Proceedings of the 9th International Workshop
on Robot Motion and Control, Wasowo Palace,
Wasowo, Poland, July 3-5, 2013

Vision-based Switching Control Strategy for a Nonholonomic Wheeled Mobile Robot with a Pan Camera

Masahide Ito[1], Takafumi Ebata[2], and Masaaki Shibata[2]

Abstract— **This paper presents a vision-based switching control strategy for a nonholonomic wheeled mobile robot with a pan camera (NHWMR+PC). The control inputs of the NHWMR+PC consist of a pan rotational velocity of the camera and linear and angular velocities of the platform. When we adopt all three inputs for image-based visual servoing (IBVS), the standard IBVS controller does not work at all due to the singularity of the interaction matrix which depends on the configuration of the camera. For avoiding the singularity, we propose a switching control strategy which consists of two control modes: one is for IBVS using only two control inputs, and the other for regulating the platform orientation and the pan camera angle. The effectiveness of this strategy was validated through a simulation.**

I. Introduction

Robotic systems need the ability to understand their workspace to behave autonomously. For this purpose, a camera is a very useful sensory device to provide a vision function for robots. An image captured via the camera contains a vast amount of information of the workspace. Introducing visual information extracted from an image into a robot control loop has the potential to increase the flexibility and accuracy of a given task. In particular, feedback control with visual information, so-called *visual servoing* or *visual feedback control*, is a considerable technique for robotic systems [1]–[4].

Visual servoing is classified into three approaches: position-based visual servoing (PBVS), image-based visual servoing (IBVS), and hybrid visual servoing (HVS). The difference between the PBVS and IBVS approaches depends on how to use visual features of a target object which are extracted from an image. One advantage of the IBVS approach over the PBVS approach is robustness against modeling errors and external disturbances. The HVS approach, e.g., 2-1/2-D visual servoing [5], partitioned visual servoing [6], and Deguchi's method [7], incorporates the advantage of both PBVS and IBVS approaches.

The IBVS controller is well-implemented for various robotic systems in the eye-in-hand configuration, e.g., manipulators, wheeled mobile robots, and aerial vehicles. Focusing on IBVS for wheeled mobile robots (WMRs), the simplest system structure is consisted of a fixed camera and a platform with two independent motorized wheels. However, due to kinematic nonholonomic constraints, it is difficult to achieve an IBVS task. One way to reduce the affect is to interchange the fixed camera with the active camera such as the pan camera and the pan tilt camera.

A vision-based control problem for a nonholonomic WMR with a pan camera (NHWMR+PC) is discussed in some literature [8]–[10]. In [8], a biologically-inspired controller with the optical flow and geometrical consideration achieve positioning a WMR to a target object at a certain location with a certain heading, which is called *docking*. Komada, *et al*. [9] propose a vision-based navigation method which consists of two parts: one is an IBVS controller for the pan angle of the camera and another is velocity controller with geometrical information for the position and orientation of the platform. Focusing on the difficulty in keeping the target object in the field of view (FOV), *i.e.*, the so-called FOV problem, Fang, *et al*. [10] present an active visual servoing strategy combined with an adaptive vision tracking controller for the camera and a switched controller for the platform configuration.

In this paper, we present a vision-based switching strategy for a NHWMR+PC. The control inputs of the NHWMR+PC consist of a pan rotational velocity of the camera and linear and angular velocities of the platform. When we adopt all three inputs for IBVS, the standard IBVS controller does not work at all due to the singularity of the interaction matrix which depends on the configuration of the camera. For avoiding the singularity, we propose a switching control strategy which consists of two control modes: one is for IBVS using only two control inputs, and the other for regulating the platform orientation and the pan camera angle. The proposed strategy does not require the geometrical information with respect to the target object over the conventional methods [8]–[10]. The effectiveness of this strategy was validated through a simulation.

The rest of the paper is organized as follows. The next section introduces kinematic models of a NHWMR+PC to provide two kinds of Jacobian matrices. In Section III, we present a vision-based switching control strategy as a main result of this paper. In Section IV, simulation results are shown to validate the proposed strategy. Finally, we conclude the paper in Section V.

Notation: We denote the $i \times i$ identity matrix as $\boldsymbol{I}_i$. The estimated value of a variable, vector and matrix is represented by $\hat{\bullet}$.

[1]M. Ito is with School of Information Science and Technology, Aichi Prefectural University, Aichi 480–1198, Japan; Phone: +81-561-76-8600, Fax: +81-561-64-1108; Email: masah-ito@ist.aichi-pu.ac.jp

[2]T. Ebata and M. Shibata are with Department of System Design Engineering, Seikei University, Tokyo 180–8633, Japan

978-1-4673-5511-7/13/$31.00 ©2013 IEEE

II. Models of a Nonholonomic Wheeled Mobile Robot with a Pan Camera

This section provides two models of a NHWMR+PC: the vision system and the robot system models. Two kinds of Jacobian matrices given from these models play important role in IBVS. In this paper, the target object is assumed to be static.

A. Vision System Model

A target object is equipped with k markers. We refer to the markers as feature points on the image plane. The geometric relation between the camera and the i-th marker is depicted in Fig. 1, where the coordinate frames Σ_w, Σ_c, Σ_s, and Σ_f represent the world frame, the camera frame, the standard camera frame, and the image plane frame, respectively. The frame Σ_s is static on Σ_w at a given time, which measures the velocity of Σ_c. The frame Σ_f is located at the center of the image plane. Let ${}^s\boldsymbol{p}_{o_i} := [\,{}^sx_{o_i},\, {}^sy_{o_i},\, {}^sz_{o_i}\,]^\top$, ${}^s\boldsymbol{p}_c := [\,{}^sx_c,\, {}^sy_c,\, {}^sz_c\,]^\top$, and ${}^c\boldsymbol{p}_{o_i} := [\,{}^cx_{o_i},\, {}^cy_{o_i},\, {}^cz_{o_i}\,]^\top$ be the position vectors of the i-th marker on Σ_s, the camera on Σ_s, and the i-th marker on Σ_c, respectively. The coordinates of the i-th feature point on Σ_f are denoted as $\boldsymbol{f}_i = [\,u_i,\, v_i\,]^\top$.

When the camera moves at a certain translational and angular velocity for a stationary target object on Σ_s (i.e., ${}^s\dot{\boldsymbol{p}}_{o_i} \equiv \boldsymbol{0}_3$), the translational velocity of the i-th marker on Σ_c can be represented as

$$
\begin{aligned}
{}^c\dot{\boldsymbol{p}}_{o_i} &= -{}^s\dot{\boldsymbol{p}}_c - {}^s\boldsymbol{\omega}_c \times {}^c\boldsymbol{p}_{o_i} \\
&= \left[\begin{array}{c|ccc} & 0 & -{}^cz_{o_i} & {}^cy_{o_i} \\ -\boldsymbol{I}_3 & {}^cz_{o_i} & 0 & -{}^cx_{o_i} \\ & -{}^cy_{o_i} & {}^cx_{o_i} & 0 \end{array}\right] \left[\begin{array}{c} {}^s\dot{\boldsymbol{p}}_c \\ \hline {}^s\boldsymbol{\omega}_c \end{array}\right], \qquad (1)
\end{aligned}
$$

where ${}^s\boldsymbol{\omega}_c$ is the angular velocity of the camera on Σ_s. We assume an ideal pinhole camera model as an imaging model. Then, from the perspective projection, the i-th feature point can be expressed as

$$
\boldsymbol{f}_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} := \frac{1}{{}^cz_{o_i}} \begin{bmatrix} \lambda_x {}^cx_{o_i} \\ \lambda_y {}^cy_{o_i} \end{bmatrix}, \qquad (2)
$$

where λ_x and λ_y are the horizontal and vertical focal lengths of the camera, respectively. Furthermore, differentiating (2) with respect to time, the motion of the i-th feature point can be described as

$$
\dot{\boldsymbol{f}}_i = \frac{1}{{}^cz_{o_i}} \begin{bmatrix} \lambda_x & 0 & -\lambda_x \frac{{}^cx_{o_i}}{{}^cz_{o_i}} \\ 0 & \lambda_y & -\lambda_y \frac{{}^cy_{o_i}}{{}^cz_{o_i}} \end{bmatrix} {}^c\dot{\boldsymbol{p}}_{o_i}. \qquad (3)
$$

By using (1) and (3), we obtain

$$
\dot{\boldsymbol{f}}_i = \boldsymbol{J}_{\rm img}(\boldsymbol{f}_i, {}^cz_{o_i})\, {}^s\boldsymbol{w}_c, \qquad (4)
$$

where $\boldsymbol{J}_{\rm img} = \left[\begin{array}{c:c} \boldsymbol{J}_{\rm img}^{(1,1)} & \boldsymbol{J}_{\rm img}^{(1,2)} \end{array}\right] \in \mathbb{R}^{2\times 6}$,

$$
\boldsymbol{J}_{\rm img}^{(1,1)}(\boldsymbol{f}_i, {}^cz_{o_i}) := \begin{bmatrix} -\frac{\lambda_x}{{}^cz_{o_i}} & 0 & \frac{u_i}{{}^cz_{o_i}} \\ 0 & -\frac{\lambda_y}{{}^cz_{o_i}} & \frac{v_i}{{}^cz_{o_i}} \end{bmatrix},
$$

$$
\boldsymbol{J}_{\rm img}^{(1,2)}(\boldsymbol{f}_i) := \begin{bmatrix} \frac{1}{\lambda_y}u_iv_i & -\left(\lambda_x + \frac{u_i^2}{\lambda_x}\right) & \frac{\lambda_x}{\lambda_y}v_i \\ \lambda_y + \frac{v_i^2}{\lambda_y} & -\frac{1}{\lambda_x}u_iv_i & -\frac{\lambda_y}{\lambda_x}u_i \end{bmatrix},
$$

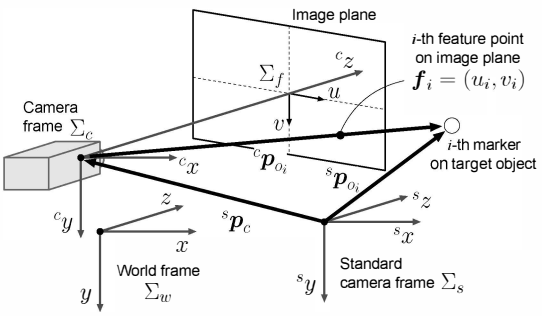


Fig. 1: Vision system model.

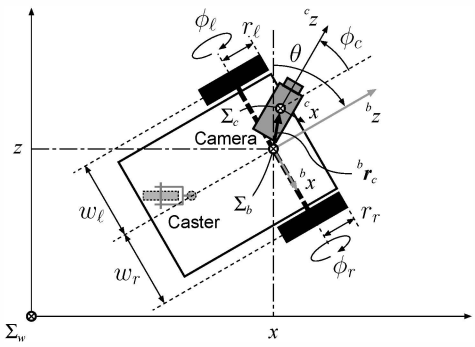


Fig. 2: Robot system model.

TABLE I: Parameters of the robot.

(x, y)	: origin coordinates of Σ_b on Σ_w
θ	: angle of the traveling direction w.r.t. the z-axis (*i.e.* positive direction of the ${}^bz$-axis w.r.t. the z-axis)
ϕ_r, ϕ_ℓ	: rotating angles of the right and left wheels
ϕ_c	: pan rotating angle of the camera w.r.t. the ${}^bz$-axis ($\boldsymbol{\phi} := [\,\phi_r,\, \phi_\ell,\, \phi_c\,]^\top$)
w_r, w_ℓ	: distance between the origin of Σ_b and the right/left wheel
${}^b\boldsymbol{r}_c$	: position vector of Σ_c on Σ_b (${}^b\boldsymbol{r}_c := [\,{}^bx_c,\, {}^bz_c\,]^\top$)
r_r, r_ℓ	: radii of the right and left wheels
ν_1	: longitudinal velocity of the platform in Σ_b
ν_2	: angular velocity of the platform around the ${}^by$-axis
ν_3	: angular velocity of the camera around the ${}^cy$-axis

and ${}^s\boldsymbol{w}_c := [\,{}^s\dot{\boldsymbol{p}}_c^\top,\, {}^s\boldsymbol{\omega}_c^\top\,]^\top$ is called the velocity twist of the camera. Summarizing (4) in terms of k feature points, we obtain

$$
\dot{\boldsymbol{f}} = \bar{\boldsymbol{J}}_{\rm img}(\boldsymbol{f}, {}^c\boldsymbol{z}_o)\, {}^s\boldsymbol{w}_c, \qquad (5)
$$

where $\boldsymbol{f} := [\,\boldsymbol{f}_1^\top, \cdots, \boldsymbol{f}_k^\top\,]^\top \in \mathbb{R}^{2k}$, ${}^c\boldsymbol{z}_o := [\,{}^cz_{o_1}, \cdots, {}^cz_{o_k}\,]^\top \in \mathbb{R}^k$, $\bar{\boldsymbol{J}}_{\rm img} := [\,\boldsymbol{J}_{\rm img}^\top(\boldsymbol{f}_1, {}^cz_{o_1}), \cdots, \boldsymbol{J}_{\rm img}^\top(\boldsymbol{f}_k, {}^cz_{o_k})\,]^\top \in \mathbb{R}^{2k\times 6}$. The matrix $\bar{\boldsymbol{J}}_{\rm img}$ is the so-called *interaction matrix* or *image Jacobian matrix*.

B. Robot System Model

The model of the NHWMR+PC is depicted in Fig. 2. The right and left wheels and the camera, which are actuated independently, are provided at the front of the robot. A passively revolving caster is also provided at the rear of the robot. The coordinate frames Σ_w and Σ_c are the same world and camera frames as in Fig. 1. The vehicle body frame Σ_b

is located at the bottom of the perpendicular line from the barycenter of the vehicle body to the axle. Note that y- and ${}^b y$-axes are not considered in modeling in this section. The parameters are defined as in Table I.

Assuming that each wheel does not skid or slip on the ground, the origin of Σ_b can move only along the ${}^b z$-axis instantaneously. This can be described by the formula

$$\dot{x} = \nu_1 \sin\theta \quad \text{and} \quad \dot{z} = \nu_1 \cos\theta, \tag{6}$$

or the equivalent expression

$$\dot{x} = \dot{z}\tan\theta \quad \Leftrightarrow \quad \dot{x}\cos\theta - \dot{z}\sin\theta = 0. \tag{7}$$

Equation (7) is a non-integrable differential equation which is described by the position and velocity, a so-called *non-holonomic constraint.*

We define the state and input variables as $(\, x,\, z,\, \theta,\, \phi_c \,)$ and $(\, \nu_1,\, \nu_2,\, \nu_3 \,)$, respectively. Using (6), the motion of equation of the robot can be written as

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x \\ z \\ \theta \\ \phi_c \end{bmatrix} = \begin{bmatrix} \sin\theta & 0 & 0 \\ \cos\theta & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{\nu}, \tag{8}$$

where $\boldsymbol{\nu} = [\, \nu_1,\, \nu_2,\, \nu_3 \,]^\top$. The velocity twist of the camera ${}^s\boldsymbol{w}_c$ is associated with the input $\boldsymbol{\nu}$ as follows:

$${}^s\boldsymbol{w}_c = \underbrace{\begin{bmatrix} -\sin\phi_c & {}^b x_c \sin\phi_c + {}^b z_c \cos\phi_c & 0 \\ 0 & 0 & 0 \\ \cos\phi_c & -{}^b x_c \cos\phi_c + {}^b z_c \sin\phi_c & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}}_{{}^c\boldsymbol{J}(\phi_c)} \boldsymbol{\nu}. \tag{9}$$

Also, the geometric relation between $\boldsymbol{\nu}$ and $\dot{\boldsymbol{\phi}}$ is expressed as:

$$\boldsymbol{\nu} = \underbrace{\begin{bmatrix} \frac{w_\ell r_r}{w_r + w_\ell} & \frac{w_r r_\ell}{w_r + w_\ell} & 0 \\ \frac{r_r}{w_r + w_\ell} & -\frac{r_\ell}{w_r + w_\ell} & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{T}_w} \dot{\boldsymbol{\phi}}.$$

Note that the column rank of the *robot Jacobian matrix* ${}^c\boldsymbol{J}(\phi_c)$ depends on the value of ${}^b z_c$, *i.e.*, the camera configuration on the platform. If ${}^b z_c \neq 0$, then $\operatorname{rank} {}^c\boldsymbol{J}(\phi_c) = 3 \ \forall \phi_c$, *i.e.*, ${}^c\boldsymbol{J}$ is full column rank; If ${}^b z_c = 0$, then $\operatorname{rank} {}^c\boldsymbol{J}(\phi_c) = 2 \ \forall \phi_c$. The former case means that we cannot design a standard IBVS controller using the pseudo-inverse of the composite matrix $\bar{\boldsymbol{J}}_{\mathrm{img}} {}^c\boldsymbol{J}$ (or $\bar{\boldsymbol{J}}_{\mathrm{img}} {}^c\boldsymbol{J}\boldsymbol{T}_w$) when the camera is located along the ${}^b x$-axis. To overcome this difficulty, we present a switching control strategy in the next section.

III. Vision-based Switching Control Strategy

This section presents a switching control strategy for the case that the camera is located along the ${}^b x$-axis. In that case, ${}^c\boldsymbol{J}$ is described as

$${}^c\boldsymbol{J}(\phi_c) = \begin{bmatrix} -\sin\phi_c & {}^b z_c \cos\phi_c & 0 \\ 0 & 0 & 0 \\ \cos\phi_c & {}^b z_c \sin\phi_c & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix},$$

which implies that we can adopt $(\, \nu_1,\, \nu_2 \,)$ or $(\, \nu_1,\, \nu_3 \,)$ as control inputs. These control inputs are defined as $\boldsymbol{\nu}_{12} := [\, \nu_1,\, \nu_2 \,]^\top$ and $\boldsymbol{\nu}_{13} := [\, \nu_1,\, \nu_3 \,]^\top$. Also, the related Jacobian matrices are defined as

$${}^c\boldsymbol{J}_{12}(\phi_c) := \begin{bmatrix} -\sin\phi_c & {}^b z_c \cos\phi_c \\ 0 & 0 \\ \cos\phi_c & {}^b z_c \sin\phi_c \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad \text{for} \quad \boldsymbol{\nu}_{12}, \tag{10}$$

and

$${}^c\boldsymbol{J}_{13}(\phi_c) := \begin{bmatrix} -\sin\phi_c & 0 \\ 0 & 0 \\ \cos\phi_c & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad \text{for} \quad \boldsymbol{\nu}_{13}. \tag{11}$$

Both matrices are full column rank without depending on values of ${}^b z_c$ and ϕ_c. Furthermore, if ${}^b z_c = 0$, then ${}^c\boldsymbol{J}_{12} = {}^c\boldsymbol{J}_{13}$.

Limiting the original control input $\boldsymbol{\nu}$ to $\boldsymbol{\nu}_{12}$ or $\boldsymbol{\nu}_{13}$, we can design the IBVS control system. Now we select $\boldsymbol{\nu}_{13}$ for IBVS, because selecting $\boldsymbol{\nu}_{12}$ for IBVS means that the camera is fixed on the platform.

From (5) and (11), $\dot{\boldsymbol{f}}$ is associated with $\boldsymbol{\nu}_{13}$ as follows:

$$\dot{\boldsymbol{f}} = \boldsymbol{J}_{\mathrm{vis}}(\boldsymbol{f}, {}^c\boldsymbol{z}_o, \phi_c)\boldsymbol{\nu}_{13}, \tag{12}$$

where $\boldsymbol{J}_{\mathrm{vis}}(\boldsymbol{f}, {}^c\boldsymbol{z}_o, \phi_c) := \bar{\boldsymbol{J}}_{\mathrm{img}}(\boldsymbol{f}, {}^c\boldsymbol{z}_o) {}^c\boldsymbol{J}_{13}(\phi_c) \in \mathbb{R}^{2k\times 2}$. Considering a minimization problem

$$\min_{\boldsymbol{\nu}_{13}} \left\| \, \dot{\boldsymbol{f}} - \boldsymbol{J}_{\mathrm{vis}}(\boldsymbol{f}, {}^c\boldsymbol{z}_o, \phi_c)\, \boldsymbol{\nu}_{13} \, \right\|,$$

a solution under the condition $\operatorname{rank} \boldsymbol{J}_{\mathrm{vis}} = 2$ is

$$\boldsymbol{\nu}_{13} = \boldsymbol{J}^+_{\mathrm{vis}}(\boldsymbol{f}, {}^c\boldsymbol{z}_o, \phi_c)\, \dot{\boldsymbol{f}}, \tag{13}$$

where $\boldsymbol{J}^+_{\mathrm{vis}}$ denotes the pseudo-inverse of $\boldsymbol{J}_{\mathrm{vis}}$. The matrix $\boldsymbol{J}^+_{\mathrm{vis}_m}$ is described as

$$\boldsymbol{J}^+_{\mathrm{vis}} = \begin{cases} \boldsymbol{J}^{-1}_{\mathrm{vis}}, & \text{for } \ k = 1 \\ (\boldsymbol{J}^\top_{\mathrm{vis}} \boldsymbol{J}_{\mathrm{vis}})^{-1} \boldsymbol{J}^\top_{\mathrm{vis}}, & \text{for } \ k > 1 \end{cases}.$$

Note that $\operatorname{rank} \boldsymbol{J}_{\mathrm{img}} = \operatorname{rank} {}^c\boldsymbol{J} = 2$ always holds for all ϕ_c, which implies that $\boldsymbol{J}_{\mathrm{vis}}$ is nonsingular without depending on the number of feature points.

The control objective is to make each feature point $\boldsymbol{f}_i$ converge into the desired one $\boldsymbol{f}^d_i$ $(i = 1, \ldots, k)$. To achieve this objective for the feature point dynamics

$$\dot{\boldsymbol{f}} = \boldsymbol{\mu}, \tag{14}$$

where $\boldsymbol{\mu}$ denotes the linear control input on Σ_f, we adopt the proportional controller

$$\boldsymbol{\mu} = \boldsymbol{K}_{\mathrm{img}}(\boldsymbol{f}^d - \boldsymbol{f}). \tag{15}$$

Then, the residual control input ν_2 is set to zero. From (13), (14) and (15), the control input $\boldsymbol{\nu}$ is therefore given by

$$\boldsymbol{\nu} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{J}^{+}_{\mathrm{vis}}(\boldsymbol{f}, {}^c\boldsymbol{z}_o, \phi_c)\, \boldsymbol{K}_{\mathrm{img}}(\boldsymbol{f}^d - \boldsymbol{f}), \tag{16}$$

where

$$\begin{aligned} \boldsymbol{K}_{\mathrm{img}} &= \text{block diag}\,\{\boldsymbol{K}_{\mathrm{img}\,1}, \cdots, \boldsymbol{K}_{\mathrm{img}\,k}\}, \\ \boldsymbol{K}_{\mathrm{img}\,i} &:= \mathrm{diag}\{K^u_{\mathrm{img}\,i}, K^v_{\mathrm{img}\,i}\} \succ 0, \\ \boldsymbol{f}^d &= [\,(\boldsymbol{f}^d_1)^\top, \cdots, (\boldsymbol{f}^d_k)^\top]^\top, \ \boldsymbol{f}^d_i := [\,u^d_i,\, v^d_i\,]^\top \\ & \qquad (i = 1, ..., k). \end{aligned}$$

The controller (16) can achieve the IBVS task, but the platform is constrained to move straight on ${}^bz$-axis. Consequently, after the IBVS task is completed, we introduce a mode that regulates the platform orientation so as to head the camera toward the forwarding direction of the platform. We design this mode by choosing the following proportional feedback controller:

$$\boldsymbol{\nu} = \begin{bmatrix} 0 \\ K_o\{(\bar{\theta} + \bar{\phi}_c) - \theta\} \\ -K_c\phi_c \end{bmatrix}, \tag{17}$$

where $\bar{\theta}$ and $\bar{\phi}_c$ are (constant) values of θ and ϕ_c at the time of completing the IBVS task. Design parameters $K_o > 0$ and $K_c > 0$ are proportional gains. The resultant motion by applying the controller (17) means to interchange the pan camera angle with the platform orientation.

Finally, the aforementioned discussion can be summarize as a switching control strategy which consists of two control modes:

- Mode I: The IBVS task is performed by using the controller (16).
- Mode II: The regulation of θ and ϕ_c is performed by using the controller (17).

Repeating switching of these two modes, it can be expected that each feature point converges to the desired one while the camera turns toward the forwarding direction of the platform.

In the next section, we verify the effectiveness of our proposed strategy via a simulation.

IV. Simulation

This section presents simulation results to show the validity of the proposed control strategy. Here we consider the case where the number of feature points is two, i.e., $k = 2$.

The image resolution and focal lengths of the camera are 640×480 pixels and $\lambda_x = \lambda_y = 128$ pixels. The frame rate of the camera is also 30 fps, i.e., the image data is updated every 33 ms. Also, the desired feature points are set to

$$\begin{aligned} \boldsymbol{f}^d_1 &= [\,40\,\text{pixels},\ 30\,\text{pixels}\,]^\top, \\ \boldsymbol{f}^d_2 &= [\,60\,\text{pixels},\ 20\,\text{pixels}\,]^\top. \end{aligned}$$

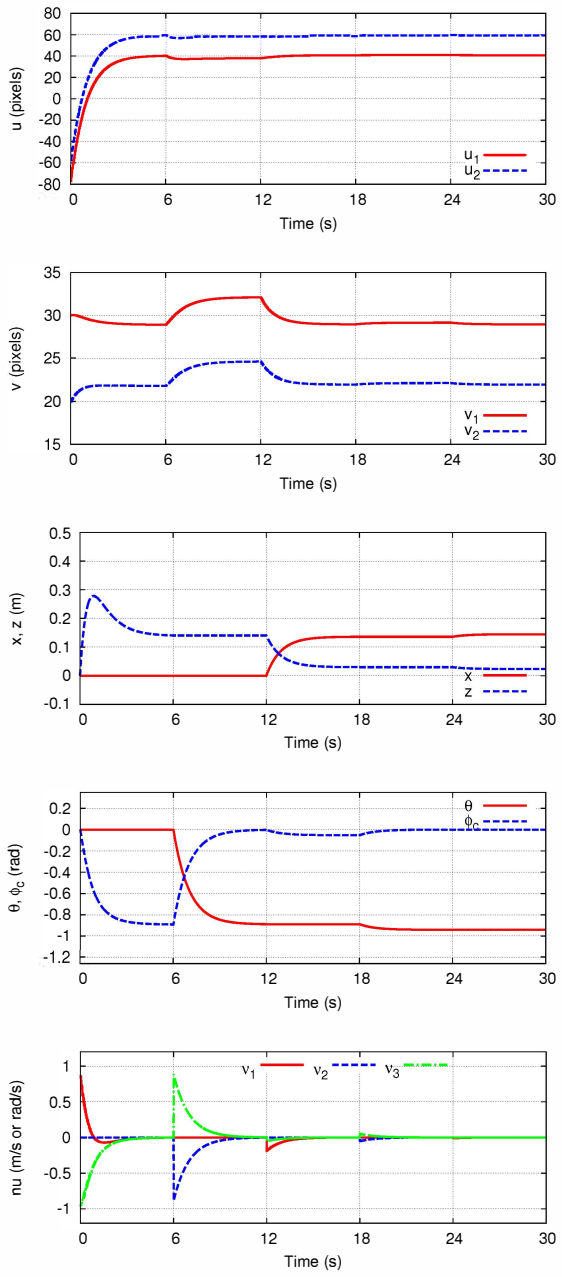


Fig. 3: Simulation results in the case of ${}^bx_c = 0.2\,\mathrm{m}$ and ${}^bz_c = 0$.

Figs. 3 and 4 show simulation results for 30 s with the following initial states and design parameters:

$$\begin{aligned} \boldsymbol{f}_1(0) &= [\,-60\,\text{pixels}, 30\,\text{pixels}\,]^\top, \\ \boldsymbol{f}_2(0) &= [\,-40\,\text{pixels}, 20\,\text{pixels}\,]^\top, \\ (x, z, \theta, \phi_c)(0) &= (\,0\,\text{m},\ 0\,\text{m},\ 0\,\text{rad},\ 0\,\text{rad}\,), \end{aligned}$$

and $\boldsymbol{K}_{\mathrm{img}} = \boldsymbol{I}_4$, $K_o = K_c = 1$. The case of ${}^bx_c = 0.2\,\mathrm{m}$ and ${}^bz_c = 0$ are shown in Fig. 3. Also, the case of ${}^bx_c = 0.2\,\mathrm{m}$ and ${}^bz_c = -0.25\,\mathrm{m}$ are shown in Fig. 4. Here any terminated conditions of each mode were not introduced, but the first mode started and the other modes changed every 6 s. The controller is in Mode I at the beginning of each simulation,

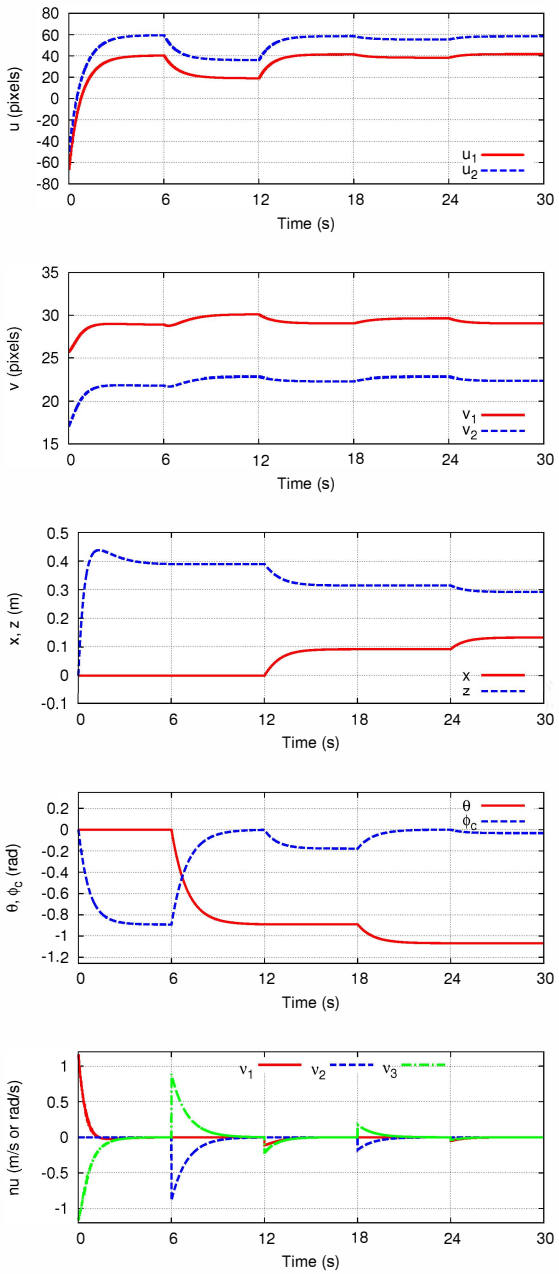


Fig. 4: Simulation results in the case of ${}^b x_c = 0.2\,\mathrm{m}$ and ${}^b z_c = -0.25\,\mathrm{m}$.

then switched to the other mode every 6 s.

From Figs. 3 and 4, it is easy to see the following things:

- In Mode I, $\boldsymbol{f}_i$ converged to $\boldsymbol{f}_i^d$ $(i = 1, 2)$. Although a steady-state error between v_i and v_i^d arose, it was kept within 3 pixels.
- In Mode II, the platform orientation was regulated so as to converge the pan camera angle into zero. Consequently, the camera was headed toward the forwarding direction of the platform.
- Repeating switching of the two modes, each feature point was kept around the desired one while the camera turned toward the forwarding direction of the platform.
- The proposed control strategy is valid without depending on the value of ${}^b x_c$.

The effectiveness of the proposed control strategy was demonstrated.

V. Conclusions

We have proposed a vision-based switching control strategy for a NHWMR+PC. In consideration of the structure of the robot Jacobian matrix, an IBVS controller using the platform's linear velocity and the camera's pan angular velocity was designed as the first control mode. A proportional controller that regulates the platform orientation and the pan camera angle was chosen as the second control mode. The validity of this strategy which switches two control modes was demonstrated through a simulation.

Appendix
Derivation of Robot Jacobian Matrix

The robot Jacobian matrix ${}^c\boldsymbol{J}(\phi_c)$ in (9) is derived from the geometric relation between the frames Σ_b and Σ_c. For the sake of simplicity, considering ${}^s\boldsymbol{w}_c = {}^c\boldsymbol{J}(\phi_c)\boldsymbol{\nu}$ as ${}^s\boldsymbol{w}_c = {}^c\boldsymbol{J}_1\nu_1 + {}^c\boldsymbol{J}_2\nu_2 + {}^c\boldsymbol{J}_3\nu_3$, we here derive each vector field ${}^c\boldsymbol{J}_i$, $i = 1, 2, 3$.

Firstly, we consider the geometric relation between ν_1 and ${}^s\boldsymbol{w}_c$ for $\nu_2 = \nu_3 = 0$ as shown in Fig. 5 (a). The relative angle ϕ_c of Σ_c with respect to Σ_b decomposes the translational velocity ν_1 into $\nu_1 \sin\phi_c$ and $\nu_1 \cos\phi_c$. These decomposed velocities are corresponded to ${}^s w_{c1}$ and ${}^s w_{c2}$ as follows:

$$\begin{bmatrix} {}^s w_{c1} \\ {}^s w_{c2} \\ {}^s w_{c3} \end{bmatrix} = \begin{bmatrix} \cos\phi_c & 0 & -\sin\phi_c \\ 0 & 1 & 0 \\ \sin\phi_c & 0 & \cos\phi_c \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \nu_1 \end{bmatrix} = \begin{bmatrix} -\nu_1 \sin\phi_c \\ 0 \\ \nu_1 \cos\phi_c \end{bmatrix}. \tag{18}$$

From (18) and ${}^s w_{ci} = 0,\ i = 4, 5, 6$, the velocity twist of the camera ${}^s\boldsymbol{w}_c$ is associated with ν_1 as follows:

$${}^s\boldsymbol{w}_c = \underbrace{\begin{bmatrix} -\sin\phi_c, & 0, & \cos\phi_c, & 0, & 0, & 0 \end{bmatrix}^\top}_{{}^c\boldsymbol{J}_1(\phi_c)} \nu_1. \tag{19}$$

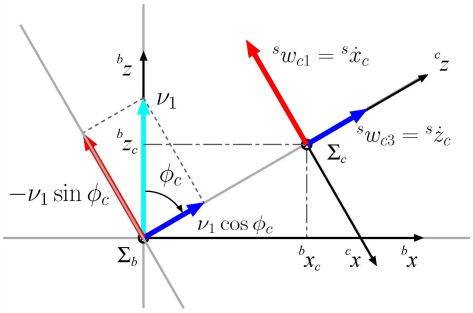
Secondly, we consider the geometric relation between ν_2 and ${}^s\boldsymbol{w}_c$ for $\nu_1 = \nu_3 = 0$ as shown in Fig. 5 (b). The translational velocity at the origin of Σ_c is given by the following vector cross product:

$$\begin{bmatrix} {}^b z_c\,\nu_2 \\ 0 \\ -{}^b x_c\,\nu_2 \end{bmatrix} = \begin{bmatrix} 0 \\ \nu_2 \\ 0 \end{bmatrix} \times \begin{bmatrix} {}^b x_c \\ 0 \\ {}^b z_c \end{bmatrix}.$$

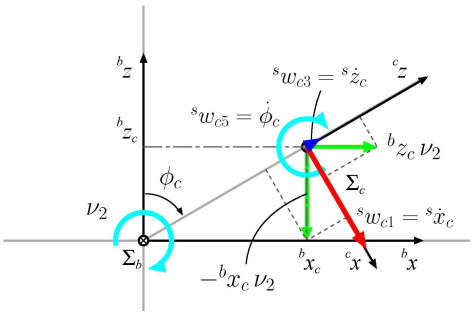
This velocity is transformed by the angle ϕ_c, which coincides with $[\,{}^s w_{c1},\ {}^s w_{c2},\ {}^s w_{c3}\,]^\top$ as follows:

$$\begin{bmatrix} {}^s w_{c1} \\ {}^s w_{c2} \\ {}^s w_{c3} \end{bmatrix} = \begin{bmatrix} \cos\phi_c & 0 & -\sin\phi_c \\ 0 & 1 & 0 \\ \sin\phi_c & 0 & \cos\phi_c \end{bmatrix} \begin{bmatrix} {}^b z_c\,\nu_2 \\ 0 \\ -{}^b x_c\,\nu_2 \end{bmatrix}. \tag{20}$$

Also, ν_2 is directly corresponded to ${}^s w_{c5}$. From (20), ${}^s w_{c5} = \nu_2$ and ${}^s w_{c4} = {}^s w_{c6} = 0$, the velocity twist of the



(a) In the case of $\nu_2 = \nu_3 = 0$.



(b) In the case of $\nu_1 = \nu_3 = 0$.

Fig. 5: Geometric relation between ${}^s\boldsymbol{w}_c$ and $\boldsymbol{\nu}$.

camera ${}^s\boldsymbol{w}_c$ is associated with ν_2 as follows:

$$ {}^s\boldsymbol{w}_c = \underbrace{\begin{bmatrix} {}^b x_c \sin\phi_c + {}^b z_c \cos\phi_c \\ 0 \\ -{}^b x_c \cos\phi_c + {}^b z_c \sin\phi_c \\ 0 \\ 1 \\ 0 \end{bmatrix}}_{{}^c\boldsymbol{J}_2(\phi_c)} \nu_2 \tag{21} $$

Lastly, we consider the geometric relation between ν_3 and ${}^s\boldsymbol{w}_c$ for $\nu_1 = \nu_2 = 0$. Then the following equation is easily derived:

$$ {}^s\boldsymbol{w}_c = \underbrace{[0,\ 0,\ 0,\ 0,\ 1,\ 0]^\top}_{{}^c\boldsymbol{J}_3} \nu_3. \tag{22} $$

The derivation of the robot Jacobian matrix ${}^c\boldsymbol{J}$ in (9) is then concluded.

REFERENCES

[1] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.

[2] K. Hashimoto, "A review on vision-based control of robot manipulators," *Advanced Robotics*, vol. 17, no. 10, pp. 969–991, 2003.

[3] F. Chaumette and S. Hutchinson, "Visual servoing and visual tracking," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib (Eds), ch. 24, pp. 563–583, Springer, 2008.

[4] N. Oda, M. Ito, and M. Shibata, "Vision-based motion control for robotic systems," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 4, no. 2, pp. 176–183, 2009.

[5] E. Malis and F. Chaumette and S. Boudet, "2-1/2-D visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 238–250, 1999.

[6] P. Corke and S. Hutchinson, "A new partitioned approach to image-based visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, pp. 507–515, 2001.

[7] K. Deguchi, "Optimal motion control for image-based visual servoing by decoupling translation and rotation," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'98)*, pp. 705–711, Victoria, Canada, 1998.

[8] E. M. P. Low, I. R. Manchester, and A. V. Savkin, "A biologically inspired method for vision-based docking of wheeled mobile robots," *Robotics and Autonomous Systems*, vol. 55, pp. 769–784, 2007.

[9] S. Komada, K. Kinoshita, T. Hirukawa, and J. Hirai, "Image feature based navigation of nonholonomic mobile robots with active camera," in *Proceedings of the 17th IFAC World Congress (IFAC'08)*, pp. 14714–14719, 2008.

[10] Y. Fang, X. Liu, and X. Zhang, "Adaptive active visual servoing of nonholonomic mobile robots," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 1, pp. 486–497, 2012.