target point $\mathbf{P}_1$ in camera frame is derived to find the change of target object position in camera frame, which depends on the attitude of the camera in inertia frame, is written as follows:

$${}^{c}\dot{\mathbf{P}}_1 = {}^{e}\dot{\mathbf{R}}_c^\intercal ({}^{e}\mathbf{P}_1 - {}^{e}\mathbf{T}_c) + {}^{e}\mathbf{R}_c^\intercal ({}^{e}\dot{\mathbf{P}}_1 - {}^{e}\dot{\mathbf{T}}_c) \tag{5}$$

where ${}^{e}\dot{\mathbf{R}}_c^\intercal$ is rotation matrix rate from camera frame to inertia that is defined as follows:

$${}^{e}\dot{\mathbf{R}}_c^\intercal = \mathbf{S}({}^{e}\boldsymbol{\Omega})\ {}^{e}\mathbf{R}_c^\intercal \tag{6}$$

${}^{e}\boldsymbol{\Omega} = [{}^{e}p \quad {}^{e}q \quad {}^{e}r]^\intercal$ is angular velocity of the camera in inertia frame and $\mathbf{S}(\cdot)$ is a skew matrix that is defined as follows:

$$\mathbf{S}({}^{e}\boldsymbol{\Omega}) = \begin{bmatrix} 0 & -{}^{e}r & {}^{e}q \\ {}^{e}r & 0 & -{}^{e}p \\ -{}^{e}q & {}^{e}p & 0 \end{bmatrix} \tag{7}$$

Afterward, the equation (5) is rewritten as follows:

$$\begin{aligned} {}^{c}\dot{\mathbf{P}}_1 &= -{}^{e}\mathbf{R}_c^\intercal\ {}^{e}\boldsymbol{\Omega} \times {}^{e}\mathbf{R}_c^\intercal ({}^{e}\mathbf{P}_1 - {}^{e}\mathbf{T}_c) - {}^{e}\mathbf{R}_c^\intercal\ {}^{e}\dot{\mathbf{T}}_c \\ &= -{}^{c}\boldsymbol{\Omega} \times {}^{c}\mathbf{P}_1 - {}^{c}\mathbf{V} \end{aligned} \tag{8}$$

where ${}^{c}\mathbf{V} = [{}^{c}U \quad {}^{c}V \quad {}^{c}W]^\intercal$ and ${}^{c}\boldsymbol{\Omega} = [{}^{c}p \quad {}^{c}q \quad {}^{c}r]^\intercal$ respectively are velocity and angular velocity of camera in camera frame.

By substituting feature position in Equation (2), the camera target position movement in Equation (8) is rewritten as follows:

$${}^{c}\dot{\mathbf{P}}_1 = \begin{bmatrix} \frac{p_{y_1} z_{c_1}}{\lambda}\ {}^{c}r - z_{c_1}\ {}^{c}q - {}^{c}U \\ z_{c_1}\ {}^{c}p - \frac{p_{x_1} z_{c_1}}{\lambda}\ {}^{c}r - {}^{c}V \\ \frac{p_{x_1} z_{c_1}}{\lambda}\ {}^{c}q - \frac{p_{y_1} z_{c_1}}{\lambda}\ {}^{c}p - {}^{c}W \end{bmatrix} \tag{9}$$

Image feature motion in camera sensor image plane then is obtained using the derivation of Equation (2) as follows:

$${}^{i}\dot{\mathbf{p}}_1 = \frac{\lambda}{z_{c_1}^2} \begin{bmatrix} \dot{x}_{c_1} z_{c_1} - x_{c_1} \dot{z}_{c_1} \\ \dot{y}_{c_1} z_{c_1} - y_{c_1} \dot{z}_{c_1} \end{bmatrix} \tag{10}$$

Hence, the image feature motion model can be represented by substituting Equation (9) into Equation (10) as follows:

$$\dot{\mathbf{p}} = \mathbf{L}(\mathbf{p}, z)[{}^{c}\mathbf{V} \quad {}^{c}\boldsymbol{\Omega}]^\intercal \tag{11}$$

where $\mathbf{p}$ is feature position in image plane of camera sensor, z is distance between camera and target object, and $\mathbf{L}(\mathbf{p}, z)$ is a Jacobian matrix that is defined as follows:

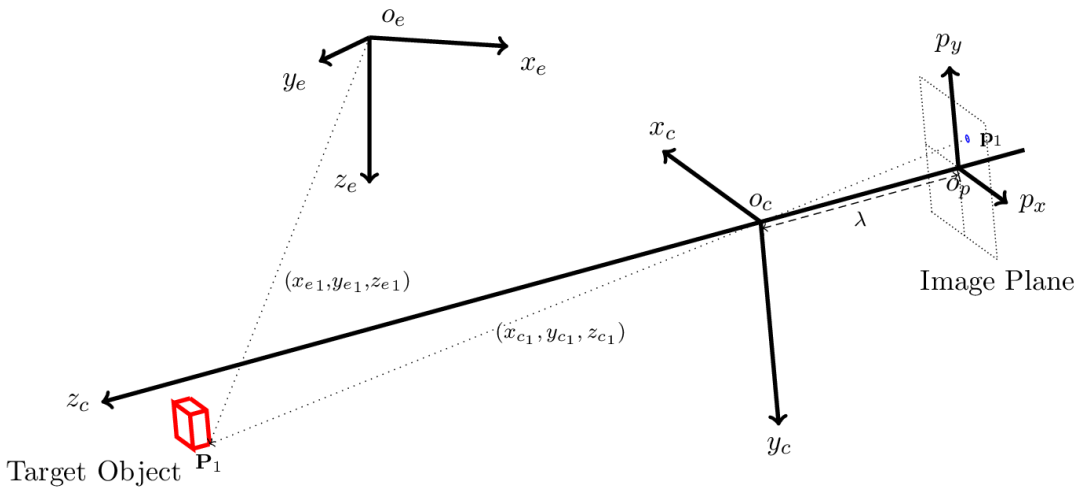


Figure 1 Target object point in 3D world space and its feature in image plane of camera sensor

$$\mathbf{L}(\mathbf{p}, z) = \begin{bmatrix} -\frac{\lambda}{z} & 0 & \frac{p_x}{z} & \frac{p_x p_y}{\lambda} & -\frac{\lambda^2 + p_x^2}{\lambda} & p_y \\ 0 & -\frac{\lambda}{z} & \frac{p_y}{z} & \frac{\lambda^2 + p_y^2}{\lambda} & -\frac{p_x p_y}{\lambda} & -p_x \end{bmatrix} \quad (12)$$

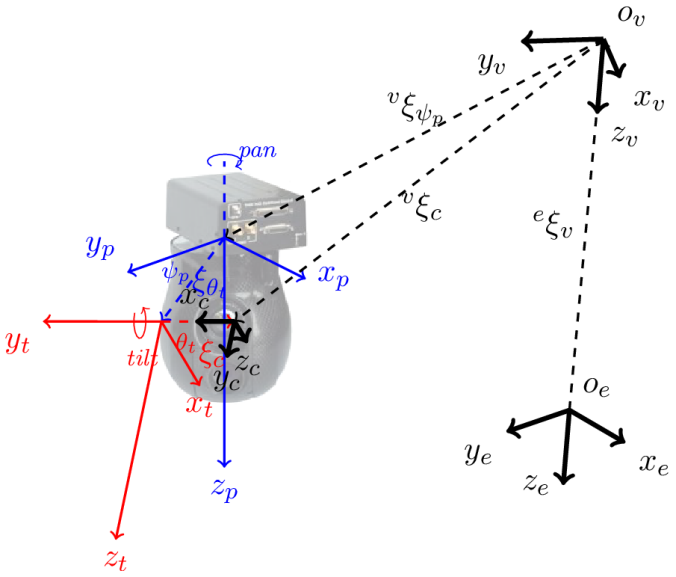


Figure 2 Correlation frames of pan-tilt gimbal, camera, vehicle, and inertia

B. Gimbal pan-tilt movement

In order to make an estimation of camera attitude based on the current pan-tilt gimbal attitude that is relative to a carrier such as UAV, a model of pan-tilt gimbal mechanism movement is proposed. **Figure 2** shows a correlation between a carrier vehicle of gimbal, a pan-tilt gimbal mechanism, and a camera. Besides the camera and inertia frames, three 3D coordinate frames i.e. pan frame $\langle o_p, x_p, y_p, z_p \rangle$, tilt frame $\langle o_t, x_t, y_t, z_t \rangle$, and vehicle frame $\langle o_v, x_v, y_v, z_v \rangle$ as the gimbal carrier are added. The vehicle frame is put exactly at z-axis above the inertia frame.

The frame transformation from vehicle to inertia frame is defined as follows:

$$^{e}\boldsymbol{\xi}_{v} = \begin{bmatrix} ^{e}\mathbf{R}_{v} & ^{e}\mathbf{T}_{v} \\ \mathbf{O}_{1\times 3} & 1 \end{bmatrix} \quad (13)$$

where $^{e}\mathbf{R}_{v} = {}^{v}\mathbf{R}_{e}^{\intercal}$ and $^{e}\mathbf{T}_{v} = [0 \quad 0 \quad -h]^{\intercal}$ respectively are rotation matrix and translation vector from vehicle frame to inertia frame, h is the altitude of gimbal carrier. Rotation matrix from inertia frame to vehicle frame is defined as follows:

$$^{v}\mathbf{R}_{e} = {}^{v}\mathbf{R}_{\phi}\,{}^{\phi}\mathbf{R}_{\theta}\,{}^{\theta}\mathbf{R}_{e} \quad (14)$$

$$^{v}\mathbf{R}_{\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \quad (15)$$

$$^{\phi}\mathbf{R}_{\theta} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \quad (16)$$

$$^{\theta}\mathbf{R}_{e} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (17)$$

where ϕ, θ, and ψ respectively are roll, pitch and yaw angles of the gimbal carrier attitude.

The target object point $\mathbf{P}_1$ is then defined as $^{e}\mathbf{P}_1$ in inertia frame and also defined as $^{v}\mathbf{P}_1 = [x_{v_1} \quad y_{v_1} \quad z_{v_1}]^{\intercal}$ in vehicle frame. Thus, the correlation between inertia and vehicle frames in homogenous form is written as follows:

$$\begin{bmatrix} ^{e}\mathbf{P}_1 \\ 1 \end{bmatrix} = {}^{e}\boldsymbol{\xi}_{v} \begin{bmatrix} ^{v}\mathbf{P}_1 \\ 1 \end{bmatrix} \quad (18)$$

The target object in vehicle frame is then represented as follows:

$$\begin{bmatrix} ^{v}\mathbf{P}_1 \\ 1 \end{bmatrix} = (\ominus\ {}^{e}\boldsymbol{\xi}_{v}) \begin{bmatrix} ^{e}\mathbf{P}_1 \\ 1 \end{bmatrix} \quad (19)$$

$$= \begin{bmatrix} {}^{e}\mathbf{R}_{v}^{\intercal} & -{}^{e}\mathbf{R}_{v}^{\intercal}\,{}^{e}\mathbf{T}_{v} \\ \mathbf{O}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} {}^{e}\mathbf{P}_{1} \\ 1 \end{bmatrix}$$

where $\ominus$ is inverse operation of frame transformation matrix.

After deriving the Equation (19), the target object movement in vehicle frame can be obtained as follows:

$${}^{v}\dot{\mathbf{P}}_{1} = {}^{e}\dot{\mathbf{R}}_{v}^{\intercal}\left({}^{e}\mathbf{P}_{1} - {}^{e}\mathbf{T}_{v}\right) + {}^{e}\mathbf{R}_{v}^{\intercal}\left({}^{e}\dot{\mathbf{P}}_{1} - {}^{e}\dot{\mathbf{T}}_{v}\right) \tag{20}$$

If the target object is assumed to be still in inertia frame, the target object movement in inertia frame is rewritten as follows:

$${}^{v}\dot{\mathbf{P}}_{1} = -\boldsymbol{\Omega} \times {}^{v}\mathbf{P}_{1} - \mathbf{V} \tag{21}$$

where $\mathbf{V} = [U \quad V \quad W]^{\intercal}$ and $\boldsymbol{\Omega} = [p \quad q \quad r]^{\intercal}$ respectively are velocity and angular velocity of the vehicle in vehicle frame.

The frame transformation from pan frame to vehicle frame is also defined as follows:

$${}^{v}\boldsymbol{\xi}_{\psi_p} = \begin{bmatrix} {}^{v}\mathbf{R}_{\psi_p} & {}^{v}\mathbf{T}_{\psi_p} \\ \mathbf{O}_{1\times 3} & 1 \end{bmatrix} \tag{22}$$

where ψ_p is pan angle of gimbal mechanism, ${}^{v}\mathbf{R}_{\psi_p}$ and ${}^{v}\mathbf{T}_{\psi_p} = \left[{}^{v}x_{\psi_p} \quad {}^{v}y_{\psi_p} \quad {}^{v}z_{\psi_p}\right]^{\intercal}$ respectively are rotation matrix and translation vector from pan frame to vehicle frame,

$${}^{v}\mathbf{R}_{\psi_p} = \begin{bmatrix} \cos\psi_p & -\sin\psi_p & 0 \\ \sin\psi_p & \cos\psi_p & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{23}$$

Target object point $\mathbf{P}_1$ is then also defined as ${}^{\psi_p}\mathbf{P}_1 = [{}^{x}\psi_{p_1} \quad {}^{y}\psi_{p_1} \quad {}^{z}\psi_{p_1}]^{\intercal}$ in pan frame. Thus, the target object in pan frame is written as follows:

$$\begin{bmatrix} {}^{\psi_p}\mathbf{P}_1 \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{v}\mathbf{R}_{\psi_p}^{\intercal} & -{}^{v}\mathbf{R}_{\psi_p}^{\intercal}\,{}^{v}\mathbf{T}_{\psi_p} \\ \mathbf{O}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} {}^{v}\mathbf{P}_1 \\ 1 \end{bmatrix} \tag{24}$$

The Equation (24) is derived to obtain target object movement in pan frame as follows:

$${}^{\psi_p}\dot{\mathbf{P}}_1 = {}^{v}\dot{\mathbf{R}}_{\psi_p}^{\intercal}\left({}^{v}\mathbf{P}_1 - {}^{v}\mathbf{T}_{\psi_p}\right) + {}^{v}\mathbf{R}_{\psi_p}^{\intercal}\left({}^{v}\dot{\mathbf{P}}_1 - {}^{v}\dot{\mathbf{T}}_{\psi_p}\right) \tag{25}$$

Usually, ${}^{v}\dot{\mathbf{T}}_{\psi_p} = \mathbf{0}$ because position of pan frame origin is fixed to the vehicle frame origin. The target object movement of Equation (21) in pan frame is rewritten as follows:

$$\begin{aligned} {}^{\psi_p}\dot{\mathbf{P}}_1 &= -\left({}^{v}\mathbf{R}_{\psi_p}^{\intercal}\boldsymbol{\Omega} + {}^{\psi_p}\boldsymbol{\Omega}_{v}\right) \times {}^{\psi_p}\mathbf{P}_1 \\ &\quad - {}^{v}\mathbf{R}_{\psi_p}^{\intercal}\left(\mathbf{V} + \boldsymbol{\Omega} \times {}^{v}\mathbf{T}_{\psi_p}\right) \\ &= -{}^{\psi_p}\boldsymbol{\Omega} \times {}^{\psi_p}\mathbf{P}_1 - {}^{\psi_p}\mathbf{V} \end{aligned} \tag{26}$$

where ${}^{\psi_p}\boldsymbol{\Omega}_{v} = \begin{bmatrix} 0 & 0 & \dot{\psi}_p \end{bmatrix}^{\intercal}$ is local pan angular velocity.

From velocity and angular velocity terms of Equation (26), the target object velocity and angular velocity in pan frame respectively are obtained as follows:

$${}^{\psi_p}\mathbf{V} = {}^{\psi_p}\mathbf{R}_{v}\left(\mathbf{V} + \boldsymbol{\Omega} \times {}^{v}\mathbf{T}_{\psi_p}\right) \tag{27}$$

$${}^{\psi_p}\boldsymbol{\Omega} = {}^{\psi_p}\mathbf{R}_{v}\boldsymbol{\Omega} + {}^{\psi_p}\boldsymbol{\Omega}_{v} \tag{28}$$

where ${}^{\psi_p}\mathbf{R}_{v} = {}^{v}\mathbf{R}_{\psi_p}^{\intercal}$ is rotation matrix from vehicle frame to pan frame.

In the same way with Equations (22) - (26), the target object movement in tilt frame is written as follows:

$$\begin{aligned} {}^{\theta_t}\dot{\mathbf{P}}_1 &= -\left({}^{\psi_p}\mathbf{R}_{\theta_t}^{\intercal}\,{}^{\psi_p}\boldsymbol{\Omega} + {}^{\theta_t}\boldsymbol{\Omega}_{\psi_p}\right) \times {}^{\theta_t}\mathbf{P}_1 \\ &\quad - {}^{\psi_p}\mathbf{R}_{\theta_t}^{\intercal}\left({}^{\psi_p}\mathbf{V} + {}^{\psi_p}\boldsymbol{\Omega} \times {}^{\psi_p}\mathbf{T}_{\theta_t}\right) \\ &= -{}^{\theta_t}\boldsymbol{\Omega} \times {}^{\theta_t}\mathbf{P}_1 - {}^{\theta_t}\mathbf{V} \end{aligned} \tag{29}$$

where θ_t is a tilt angle of gimbal, ${}^{\theta_t}\boldsymbol{\Omega}_{\psi_p} = [0 \quad \dot{\theta}_t \quad 0]^{\intercal}$ is local tilt angular velocity, ${}^{\psi_p}\mathbf{R}_{\theta_t}$ and ${}^{\psi_p}\mathbf{T}_{\theta_t} = \left[{}^{\psi_p}x_{\theta_t} \quad {}^{\psi_p}y_{\theta_t} \quad {}^{\psi_p}z_{\theta_t}\right]^{\intercal}$ are rotation matrix and translation vector respectively from tilt frame to pan frame,

$${}^{\psi_p}\mathbf{R}_{\theta_t} = \begin{bmatrix} \cos\theta_t & 0 & \sin\theta_t \\ 0 & 1 & 0 \\ -\sin\theta_t & 0 & \cos\theta_t \end{bmatrix} \tag{30}$$

The target object velocity and angular velocity in tilt frame respectively are written as follows:

$$\begin{aligned} {}^{\theta_t}\mathbf{V} &= {}^{\theta_t}\mathbf{R}_{\psi_p}\left({}^{\psi_p}\mathbf{V} + {}^{\psi_p}\boldsymbol{\Omega} \times {}^{\psi_p}\mathbf{T}_{\theta_t}\right) \\ &= {}^{\theta_t}\mathbf{R}_{\psi_p}\left\{{}^{\psi_p}\mathbf{R}_{v}\left(\mathbf{V} + \boldsymbol{\Omega} \times {}^{v}\mathbf{T}_{\psi_p}\right)\right. \\ &\quad \left. + \left({}^{\psi_p}\mathbf{R}_{v}\boldsymbol{\Omega} + {}^{\psi_p}\boldsymbol{\Omega}_{v}\right) \times {}^{\psi_p}\mathbf{T}_{\theta_t}\right\} \end{aligned} \tag{31}$$

$$\begin{aligned} {}^{\theta_t}\boldsymbol{\Omega} &= {}^{\theta_t}\mathbf{R}_{\psi_p}\,{}^{\psi_p}\boldsymbol{\Omega} + {}^{\theta_t}\boldsymbol{\Omega}_{\psi_p} \\ &= {}^{\theta_t}\mathbf{R}_{\psi_p}\left({}^{\psi_p}\mathbf{R}_{v}\boldsymbol{\Omega} + {}^{\psi_p}\boldsymbol{\Omega}_{v}\right) + {}^{\theta_t}\boldsymbol{\Omega}_{\psi_p} \end{aligned} \tag{32}$$

where ${}^{\theta_t}\mathbf{R}_{\psi_p} = {}^{\psi_p}\mathbf{R}_{\theta_t}^{\intercal}$ is rotation matrix from pan frame to tilt frame.

Finally, the camera x_c-axis, y_c-axis and z_c-axis respectively are put parallel and same direction with the tilt x_t-axis, x_t-axis, and x_t-axis. Thus, the frame transformation from camera frame to tilt frame can be written as follows:

$${}^{\theta_t}\boldsymbol{\xi}_{c} = \begin{bmatrix} {}^{\theta_t}\mathbf{R}_{c} & {}^{\theta_t}\mathbf{T}_{c} \\ \mathbf{O}_{1\times 3} & 1 \end{bmatrix} \tag{33}$$

where ${}^{\theta_t}\mathbf{R}_{c}$ and ${}^{\theta_t}\mathbf{T}_{c} = \left[{}^{\theta_t}x_{c} \quad {}^{\theta_t}y_{c} \quad {}^{\theta_t}z_{c}\right]^{\intercal}$ are rotation matrix and translation vector respectively from camera frame to tilt frame,

$${}^{\theta_t}\mathbf{R}_{c} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{34}$$

The target object velocity and angular velocity in camera frame respectively are obtained as follows:

$$
\begin{aligned}
{}^{c}\mathbf{V} &= {}^{c}\mathbf{R}_{\theta_t}\left({}^{\theta_t}\mathbf{V} + {}^{\theta_t}\boldsymbol{\Omega} \times {}^{\theta_t}\mathbf{T}_c\right) \\
&= {}^{c}\mathbf{R}_{\theta_t}\Big\{{}^{\theta_t}\mathbf{R}_{\psi_p}\left({}^{\psi_p}\mathbf{R}_v\left(\mathbf{V} + \boldsymbol{\Omega} \times {}^{v}\mathbf{T}_{\psi_p}\right)\right. \\
&\qquad \left. + \left({}^{\psi_p}\mathbf{R}_v\boldsymbol{\Omega} + {}^{\psi_p}\boldsymbol{\Omega}_v\right) \times {}^{\psi_p}\mathbf{T}_{\theta_t}\right) \\
&\qquad + \left({}^{\theta_t}\mathbf{R}_{\psi_p}\left({}^{\psi_p}\mathbf{R}_v\boldsymbol{\Omega} + {}^{\psi_p}\boldsymbol{\Omega}_v\right) + {}^{\theta_t}\boldsymbol{\Omega}_{\psi_p}\right) \times {}^{\psi_p}\mathbf{T}_{\theta_{t_p}}\Big\} \\
&= {}^{c}\mathbf{R}_v\mathbf{V} + {}^{c}\mathbf{R}_v^{\dagger}\boldsymbol{\Omega} + {}^{c}\mathbf{R}_v^{\ddagger}\dot{\boldsymbol{\Theta}}
\end{aligned} \tag{35}
$$

$$
\begin{aligned}
{}^{c}\boldsymbol{\Omega} &= {}^{c}\mathbf{R}_{\theta_t}{}^{\theta_t}\boldsymbol{\Omega} \\
&= {}^{c}\mathbf{R}_{\theta_t}\left\{{}^{\theta_t}\mathbf{R}_{\psi_p}\left({}^{\psi_p}\mathbf{R}_v\boldsymbol{\Omega} + {}^{\psi_p}\boldsymbol{\Omega}_v\right) + {}^{\theta_t}\boldsymbol{\Omega}_{\psi_p}\right\} \\
&= {}^{c}\mathbf{R}_v\boldsymbol{\Omega} + {}^{c}\mathbf{R}_v^{\wedge}\dot{\boldsymbol{\Theta}}
\end{aligned} \tag{36}
$$

where ${}^{c}\mathbf{R}_{\theta_t} = {}^{\theta_t}\mathbf{R}_c^{\intercal}$ is rotation matrix from tilt frame to camera frame, ${}^{c}\mathbf{R}_v = {}^{c}\mathbf{R}_{\theta_t}{}^{\theta_t}\mathbf{R}_{\psi_p}{}^{\psi_p}\mathbf{R}_v$ is rotation matrix from vehicle frame to camera frame, $\boldsymbol{\Theta} = [\theta_t \quad \psi_p]^{\intercal}$ is pan-tilt angle vector,

$$
{}^{c}\mathbf{R}_v^{\dagger} = -\,{}^{c}\mathbf{R}_v\mathbf{S}({}^{v}\mathbf{T}_c) \tag{37}
$$

$$
{}^{c}\mathbf{R}_v^{\ddagger} = \begin{bmatrix} 0 & {}^{\psi_p}x_c \\ -{}^{\theta_t}x_c & -{}^{\psi_p}y_c \sin\theta_t \\ {}^{\theta_t}z_c & -{}^{\psi_p}y_c \cos\theta_t \end{bmatrix} \tag{38}
$$

$$
{}^{c}\mathbf{R}_v^{\wedge} = \begin{bmatrix} 1 & 0 \\ 0 & \cos\theta_t \\ 0 & -\sin\theta_t \end{bmatrix} \tag{39}
$$

where ${}^{v}\mathbf{T}_c = {}^{v}\mathbf{T}_{\psi_p} + {}^{\psi_p}\mathbf{T}_{\theta_t} + {}^{\theta_t}\mathbf{T}_c$, ${}^{\psi_p}x_c = {}^{\psi_p}x_{\theta_t} + {}^{\theta_t}x_c$, and ${}^{\psi_p}y_c = {}^{\psi_p}y_{\theta_t} + {}^{\theta_t}y_c$.

The camera velocity of Equation (35) and angular velocity of Equation (36) are then substituted into Equation (11) to get a model set of IBVS in affine strict feedback system form that can be written as follows:

$$
\dot{\mathbf{p}} = \mathbf{L}(\mathbf{p}, z)\left({}^{c}\mathbf{R}_v^{\boldsymbol{\Omega}}\begin{bmatrix}\mathbf{V} \\ \boldsymbol{\Omega}\end{bmatrix} + {}^{c}\mathbf{R}_v^{\boldsymbol{\Theta}}\mathbf{u}_{\boldsymbol{\Theta}}\right) \tag{40}
$$

$$
\dot{\boldsymbol{\Theta}} = \mathbf{u}_{\boldsymbol{\Theta}} \tag{41}
$$

where $\mathbf{u}_{\boldsymbol{\Theta}} = [u_{\theta_t} \quad u_{\psi_p}]^{\intercal}$ is pan-tilt control input,

$$
{}^{c}\mathbf{R}_v^{\boldsymbol{\Omega}} = \begin{bmatrix} {}^{c}\mathbf{R}_v & {}^{c}\mathbf{R}_v^{\dagger} \\ \mathbf{0}_{3\times 3} & {}^{c}\mathbf{R}_v \end{bmatrix} \tag{42}
$$

$$
{}^{c}\mathbf{R}_v^{\boldsymbol{\Theta}} = \begin{bmatrix} {}^{c}\mathbf{R}_v^{\ddagger} \\ {}^{c}\mathbf{R}_v^{\wedge} \end{bmatrix} \tag{43}
$$

The first term of Equation (40) is gimbal carrier movements and the second term is gimbal mechanism controls. The gimbal carrier velocity and angular velocity is complex model especially for nonlinear UAV dynamics so it is difficult to derive as time-differential analytically.

III. Adaptive Backstepping Control

In this section, IBVS is employed to design nonlinear adaptive backstepping control system that has prominent advantages [11] such as (1) globally asymptotically stable; (2) model uncertainties are well handled; (3) systematic procedures; (4) potential reduction in development time; and (5) useful nonlinearities are retained. However, there are some disadvantages for conventional backstepping methods such as (1) analytical time-differential derivation; and (2) feedback control algorithm complexity, especially for high order systems.

A. Integral Backstepping

To initiate discussions on nonlinear adaptive control using IBS, the model that is used to design the control system must be established. To reduce the complexity of analytic derivative calculation for IBS, the gimbal carrier is assumed to not move and then the model of Equation (40) is simplified without gimbal carrier dynamics term as follows:

$$
\dot{\mathbf{p}} = \mathbf{L}_{\otimes}\mathbf{U}_{\boldsymbol{\Theta}} \tag{44}
$$

$$
\dot{\mathbf{U}}_{\boldsymbol{\Theta}} = \boldsymbol{\Gamma}_{\boldsymbol{\Theta}} \tag{45}
$$

where $\mathbf{U}_{\boldsymbol{\Theta}} = \begin{bmatrix}U_{\theta_t} & U_{\psi_p}\end{bmatrix}^{\intercal}$ and $\boldsymbol{\Gamma}_{\boldsymbol{\Theta}} = \begin{bmatrix}\zeta_{\theta_t} & \zeta_{\psi_p}\end{bmatrix}^{\intercal}$ respectively are virtual control and backstepping output, and $\mathbf{L}_{\otimes}$ is input coefficient matrix that is defined as follows:

$$
\mathbf{L}_{\otimes} = \mathbf{L}(\mathbf{p}, z)\ {}^{c}\mathbf{R}_v^{\boldsymbol{\Theta}} \tag{46}
$$

Assuming the distance of $z \gg p_x, p_y, \lambda, {}^{\theta_t}x_c, {}^{\theta_t}z_c, {}^{\psi_p}x_c, {}^{\psi_p}y_c$, the input coeficient matrix is then written as follows:

$$
\mathbf{L}_{\otimes} = \begin{bmatrix} L_{\otimes_{1.1}} & L_{\otimes_{1.2}} \\ L_{\otimes_{2.1}} & L_{\otimes_{2.2}} \end{bmatrix} \tag{47}
$$

$$
L_{\otimes_{1.1}} = \frac{p_x p_y}{\lambda} \tag{48}
$$

$$
L_{\otimes_{1.2}} = -\frac{\lambda^2 + p_x^2}{\lambda}\cos\theta_t - p_y \sin\theta_t \tag{49}
$$

$$
L_{\otimes_{2.1}} = \frac{\lambda^2 + p_y^2}{\lambda} \tag{50}
$$

$$
L_{\otimes_{2.2}} = p_x \sin\theta_t - \frac{p_x p_y}{\lambda}\cos\theta_t \tag{51}
$$

A stabilization function for reducing image feature residual is then chosen to estimate the desired virtual control using simple model of Equation (44) as follows:

$$
\mathbf{U}_{\boldsymbol{\Theta}_{des}} = -\gamma_{\mathbf{p}}\mathbf{L}_{\otimes}^{-1}(\mathbf{p} - \mathbf{p}_c) = -\gamma_{\mathbf{p}}\mathbf{L}_{\otimes}^{-1}\tilde{\mathbf{p}} \tag{52}
$$

where $\gamma_{\mathbf{p}}$ is positive scalar gain for reducing feature residual, $\mathbf{p}_c$ is commanded feature at the center of camera sensor image plane, and $\tilde{\mathbf{p}}$ is image feature residual.

Next, a virtual control error is then defined as follows:

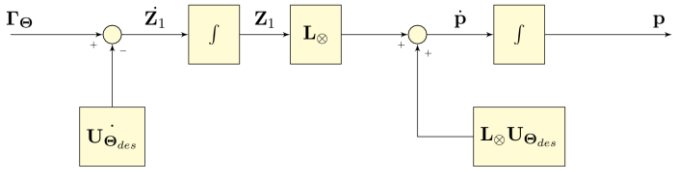
$$
\mathbf{Z}_1 = \mathbf{U}_{\boldsymbol{\Theta}} - \mathbf{U}_{\boldsymbol{\Theta}_{des}} \tag{53}
$$

The virtual control $\mathbf{U}_{\boldsymbol{\Theta}}$ in Equation (53) is then substituted into Equation (44) to obtain IBS model design as follows:

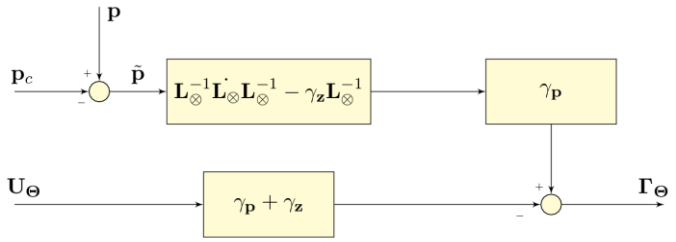
$$
\dot{\mathbf{p}} = \mathbf{L}_{\otimes}\mathbf{U}_{\boldsymbol{\Theta}_{des}} + \mathbf{L}_{\otimes}\mathbf{Z}_1 \tag{54}
$$

$$\dot{\mathbf{Z}}_1 = \boldsymbol{\Gamma}_{\boldsymbol{\Theta}} - \dot{\mathbf{U}}_{\boldsymbol{\Theta}_{des}} \tag{55}$$

Figure 3(a) shows the IBS model design procedures that need the analytical time-differential derivation of the desired virtual control function of Equation (52) in the procedure of Equation (55) that make more complex calculation if the gimbal carrier dynamics is involved.



(a)



(b)

Figure 3 Integral backstepping design for camera gimbal IBVS. (a) Design procedures, and (b) control law design

After deriving Equation (52), Equation (55) is rewritten as follows:

$$\dot{\mathbf{Z}}_1 = \boldsymbol{\Gamma}_{\boldsymbol{\Theta}} + \gamma_{\mathbf{p}}\left(\dot{\mathbf{L}}_{\otimes}^{-1}\tilde{\mathbf{p}} + \mathbf{U}_{\boldsymbol{\Theta}}\right) \tag{56}$$

Afterwards, a stabilization function for reducing the virtual control error $\mathbf{Z}_1$ is then defined as follows:

$$\dot{\mathbf{Z}}_1 = -\gamma_{\mathbf{Z}}\mathbf{Z}_1 \tag{57}$$

where $\gamma_{\mathbf{Z}}$ is positive scalar gain to reduce virtual control error.

Thus, the backstepping output is obtained as follows:

$$\begin{aligned}\boldsymbol{\Gamma}_{\boldsymbol{\Theta}} &= -\gamma_{\mathbf{p}}\left(\dot{\mathbf{L}}_{\otimes}^{-1}\tilde{\mathbf{p}} + \mathbf{U}_{\boldsymbol{\Theta}}\right) - \gamma_{\mathbf{Z}}\mathbf{Z}_1 \\ &= -\gamma_{\mathbf{p}}\left(\mathbf{L}_{\otimes}^{-1}\dot{\mathbf{L}}_{\otimes}\mathbf{L}_{\otimes}^{-1} - \gamma_{\mathbf{Z}}\mathbf{L}_{\otimes}^{-1}\right)\tilde{\mathbf{p}} - \left(\gamma_{\mathbf{p}} + \gamma_{\mathbf{Z}}\right)\mathbf{U}_{\boldsymbol{\Theta}}\end{aligned} \tag{58}$$

Figure 3(b) shows the IBS control law design of backstepping output $\boldsymbol{\Gamma}_{\boldsymbol{\Theta}}$. Finally, the time-differential derivation of Jacobian matrix is calculated as follows:

$$\dot{\mathbf{L}}_{\otimes} = \begin{bmatrix} \dot{L}_{\otimes_{1.1}} & \dot{L}_{\otimes_{1.2}} \\ \dot{L}_{\otimes_{2.1}} & \dot{L}_{\otimes_{2.2}} \end{bmatrix} \tag{59}$$

$$\dot{L}_{\otimes_{1.1}} = \begin{bmatrix} \frac{p_y}{\lambda} & \frac{p_x}{\lambda} \end{bmatrix}\mathbf{L}_{\otimes}\mathbf{U}_{\boldsymbol{\Theta}} \tag{60}$$

$$\dot{L}_{\otimes_{1.2}} = \left(\begin{bmatrix} \frac{2p_x}{\lambda}\cos\theta_t & -\sin\theta_t \end{bmatrix}\mathbf{L}_{\otimes} - \begin{bmatrix} \frac{\lambda^2 + p_x^2}{\lambda}\sin\theta_t + p_y\cos\theta_t & 0 \end{bmatrix}\right)\mathbf{U}_{\boldsymbol{\Theta}} \tag{61}$$

$$\dot{L}_{\otimes_{2.1}} = \begin{bmatrix} 0 & \frac{2p_y}{\lambda} \end{bmatrix}\mathbf{L}_{\otimes}\mathbf{U}_{\boldsymbol{\Theta}} \tag{62}$$

$$\dot{L}_{\otimes_{2.2}} = \left(\begin{bmatrix} -\frac{p_y}{\lambda}\cos\theta_t + \sin\theta_t & -\frac{p_x}{\lambda}\cos\theta_t \end{bmatrix}\mathbf{L}_{\otimes} + \begin{bmatrix} \frac{p_x p_y}{\lambda}\sin\theta_t + p_x\cos\theta_t & 0 \end{bmatrix}\right)\mathbf{U}_{\boldsymbol{\Theta}} \tag{63}$$

B. Command Filtered Backstepping

Similar to the IBS, the simple model of Equations (44) and (45) is employed to design CFBS control law for comparing the design procedure and the performance of IBVS. The residuals of feature and virtual control are defined as follows:

$$\tilde{\mathbf{p}} = \mathbf{p} - \mathbf{p}_c \tag{64}$$

$$\tilde{\mathbf{U}}_{\boldsymbol{\Theta}} = \mathbf{U}_{\boldsymbol{\Theta}} - \mathbf{U}_{\boldsymbol{\Theta}_c} \tag{65}$$

where $\mathbf{U}_{\boldsymbol{\Theta}_c}$ is commanded virtual control.

Based on the number of residuals, two stages of CFBS control law design are then built for IBVS. The stabilization function for the first stage is chosen to estimate the desired virtual control using feature residual as follows:

$$\mathbf{U}_{\boldsymbol{\Theta}_{des}} = \mathbf{L}_{\otimes}^{-1}\left(-\gamma_{\mathbf{p}}\tilde{\mathbf{p}} + \dot{\mathbf{p}}_c\right) \tag{66}$$

where $\dot{\mathbf{p}}_c = \mathbf{0}$ is commanded feature motion rate to suppress the movement.

Furthermore, an unfiltered commanded virtual control is defined along with a low pass filter (LPF) transfer function as follows:

$$\hat{\mathbf{U}}_{\boldsymbol{\Theta}_c} = \mathbf{U}_{\boldsymbol{\Theta}_{des}} - \boldsymbol{\Upsilon}_{\mathbf{U}_{\boldsymbol{\Theta}}} \tag{67}$$

$$\dot{\boldsymbol{\Upsilon}}_{\mathbf{p}} = -\gamma_{\mathbf{p}}\boldsymbol{\Upsilon}_{\mathbf{p}} + \mathbf{L}_{\otimes}\left(\mathbf{U}_{\boldsymbol{\Theta}_c} - \hat{\mathbf{U}}_{\boldsymbol{\Theta}_c}\right) \tag{68}$$

where $\boldsymbol{\Upsilon}_{\mathbf{U}_{\boldsymbol{\Theta}}}$ is compensated virtual control residual that will be defined at the second stage, $\mathbf{U}_{\boldsymbol{\Theta}_c}$ is a filtered commanded virtual control, $\boldsymbol{\Upsilon}_{\mathbf{p}}$ is compensated feature residual.

Compensated feature tracking error is also defined as follows:

$$\bar{\mathbf{p}} = \tilde{\mathbf{p}} - \boldsymbol{\Upsilon}_{\mathbf{p}} \tag{69}$$

The stabilization function for the second stage is then chosen to estimate the desired backstepping output as follows:

$$\boldsymbol{\Gamma}_{\boldsymbol{\Theta}_{des}} = -\gamma_{\mathbf{U}_{\boldsymbol{\Theta}}}\tilde{\mathbf{U}}_{\boldsymbol{\Theta}} + \dot{\mathbf{U}}_{\boldsymbol{\Theta}_c} \tag{70}$$

where $\gamma_{\mathbf{U}_{\boldsymbol{\Theta}}}$ is positive scalar gain for reducing virtual control residual, and $\dot{\mathbf{U}}_{\boldsymbol{\Theta}_c}$ is a filtered commanded virtual control rate.

Similar to the first stage procedure, an unfiltered commanded backstepping output is defined along with LPF transfer function as follows:

$$\hat{\boldsymbol{\Gamma}}_{\boldsymbol{\Theta}_c} = \boldsymbol{\Gamma}_{\boldsymbol{\Theta}_{des}} - \bar{\mathbf{p}}^{\mathrm{T}}\mathbf{L}_{\otimes} \tag{71}$$

$$\dot{\boldsymbol{\Upsilon}}_{\mathbf{U}_{\boldsymbol{\Theta}}} = -\gamma_{\mathbf{U}_{\boldsymbol{\Theta}}}\boldsymbol{\Upsilon}_{\mathbf{U}_{\boldsymbol{\Theta}}} + \boldsymbol{\Gamma}_{\boldsymbol{\Theta}_c} - \hat{\boldsymbol{\Gamma}}_{\boldsymbol{\Theta}_c} \tag{72}$$

where $\boldsymbol{\Gamma}_{\boldsymbol{\Theta}_c}$ is a filtered backstepping output.

Figure 4 shows the CFBS design that all procedures are described from Equations (64) to (72). **Figure 5** shows the detailed diagram for command filter block to saturate the magnitude and the rate of unfiltered signals of virtual control and backstepping output in their maximum values.

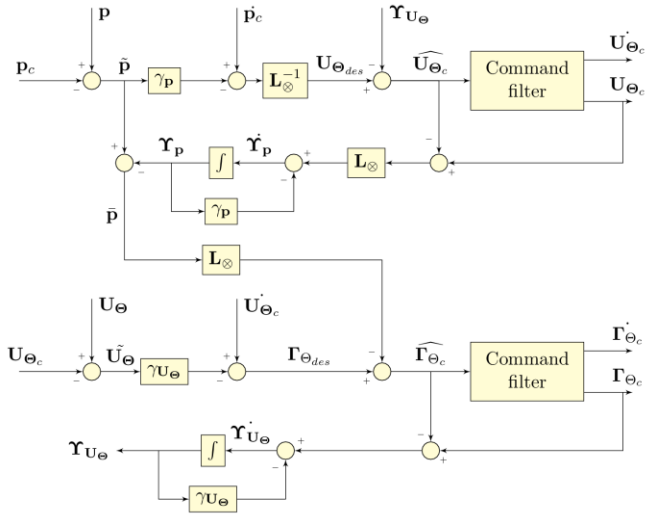


Figure 4 Command filtered backstepping control law design for camera gimbal IBVS

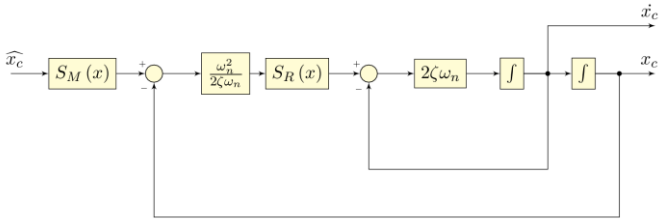


Figure 5 Command filter block

The command filter block is defined as follows:

$$x_c = q_1 \tag{73}$$

$$\dot{x}_c = q_2 \tag{74}$$

$$\dot{q}_1 = q_2 \tag{75}$$

$$\dot{q}_2 = 2\zeta\omega_n\left\{S_R\left(\frac{\omega_n}{2\zeta}\{S_M(\hat{x}_c) - q_1\}\right) - q_2\right\} \tag{76}$$

where ω_n and ζ respectively are natural frequency and damping ratio, S_M and S_R respectively are magnitude and rate limit function that is defined as follows:

$$S_M(x) = \begin{cases} M & x \geq M \\ x & |x| < M \\ -M & x \leq -M \end{cases} \tag{77}$$

$$S_R(x) = \begin{cases} R & x \geq R \\ x & |x| < R \\ -R & x \leq -R \end{cases} \tag{78}$$

where M and R respectively are magnitude and rate limit values.

In this paper, the natural frequency can be determined by time-sampling of digital computation and damping ratio of 3. Magnitude and rate limit for command filter $\mathbf{U}_{\boldsymbol{\Theta}}$ are the physical characteristics of the utilized servo motors.

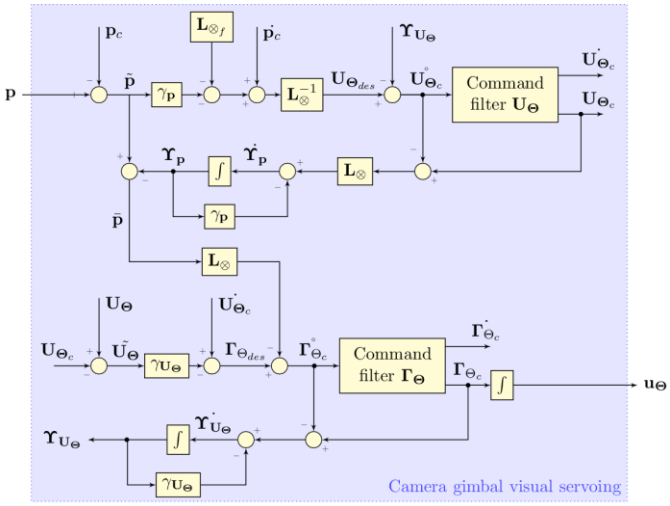


Figure 6 CFBS control law for camera gimbal IBVS with gimbal carrier dynamics

C. *Camera Gimbal Visual Servoing*

Finally, the camera gimbal IBVS is designed using the complete model of Equation (40) to replace the Equation (44) that is used in the previous sub-section. If IBS control law is employed, the gimbal carrier dynamics term in Equation (40) must be derived as analytical time-differential. Fortunately, CFBS simplifies the control design to eliminate the analytical time-differential procedures. **Figure 6** shows the design of CFBS control law for camera gimbal IBVS with gimbal carrier dynamics. A new block of gimbal carrier dynamics $\mathbf{L}_{\otimes_f}$ that is defined in the following is only added without time-differential process.

$$\mathbf{L}_{\otimes_f} = \mathbf{L}(\mathbf{p}, z)\ {}^{c}\mathbf{R}_{v}^{\boldsymbol{\Omega}}\begin{bmatrix}\mathbf{V} \\ \boldsymbol{\Omega}\end{bmatrix} \tag{79}$$

Comparing to the previous sub-section, the stabilization function of Equation (66) becomes to be

$$\mathbf{U}_{\boldsymbol{\Theta}_{des}} = \mathbf{L}_{\otimes}^{-1}\left(-\gamma_{\mathbf{p}}\tilde{\mathbf{p}} + \dot{\mathbf{p}}_c - \mathbf{L}_{\otimes_f}\right) \tag{80}$$

The appropriate pan-tilt control input $\mathbf{u}_{\boldsymbol{\Theta}}$ can be obtained from commanded backstepping output $\boldsymbol{\Gamma}_{\boldsymbol{\Theta}_c}$ using time-integral transfer function to reduce the image feature residual.

IV. Simulation Results

Table I shows the characteristics of camera image sensor that is used in simulation. **Figure 7** shows the simulation scenario of the gimbal carrier in inertia frame $\langle x_{north}, y_{east}, z_{down} \rangle$. The gimbal carrier of fixed-wing UAV is cruising westward in fixed velocity. Initial pan-tilt angles set the camera to not aim the target object directly but the image feature of the target object is always available in camera image plane.

Table I Camera sensor characteristics

Description	Parameter	Value
Focal length	λ	3.5 mm
Field of view	FoV	54.2°
Sensor cell horizontal length	H_{cell}	5.15 μm
Sensor cell vertical length	V_{cell}	4.7 μm
Effective horizontal pixel number	H_{pix}	702
Effective vertical pixel number	V_{pix}	575
Effective horizontal length	H_{length}	3.6 mm
Effective vertical length	V_{length}	2.7 mm

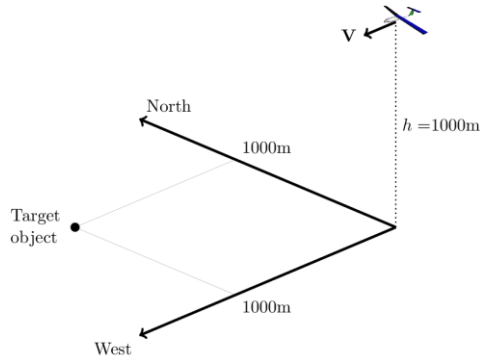


Figure 7 Simulation scenario of the gimbal carrier

A. *IBS and CFBS Comparison*

Table II IBS and CFBS comparison parameters

Description	Param	IBS	CFBS
UAV velocity	**V**	**0**	**0**
UAV angular velocity	**Ω**	**0**	**0**
UAV attitude	ϕ	0°	0°
	θ	0°	0°
	ψ	−90°	−90°
Initial pan-tilt angles	θ_t	−20°	−20°
	ψ_p	20°	20°
Initial UAV position	North	0 m	0 m
	East	0 m	0 m
	Down	−1000 m	−1000 m
Target object position	North	1000 m	1000 m
	East	−1000 m	−1000 m
	Down	0 m	0 m
Backstepping gain	$\gamma_{\mathbf{p}}$	2	2
	$\gamma_{\mathbf{z}}$	4	–
	$\gamma_{\mathbf{U_\Theta}}$	–	4
Frequency ($\omega_n/2\pi$)	f	–	25 Hz
Damping ratio	ζ	–	3
Command filter $\mathbf{U_\Theta}$	M	–	300°/s
	R	–	300°/s²
Command filter $\mathbf{\Gamma_\Theta}$	M	–	300°/s²
	R	–	∞

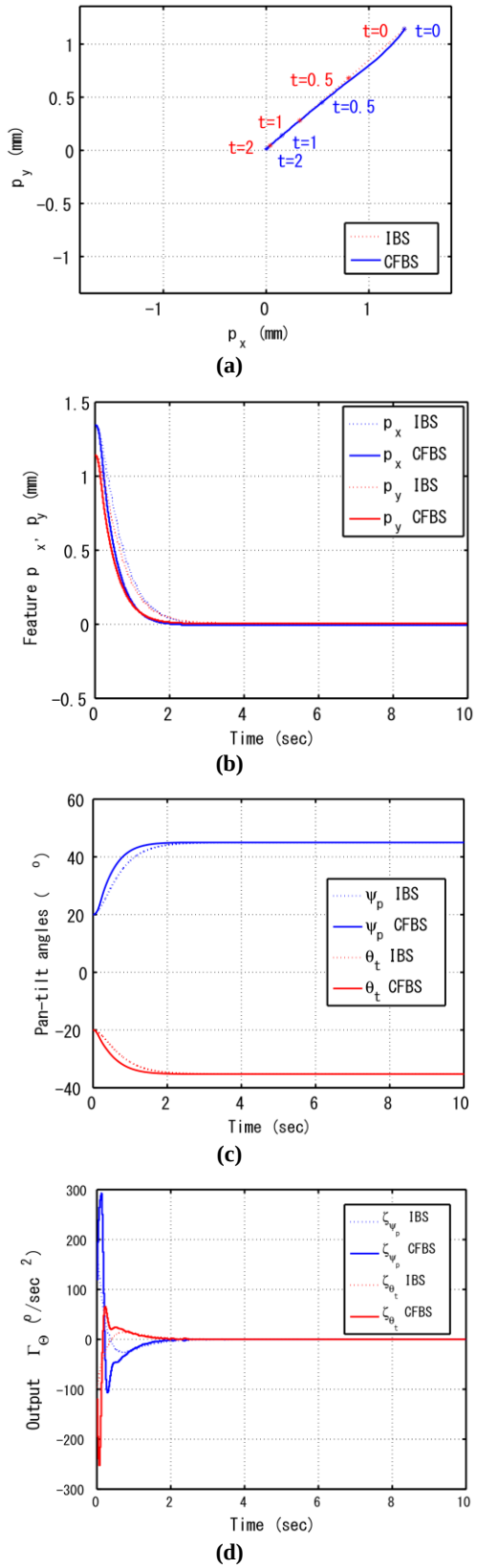
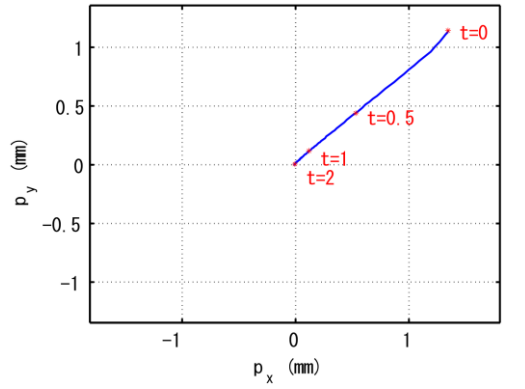
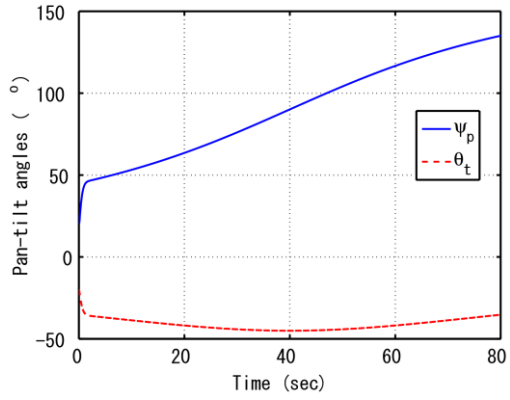


Figure 8 IBS and CFBS comparison. (a) Feature in camera image plane, (b) feature residual, (c) camera gimbal mechanism pan-tilt angles, and (d) backstepping output

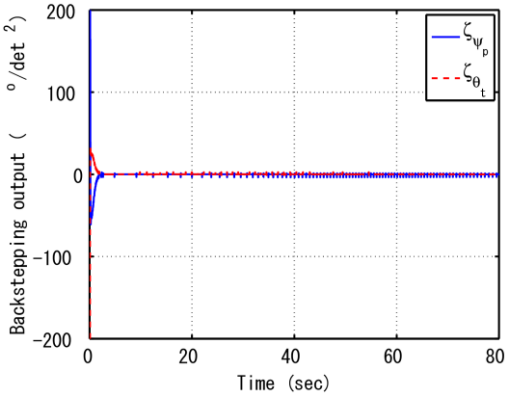
Both IBS and CFBS control law algorithm using simplified model of Equation (44) for camera gimbal IBVS are built to demonstrate each performance for reducing the feature residual. **TABLE II** shows the simulation parameters for comparing between IBS and CFBS performances. **Figure 8** shows the comparison between IBS and CFBS control law to reduce the image feature residual by moving the camera gimbal pan-tilt. It shows that CFBS control law performaces are better than IBS.



(a)

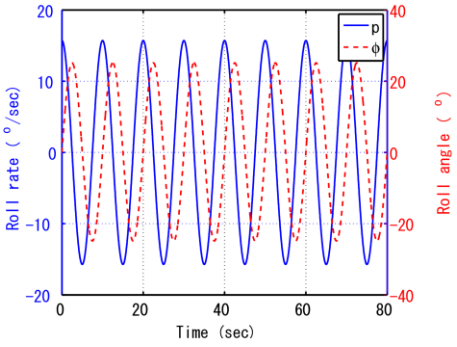


(b)

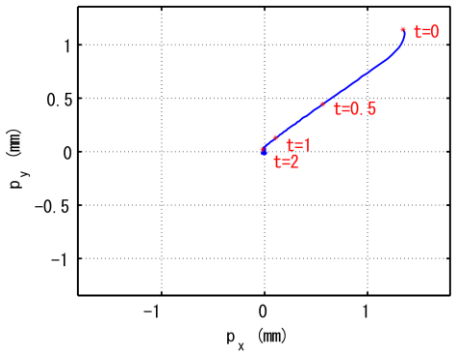


(c)

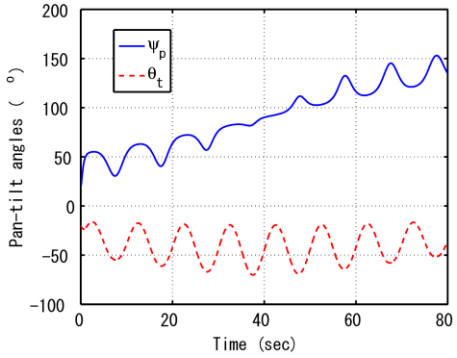
Figure 9 Camera gimbal IBVS simulation with gimbal carrier movements. (a) Feature in camera image plane; (b) camera gimbal mechanism pan-tilt angles; and (c) Backstepping output



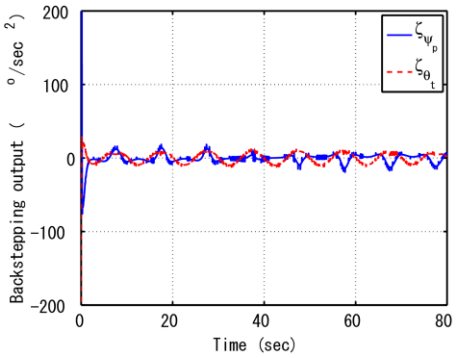
(a)



(b)



(c)



(d)

Figure 10 Camera gimbal IBVS simulation with extreme gimbal carrier movements. (a) Gimbal carrier roll and roll rate, (b) feature in camera image plane, (c) camera gimbal mechanism pan-tilt angles, and (d) backstepping output

B. Camera Gimbal Visual Servoing Simulation

The complete model of Equation (40) is then employed for developing CFBS control law algorithm. **TABLE III** shows the simulation parameters for demonstrating the camera gimbal IBVS performances. Other simulation parameters are described in **TABLE II**. **Figure 9** shows the simulation results of camera gimbal IBVS with gimbal carrier movements. There is no extreme movement of roll-shaking to show that the feature residual converges fast and the pan-tilt angles move well. The small spikes in backstepping output show the effort of CFBS control law to aim the camera pointing to the target object.

TABLE III Camera gimbal IBVS simulation parameters

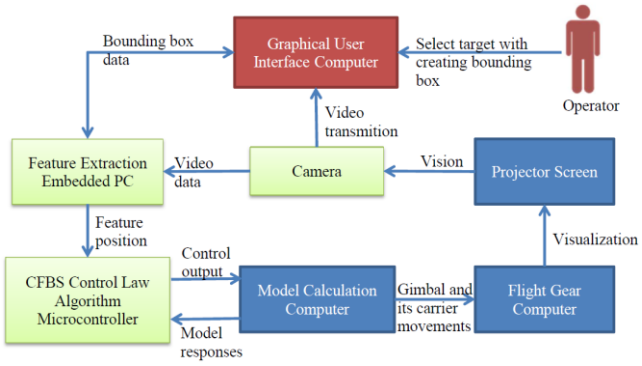
Description	Param	Value
UAV velocity	$\mathbf{V}$	$[25 \quad 0 \quad 0]^{\mathrm{T}}$ m/s
Initial UAV angular velocity	$\boldsymbol{\Omega}$	$\mathbf{0}$
Initial UAV attitude	ϕ	0^{o}
	θ	0^{o}
	ψ	-90^{o}

Figure 10 shows the simulation results of camera gimbal IBVS with extreme gimbal carrier movements. The gimbal carrier encounters roll-shaking movement as shown in **Figure 10(a)**. However the feature residual still converges well as shown in **Figure 10(b)**. **Figure 10(c)** and **(d)** show the effort of camera gimbal mechanism to maintain the camera direction always pointing to the target object.

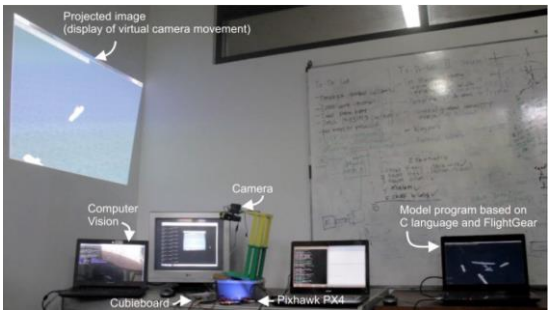
C. Hardware-In-the-Loop Simulation

Figure 11 shows the HILS configurations to test the real hardwares that will be installed on the gimbal carrier. **Figure 11(a)** shows block diagrams with description as follows: (i) the green blocks are on-board hardware components; (ii) the red blocks is on-ground hardware components; and (iii) the blue blocks are simulation hardware tools of HILS. **Figure 11(b)** shows laboratory HILS experiments for IBVS test. **Figure 11(c)** shows that the on-board hardwares are installed on a gimbal carrier test platform of fixed-wing UAV.

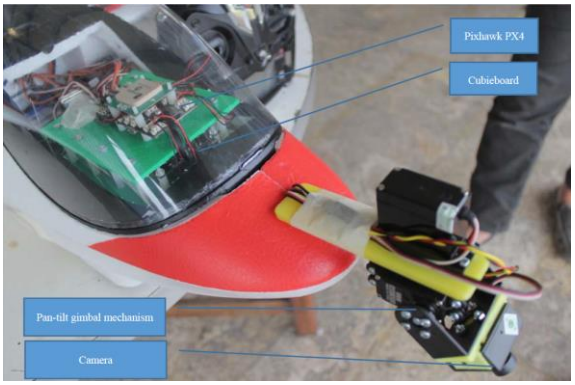
An on-board camera captures projected images that from a projector screen and distributes video signals to (1) an on-ground graphical user interface (GUI) computer through video transmitter and (2) an embedded PC of Cubieboard that calculates the feature extraction algorithm of TLD. TLD needs an initial reference bounding box image as an image reference of suspicous unknown target object that is selected by an operator through GUI computer. TLD feature extraction produces current feature position in image plane which is assigned to a microcontroller of Pixhawk PX4 for further IBVS calculation and is also transmitted back to GUI computer for the image feature monitor on GUI screen. The microcontroller then calculates the camera gimbal IBVS using CFBS control law with the input requirements of current feature position and gimbal/carrier responses to produce control output. A computer then calculates the model of IBVS and the gimbal carrier to yield model responses and the movements of gimbal and its carrier. Flight gear software is then visualize a virtual camera to the projector screen. The attitude of camera in world cordinate frame is obtained from the movements calculation of gimbal and its carrier from model calculation computer.



(a)



(b)



(c)

Figure 11 HILS configurations. (a) Block diagrams of HILS, (b) laboratory HILS experiments, and (c) test platform

Cubieboard 2 with dual-core arm cortex-A7 processor and 1GB RAM is employed for calculating the feature extration algorithm of TLD. Linaro lubuntu is installed on cubieboard as its operating system. OpenTLD that is developed by Georg Nebehay et al. [15] is then implemented on linaro lubuntu to produce feature from the target object. Pixhawk PX4 with arm cortex-M4F 168Mhz is employed to calculate the camera gimbal IBVS algorithm using CFBS control law. NuttX real-time operating system is used to develop a new software module that implements the camera gimbal IBVS algorithm using CFBS.

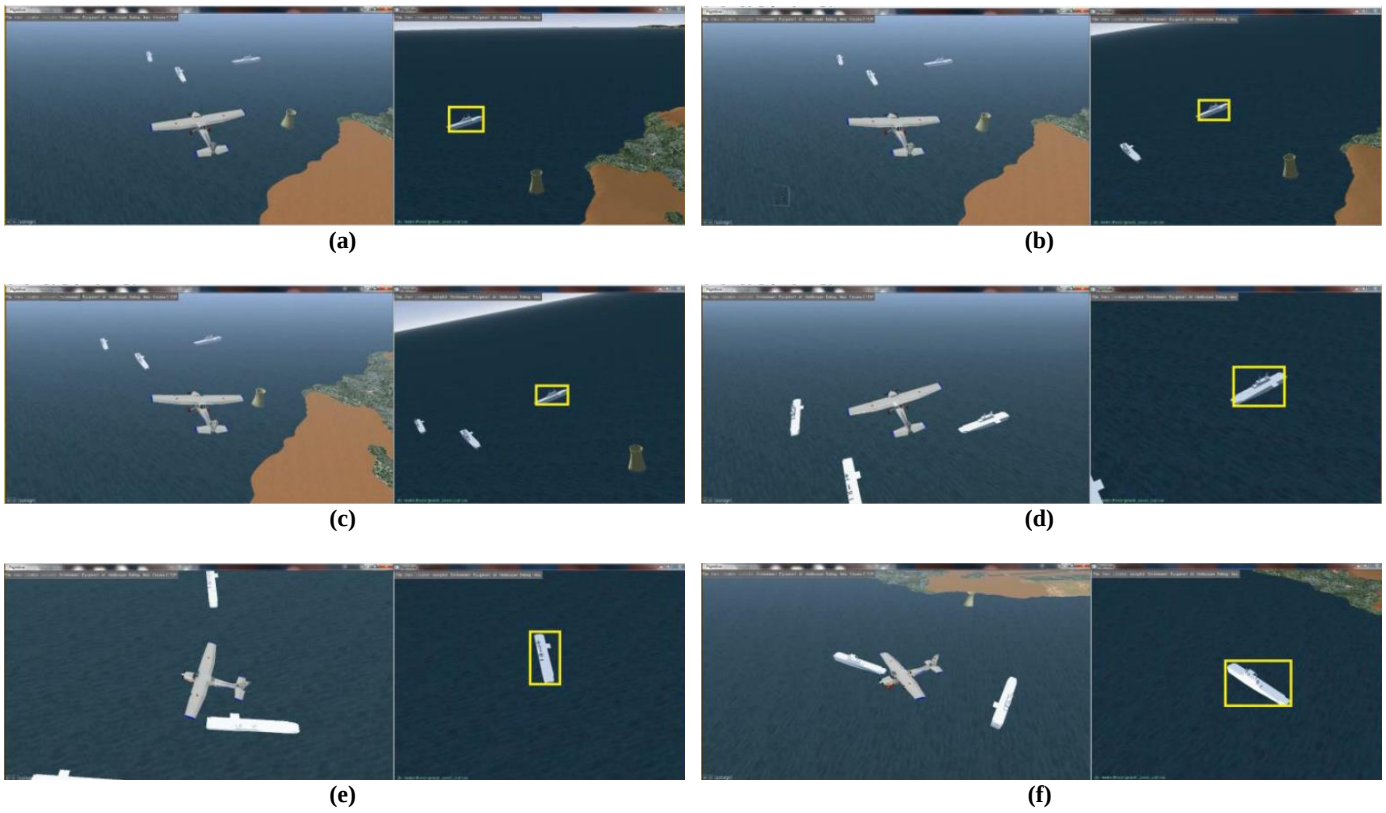


Figure 12 Hardware in the loop simulation video (left hand is gimbal carrier attitude and right hand is virtual camera view). (a) Initial feature extraction, (b) pan-tilt movement to make feature position toward the center of the image, (c)(d)(e)(f) keep the feature position at the center of image despite of the movement of the carrier

Figure 12 shows the HILS video that proves the effectiveness of the proposed nonlinear camera gimbal IBVS using CFBS control law. **Figure 12(a)** shows the initial pose of camera view and gimbal carrier attitude. The feature position is not at the center of camera image screen. **Figure 12(b)** shows the effort of pan-tilt gimbal mechanism to make feature position at the center of camera image screen. **Figure 12(c)**, **(d)**, **(e)**, and **(f)** show that the camera gimbal IBVS using CFBS control law can maintain the feature position at the center of the camera image screen even in the radical movements of the gimbal carrier.

V. Conclusion

This work presented a nonlinear camera gimbal visual servoing using CFBS control law. The approach employed nonlinear model of image feature motion and gimbal pan-tilt mechanism movement. IBS and CFBS control law comparison shows that the CFBS is easier to design and better performance than IBS. The radical movement of the gimbal carrier is also well handled using CFBS control law that adopts the gimbal carrier dynamics model. The simulation results show the effectiveness of the proposed nonlinear camera gimbal IBVS using CFBS control law to make feature residual always to be converged. SILS/HILS proves the camera gimbal IBVS effectiveness to maintain the feature position at the center of the image although the gimbal carrier move radically.

For the future work, a nonlinear model of the fixed-wing UAV movements including its aerodynamics will be developed. Then, a nonlinear CFBS control law will be designed for visual based autonomous flight controller by combining the fixed-wing UAV dynamics model with the proposed gimbal IBVS model in this paper.

Acknowledgment

The authors of this work gratefully acknowledge Widhi Surya Atman, Aditya Wildan Farras, Yaqub Aris and Primawan Dwi Nugroho for their technical assistance on the implementation of HILS. This work is supported by center for security and defense industry technology, BPPT Indonesia.

References

[1] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, July 2012. CrossRef

[2] S. Hutchinson, G. Hager, and P. Cork, "A tutorial on visual servo control," *IEEE Transactions on Robotics Automation*, vol. 12, no. 5, pp. 651-670, Oct. 1996. CrossRef

[3] F. Chaumette and S. Hutchinson, "Visual servo control, part I: basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82-90, 2006. CrossRef

[4] F. Chaumette and S. Hutchinson, "Visual servo control, part II: advanced approaches," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 109-118, 2007. CrossRef

[5] T. Hamel and R. Mahony, "Visual servoing of an under-actuated dynamic rigid-body system: An image-based approach," *IEEE Transactions on Robotics Automation*, vol. 18, no. 2, pp. 187-198, 2002. CrossRef

[6] O. Bourquardez, R. Mahony, N. Guenard, F. Chaumette, T. Hamel, and L. Eek, "Image-based visual servo control of the translation kinematics of a quadrotor aerial vehicle," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 743-749, June 2009. CrossRef

[7] F. Le Bras, T. Hamel, C. Barat, and R. Mahony, "Image-based visual servo controller for automatic landing guidance of a fixed-wing aircraft," *IEEE European Control Conference*, pp. 1836-1841, Aug. 23-26, 2009.

[8] F. Le Bras, T. Hamel, and R. Mahony, "Image-based visual servo control for circular trajectories for a fixed-wing aircraft," *Proceeding of the 48th IEEE Conf. on Decision and Control*, pp. 3430 - 3435, Dec. 15-18, 2009. CrossRef

[9] P. Peliti, L. Rosa, G. Oriolo, and M. Vendittelli, "Vision-based loitering over a target for a fixed-wing UAV," Proc. of IFAC Sym. on Robot Control, pp. 51-57, 2012.

[10] Hasan K. Khalil, *Nonlinear Systems*. New Jersey: Prentice Hall, 2002.

[11] Miroslav Krstic, Ioannis Kanellakopoulos, and Petar Kokotovic, *Nonlinear and Adaptive Control Design*. New York: John Willey & Sons, 1995.

[12] Jay A. Farrell and Marios M. Polycarpou, *Adaptive Approximation Based Control*. New York: John Willey & Sons, 2006.

[13] Jay A. Farrel, M. M. Polycarpou, and M. Sharma, "Command Filtered Backstepping," *IEEE Transaction on Automatic Control*, vol. 54, no. 6, pp. 1391-1395, June 2009. CrossRef

[14] Peter Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Berlin Heidelberg: Springer Publishing, 2013.

[15] Georg Nebehay, Branislav Micusik, Cristina Picus, and Roman Pflugfelder, *Evaluation of an online learning approach for robust object tracking*. Technical Report AIT-DSS-TR-0279, AIT Austrian Institute of Technology, 2011.