Journal of Computer Science and Cybernetics, V.33, N.4 (2017), 339–355
DOI 10.15625/1813-9663/33/4/9736

MODELING THE DIFFERENTIAL MOTION OF A MOBILE MANIPULATOR AND DESIGNING A NEW VISUAL SERVOING FOR TRACKING A FLYING TARGET

NGUYEN TIEN KIEM[1], HOANG THI THUONG[2], NGUYEN VAN TINH[3]

[1] *Faculty of Electronics Engineering Technology, Ha Noi University of Industry*
[2] *University of Information and Communication Technology, Thai Nguyen University*
[3] *Institute of Information Technology, Viet Nam Academy of Science and Technology*
[1] *kiemnt@haui.edu.vn*

Abstract. This article describes a process to model the differential motion of a mobile manipulator which is a two-degree-of-freedom robotic arm (pan-tilt) mounted on a wheeled mobile robot (WMR). Next, a new visual servoing is designed for this pan-tilt arm with the purpose of making the image feature of a target converge to the center of the image plane of a camera attached to the arm's end-effector. Furthermore, this new visual servoing is able to deal with the uncertainties due to the unknown motions of both the flying target considered as a material point and the WMR moving on the floor. The global uniform asymptotic stability of this visual servoing is guaranteed by Lyapunov criteria. Simulation results implemented by Matlab/Simulink software have confirmed the both validity and performance of the entire control system.

Keywords. Global uniform asymptotic stability, image feature, mobile manipulator, track a flying target, unknown trajectory.

1. INTRODUCTION

In recent years, mobile manipulators are increasingly applied in many various areas which demand high performance all over the world such as assembly, mining, construction, part transfer in complex works composed of a variety of obstacles (may be known or unknown) and so on.

When it comes to the motion problem of mobile manipulators, many researchers have been developing control strategies for the mobile manipulators, or, more precisely, the goal of solving a motion problem is to control a mobile manipulator from an initial configuration to another configuration where the end-effector is a desired location. To be specific, the methods in [1-3] have been some remarkable strategies to solve these motion problems. In addition, the work in [4] has expressed an adaptive tracking control method for a welding mobile manipulator with a kinematic model in the presence of some unknown dimensional parameters. Based on Lyapunov stability theory, the author in [5] has addressed a position control problem with kinematic and dynamic uncertainties and unknown obstacles. Furthermore, a torque compensation controller has been proposed in [6] for motion controlling of a mobile arm.

Recently, many works with the purpose of integrating visual servoing into mobile robots have been proposed for grasping tasks [7–8] and for addressing visual based tracking problem

[9–10], which leads to vision-based mobile autonomous manipulation systems. Moreover, experts have proposed a path-planning algorithm in addtition to a reactive visual servoing strategy. The planning stage considers various critical constraints or system uncertainties, achieving a more robust visual servoing system.

As regards to vision, whenever an articulated arm manipulates in dynamic and unstructured work spaces, it is necessary to receive sensory information from feedback signals like visual information in a closed loop control system [11]. Vision is a helpful sensor for such an articulated arm as it copies biomimetic eyes to get information in the absence of any contact with the object.

For robotic manipulators, *visual servoing* is the name of control methods composed of a combination of robotic kinematics, dynamics, and computer vision to efficiently drive a manipulator's motion. These methods are categorized as two groups [12], namely, *position-based visual servoing* (PBVS) and *image-based visual servoing* (IBVS).

Image features, in PBVS, are dealt with so as to estimate the relative three-dimensional (3D) position between the camera and the target, followed by a strategy to control the motion of a robotic arm with a camera, where the 3D position is used as an error signal [13]. In other words, based on image data, the references have been designed and expressed in 3D Cartesian space. The control objective here is to drive the camera (or the hand) from an arbitrarily initial to a desired relative position.

Alternatively, in IBVS, errors are calculated directly in terms of image features whose differential motions in the image plane are related to the differential motion of the mobile arm through Jacobian matrice [9-16]. It should be noted that as opposed to PBVS, IBVS has some advantages as follows: 1) the 3D coordinate of a target is not essential; 2) IBVS has more robustness than PBVS in performance with respect to disturbance, for instance, calibration errors; 3) IBVS is more convenient and easier than PBVS to track a moving target so that this target is always in the field of view of the camera.

The main contribution of this paper is that we show a completely new method to compute the derivative of the image feature of a flying target by modeling the differential motion of a camera mounted on a mobile arm. Afterwards, a new visual servoing law is proposed in order to control the angular velocities of the pan-tilt joints with the purpose of making the image feature of the flying target converge asymptotically to the center of the image plane of the camera even though the target's motion trajectory is unknown. Furthermore, apart from tracking the flying target, this visual servoing controller has to also compensate the motion of the WMR which is also moving on the floor with another unknown trajectory.

In comparison with other methods in [14-21], the advantages of our visual servoing include two strong points as follows:

- Firstly, this method does not use the pseudo-inverse of the image interaction matrix (image Jacobi matrix) to control the angular velocities of the pan-tilt's joints for tracking a flying target. Instead of using the pseudo-inverse, it uses the inverse of an invertible 2×2 matrix which is derived from both the image interaction matrix of the camera and the robotics Jacobian matrix of the mobile manipulator. Therefore, the robustness in performance is enhanced.

- Secondly, instead of separately estimating the variations of the image errors due to both the target's unknown motion and the depth of the target, our method estimates an expression consisting of both of them. Therefore, this makes the expression of this new visual servoing easier than that of other ones. Consequently, the burden of computing the control law is also reduced.

The paper is organized as follows. Section 2 describes how to model differential motion of a camera attached to the end-effector of a pan-tilt platform by using Paul's algorithm [22]. Section 3 represents a process by which a new visual servoing for tracking a flying target is designed. Simulation results and discussions are expressed in Section 4. Finally, our conclusion is shown in Section 5.

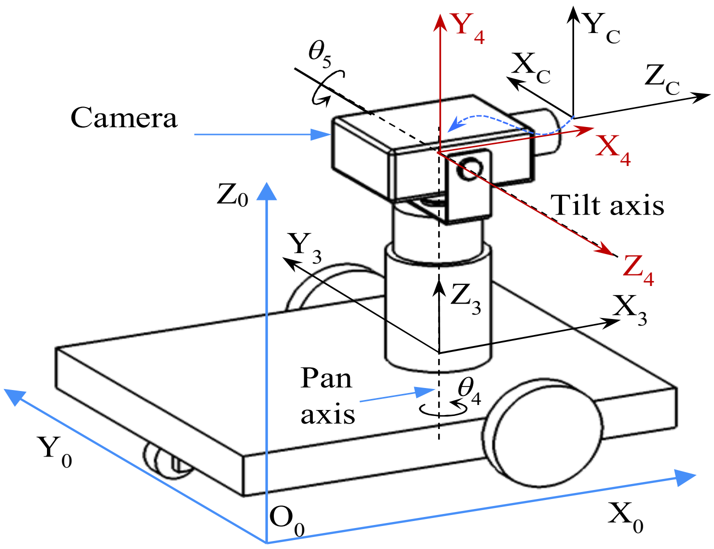

Figure 1. A two-degree-of-freedom manipulator (pan-tilt) with a camera on a wheeled mobile robot

2. MODELLING THE DIFFERENTIAL MOTION OF A CAMERA ON A MOBILE MANIPULATOR

2.1. Describing coordinate systems

To begin with, let us consider a mobile manipulator with a camera as Figure 1. We define a coordinate system $O_2X_2Y_2Z_2$ as Figure 2. Particularly, its origin O_2 coincides with point M, and its axes are always parallel to those of the base frame $O_0X_0Y_0Z_0$.

$O_3X_3Y_3Z_3$ is attached to the platform of the WMR as Figures 1, 2, and 3. $O_4X_4Y_4Z_4$ is attached to the link pan as Figure 1 and Figure 3. $O_CX_CY_CZ_C$ is attached to the platform of the camera (see Figures 1, 3 and 4). It should be noted here that O_4 is at the intersection of the pan axis and the tilt axis.

Finally, the homogeneous matrix expressing the position and direction of $O_CX_CY_CZ_C$ in $O_0X_0Y_0Z_0$ is shown in the following formula

$$\mathbf{T}_C^0 = \begin{bmatrix} -s_{34} & -c_{34}s_5 & c_{34}c_5 & x_M + x_c \\ c_{34} & -s_{34}s_5 & s_{34}c_5 & y_M + y_c \\ 0 & c_5 & s_5 & h_T + z_c \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{1}$$

where $s_i = \sin\theta_i$, $c_i = \cos\theta_i$, $s_{ij} = \sin(\theta_i + \theta_j)$, $c_{ij} = \cos(\theta_i + \theta_j)$, θ_3 is the direction of the mobile platform, x_M, y_M are Cartesian position coordinates of point M in the base frame (Figure 2), θ_4 is the angular coordinate of the pan joint, θ_5 is the angular coordinate of the tilt joint, h_T is the height of the tilt axis (see Figures 3 and 4), $(x_c, y_c, z_c)^T$ is the position coordinate vector of O_c in $O_4X_4Y_4Z_4$.

For convenience, we define extra variables as follows

$$\begin{array}{llll} x_x = -s_{34}, & y_x = -c_{34}s_5, & z_x = c_{34}c_5, & p_x = x_M + x_c, \\ x_y = c_{34}, & y_y = -s_{34}s_5, & z_y = s_{34}c_5, & p_y = y_M + y_c, \\ x_z = 0, & y_z = c_5, & z_z = s_5, & p_z = h_T + z_c. \end{array}$$

Therefore, (1) can be expressed as follows

$$\mathbf{T}_C^0 = \begin{bmatrix} x_x & y_x & z_x & p_x \\ x_y & y_y & z_y & p_y \\ x_z & y_z & z_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

2.2. Differential motion

As $O_CX_CY_CZ_C$ is attached to the body of the camera, in order to model the differential motion of the camera, we only model that of $O_CX_CY_CZ_C$ (see Figure 5).

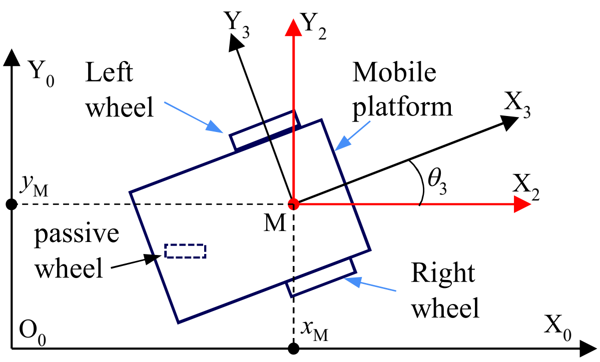

Figure 2. The mobile platform and two coordinate systems $O_2X_2Y_2Z_2$ and $O_3X_3Y_3Z_3$ in base frame

On one hand, if $O_CX_CY_CZ_C$ experiences *differential translations* $d^0\mathbf{trans}$ along the axes of the base frame O_0X_0, O_0Y_0, O_0Z_0 and rotates *differential rotations* $d^0\mathbf{rot}$ about the axes of the base frame, then its new posture (consisting of both location and direction) with respect to the base frame will be illustrated by premultiplying $\mathbf{T}_C^0$ by the differential translations and rotations as follows

$$new\_posture = d^0\mathbf{trans}.d^0\mathbf{rot}.\mathbf{T}_C^0 = d\mathbf{T}_C^0 + \mathbf{T}_C^0.$$

Thus, *differential change* $d\mathbf{T}_C^0$ is computed by

$$d\mathbf{T}_C^0 = \left(d^0\mathbf{trans}.d^0\mathbf{rot} - \mathbf{I}\right)\mathbf{T}_C^0 = \mathbf{\Xi}^0.\mathbf{T}_C^0, \tag{2}$$

where

$$\mathbf{\Xi}^0 = d^0\mathbf{trans}.d^0\mathbf{rot} - \mathbf{I} = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -\delta_z & \delta_y & 0 \\ \delta_z & 1 & -\delta_x & 0 \\ -\delta_y & \delta_x & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - \mathbf{I} \tag{3}$$
$$= \begin{bmatrix} 0 & -\delta_z & \delta_y & d_x \\ \delta_z & 0 & -\delta_x & d_y \\ -\delta_y & \delta_x & 0 & d_z \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

where d_x, d_y, and d_z are very small distances in $d^0\mathbf{trans}$ of $\mathrm{O}_C\mathrm{X}_C\mathrm{Y}_C\mathrm{Z}_C$ along axes $\mathrm{O}_0\mathrm{X}_0$, $\mathrm{O}_0\mathrm{Y}_0$, $\mathrm{O}_0\mathrm{Z}_0$ respectively. In addition, δ_x, δ_y and δ_z are tiny angles in $d^0\mathbf{rot}$ of $\mathrm{O}_C\mathrm{X}_C\mathrm{Y}_C\mathrm{Z}_C$ about axes $\mathrm{O}_0\mathrm{X}_0$, $\mathrm{O}_0\mathrm{Y}_0$, $\mathrm{O}_0\mathrm{Z}_0$.

On the other hand, if the camera witnesses differential translations $d^C\mathbf{trans}$ along $\mathrm{O}_C\mathrm{X}_C$, $\mathrm{O}_C\mathrm{Y}_C$, $\mathrm{O}_C\mathrm{Z}_C$ with very small distances d_x^C, d_y^C, d_z^C respectively and differential rotations $d^C\mathbf{rot}$ about $\mathrm{O}_C\mathrm{X}_C$, $\mathrm{O}_C\mathrm{Y}_C$, $\mathrm{O}_C\mathrm{Z}_C$ with tiny angles δ_x^C, δ_y^C and δ_z^C (see Figure 5) respectively, then its new status with respect to $\mathrm{O}_C\mathrm{X}_C\mathrm{Y}_C\mathrm{Z}_C$ will be described by postmultiplying $\mathbf{T}_0^C$ with $d^C\mathbf{trans}$ and $d^C\mathbf{rot}$

$$\mathrm{new\_posture} = \mathbf{T}_C^0 + d\mathbf{T}_C^0 = \mathbf{T}_C^0.d^C\mathbf{trans}.d^C\mathbf{rot}. \tag{4}$$

We can rewrite (4) as follows

$$d\mathbf{T}_C^0 = \mathbf{T}_C^0.\left(d^C\mathbf{trans}.d^C\mathbf{rot} - \mathbf{I}\right) = \mathbf{T}_C^0.\mathbf{\Xi}^C, \tag{5}$$

where

$$\mathbf{\Xi}^C = d^C\mathbf{trans}.d^C\mathbf{rot} - \mathbf{I} = \begin{bmatrix} 0 & -\delta_z^C & \delta_y^C & d_x^C \\ \delta_z^C & 0 & -\delta_x^C & d_y^C \\ -\delta_y^C & \delta_x^C & 0 & d_z^C \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{6}$$

Combining (2) and (5) results in

$$\mathbf{\Xi}^C = \left(\mathbf{T}_C^0\right)^{-1}.\mathbf{\Xi}^0.\mathbf{T}_C^0. \tag{7}$$

For convenience, from (1) and (3), we define new vectors as follows

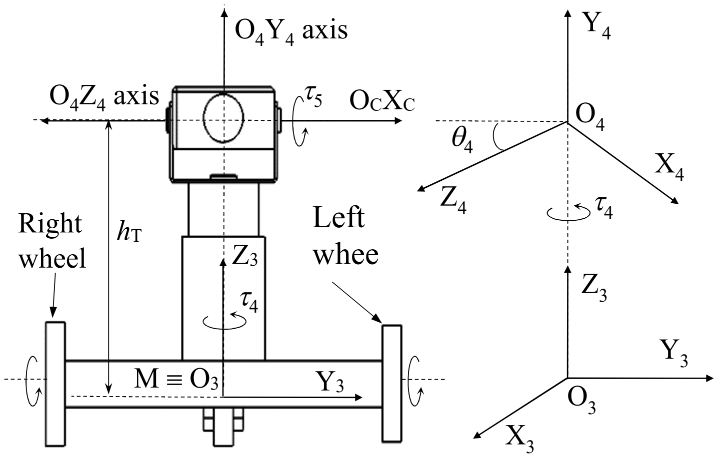

Figure 3. The front side of the system, and the position and direction of $O_4X_4Y_4Z_4$ in $O_3X_3Y_3Z_3$

$$\mathbf{x} = \begin{bmatrix} x_x & x_y & x_z \end{bmatrix}^T, \quad \mathbf{y} = \begin{bmatrix} y_x & y_y & y_z \end{bmatrix}^T, \quad \mathbf{z} = \begin{bmatrix} z_x & z_y & z_z \end{bmatrix}^T,$$

$$\mathbf{p} = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix}^T, \quad \boldsymbol{\delta} = \begin{bmatrix} \delta_x & \delta_y & \delta_z \end{bmatrix}^T, \quad \mathbf{d} = \begin{bmatrix} d_x & d_y & d_z \end{bmatrix}^T.$$

That is to say, (7) can be rewritten as follows

$$\Xi^C = \begin{bmatrix} 0 & -\boldsymbol{\delta}^T.\mathbf{z} & \boldsymbol{\delta}^T.\mathbf{y} & \mathbf{x}^T\left[(\boldsymbol{\delta}\times\mathbf{p})+\mathbf{d}\right] \\ \boldsymbol{\delta}^T.\mathbf{z} & 0 & -\boldsymbol{\delta}^T.\mathbf{x} & \mathbf{y}^T\left[(\boldsymbol{\delta}\times\mathbf{p})+\mathbf{d}\right] \\ -\boldsymbol{\delta}^T.\mathbf{y} & \boldsymbol{\delta}^T.\mathbf{x} & 0 & \mathbf{z}^T\left[(\boldsymbol{\delta}\times\mathbf{p})+\mathbf{d}\right] \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{8}$$

where $(\boldsymbol{\delta} \times \mathbf{p})$ is the cross product of these two vectors.

Comparing (6) and (8) yields

$$d_x^C = \mathbf{x}^T\left[(\boldsymbol{\delta}\times\mathbf{p})+\mathbf{d}\right], \tag{9}$$

$$d_y^C = \mathbf{y}^T\left[(\boldsymbol{\delta}\times\mathbf{p})+\mathbf{d}\right], \tag{10}$$

$$d_z^C = \mathbf{z}^T\left[(\boldsymbol{\delta}\times\mathbf{p})+\mathbf{d}\right], \tag{11}$$

$$\delta_x^C = \boldsymbol{\delta}^T.\mathbf{x}, \tag{12}$$

$$\delta_y^C = \boldsymbol{\delta}^T.\mathbf{y}, \tag{13}$$

$$\delta_z^C = \boldsymbol{\delta}^T.\mathbf{z}. \tag{14}$$

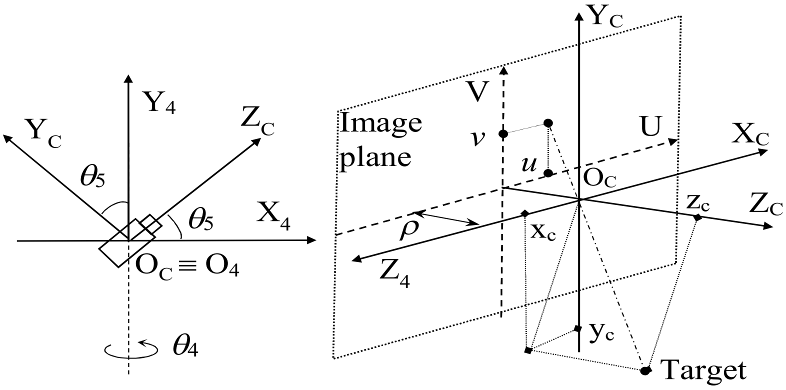

Figure 4. The position and direction of $O_CX_CY_CZ_C$ in $O_4X_4Y_4Z_4$, and the model pinhole of the camera

Let us define the differential motion vector of the camera with respect to the camera frame $O_CX_CY_CZ_C$ as follows

$$\mathbf{D} = \begin{bmatrix} d_x^C & d_y^C & d_z^C & \delta_x^C & \delta_y^C & \delta_z^C \end{bmatrix}^T. \tag{15}$$

Alternatively, we compute the robotic Jacobian matrix $\mathbf{J}$ so that it satisfies the following fomula

$$\mathbf{D} = \mathbf{J} \begin{bmatrix} dx_M & dy_M & d\theta_3 & d\theta_4 & d\theta_5 \end{bmatrix}^T, \tag{16}$$

where

$$\begin{aligned} \mathbf{J} &= \begin{bmatrix} \mathbf{J}_1 & \mathbf{J}_2 & \mathbf{J}_3 & \mathbf{J}_4 & \mathbf{J}_5 \end{bmatrix} \\ &= \begin{bmatrix} \dfrac{\partial \mathbf{D}}{\partial x_M} & \dfrac{\partial \mathbf{D}}{\partial y_M} & \dfrac{\partial \mathbf{D}}{\partial \theta_3} & \dfrac{\partial \mathbf{D}}{\partial \theta_4} & \dfrac{\partial \mathbf{D}}{\partial \theta_5} \end{bmatrix}. \end{aligned} \tag{17}$$

For the differential translation along the O_0X_0 (see Figure 5), we have $\mathbf{d} = \begin{bmatrix} dx_M & 0 & 0 \end{bmatrix}^T$ and $\boldsymbol{\delta} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$. Therefore, according to (1), (9)-(14) and (17), the following formula is achieved

$$\mathbf{J}_1 = \begin{bmatrix} -s_{34} & -c_{34}s_5 & c_{34}c_5 & 0 & 0 & 0 \end{bmatrix}^T. \tag{18}$$

Similarly, for the differential translation along the O_0Y_0 (see Figure 5), we have $\mathbf{d} = \begin{bmatrix} 0 & dy_M & 0 \end{bmatrix}^T$ and $\boldsymbol{\delta} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$. It also results in

$$\mathbf{J}_2 = \begin{bmatrix} c_{34} & -s_{34}s_5 & s_{34}c_5 & 0 & 0 & 0 \end{bmatrix}^T. \tag{19}$$

Now, if we consider differential rotations about the corresponding axes O_2Z_2, O_3Z_3, O_4Z_4 respectively, then the role of $\mathbf{T}_C^0$ in both (1) and (7) will be respectively replaced by the corresponding matrices as follows

$$\mathbf{T}_C^2 = \begin{bmatrix} -s_{34} & -c_{34}s_5 & c_{34}c_5 & x_c \\ c_{34} & -s_{34}s_5 & s_{34}c_5 & y_c \\ 0 & c_5 & s_5 & h_T + z_c \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{20}$$

which represents the position and direction of $O_CX_CY_CZ_C$ in $O_2X_2Y_2Z_2$,

$$\mathbf{T}_C^3 = \begin{bmatrix} -s_4 & -c_4s_5 & c_4c_5 & x_c \\ c_4 & -s_4s_5 & s_4c_5 & y_c \\ 0 & c_5 & s_5 & h_T + z_c \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{21}$$

which represents the position and direction of $O_CX_CY_CZ_C$ in $O_3X_3Y_3Z_3$, and

$$\mathbf{T}_C^4 = \begin{bmatrix} 0 & -\sin\theta_5 & \cos\theta_5 & x_c \\ 0 & \cos\theta_5 & \sin\theta_5 & y_c \\ -1 & 0 & 0 & z_c \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{22}$$

which illustrates the position and direction of $O_CX_CY_CZ_C$ in $O_4X_4Y_4Z_4$ (see Figure 1). In these three cases, we have $\boldsymbol{\delta} = \begin{bmatrix} 0 & 0 & d\theta_i \end{bmatrix}^T$, $i = 3, 4, 5$, and $\mathbf{d} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$. Combining (9)-(14), (17), and (20)-(22), results in that the Jacobian vectors in (17) can be written as follows

$$\mathbf{J}_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & c_5 & s_5 \end{bmatrix}^T, \tag{23}$$

$$\mathbf{J}_4 = \begin{bmatrix} 0 & 0 & 0 & 0 & c_5 & s_5 \end{bmatrix}^T, \tag{24}$$

$$\mathbf{J}_5 = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix}^T. \tag{25}$$

Combining (18)-(19) and (23)-(25) allows one to show the robotics Jacobian matrix as follows

$$\mathbf{J} = \begin{bmatrix} -s_{34} & c_{34} & 0 & 0 & 0 \\ -c_{34}s_5 & -s_{34}s_5 & 0 & 0 & 0 \\ c_{34}c_5 & s_{34}c_5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & c_5 & c_5 & 0 \\ 0 & 0 & s_5 & s_5 & 0 \end{bmatrix}. \tag{26}$$

2.3. Calculating the derivative of the image feature

Figure 4 shows the pinhole model of the camera, where u, v are the image coordinates of the target in the image plane. The image feature vector of the target is computed as follows

$$\boldsymbol{\xi} = \begin{bmatrix} u \\ v \end{bmatrix} = -\frac{\rho}{z_{Tc}} \begin{bmatrix} x_{Tc} \\ y_{Tc} \end{bmatrix}, \tag{27}$$

where ρ is the focus length of the camera, $\mathbf{r}_{Tc} = \begin{bmatrix} x_{Tc} & y_{Tc} & z_{Tc} \end{bmatrix}^T$ is the coordinate vector of the target in the camera frame ($O_CX_CY_CZ_C$).

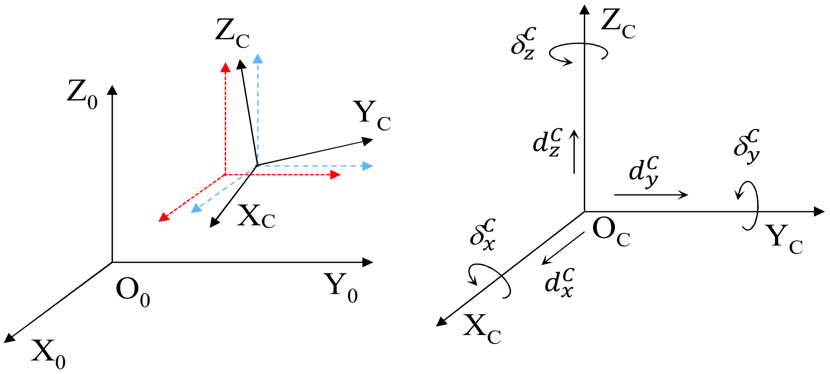

Figure 5. Expressing the differential motion of the camera via that of $O_CX_CY_CZ_C$

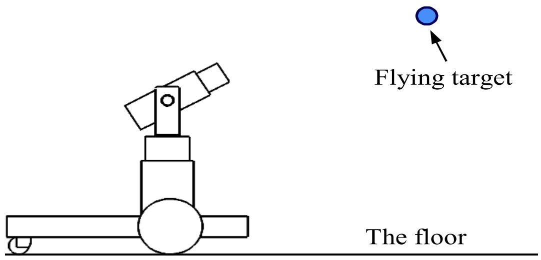

Figure 6. Tracking a flying target

Let $\mathbf{d}^C = \begin{bmatrix} d_x^C & d_y^C & d_z^C \end{bmatrix}^T$ be the differential translation vector, $\boldsymbol{\delta}^C = \begin{bmatrix} \delta_x^C & \delta_y^C & \delta_z^C \end{bmatrix}^T$ be the differential rotation vector of the camera with respect to $O_CX_CY_CZ_C$. The differential motion of the target in $O_CX_CY_CZ_C$ is computed by an equation as follows [11]

$$d\mathbf{r}_{Tc} = \begin{bmatrix} dx_c & dy_c & dz_c \end{bmatrix}^T = -\boldsymbol{\delta}^C \times \mathbf{r}_{Tc} - \mathbf{d}^C + \frac{\partial \mathbf{r}_{Tc}}{\partial t}dt, \tag{28}$$

where $\dfrac{\partial \mathbf{r}_{Tc}}{\partial t}dt$ expresses a component which only depends on the unknown motion of the target in 3D-space. In other words, it does not depend on the motion of the camera.

In particular, we can rewrite (28) as follows

$$dx_{Tc} = -z_{Tc}\left(\delta_y^C + \frac{v}{\rho}\delta_z^C\right) - d_x^C + \frac{\partial x_{Tc}}{\partial t}dt, \tag{29}$$

$$dy_{Tc} = z_{Tc}\left(\frac{u}{\rho}\delta_z^C + \delta_x^C\right) - d_y^C + \frac{\partial y_{Tc}}{\partial t}dt, \tag{30}$$

$$dz_{Tc} = \frac{z_{Tc}}{\rho}\left(v\delta_x^C - u\delta_y^C\right) - d_z^C + \frac{\partial z_{Tc}}{\partial t}dt. \tag{31}$$

According to (27), the differential expressions of the image coordinates are represented in the following forms

$$du = -\rho \frac{z_{Tc} dx_{Tc} - x_{Tc} dz_{Tc}}{z_{Tc}^2}, \tag{32}$$

$$dv = -\rho \frac{z_{Tc} dy_{Tc} - y_{Tc} dz_{Tc}}{z_{Tc}^2}. \tag{33}$$

Substituting (29), (30), and (31) into (32)-(33) leads to

$$d\boldsymbol{\xi} = \begin{bmatrix} du \\ dv \end{bmatrix} = \mathbf{J}_{im}.\mathbf{D} - \boldsymbol{\zeta} dt, \tag{34}$$

where

$$\mathbf{J}_{im} = \begin{bmatrix} \frac{\rho}{z_c} & 0 & \frac{u}{z_c} & -\frac{uv}{\rho} & \frac{u^2+\rho^2}{\rho} & v \\ 0 & \frac{\rho}{z_c} & \frac{v}{z_c} & -\frac{v^2+\rho^2}{\rho} & \frac{uv}{\rho} & -u \end{bmatrix}$$

is the image Jacobian matrix (interaction matrix) of the camera, and

$$\boldsymbol{\zeta} = \begin{bmatrix} \left(\frac{\rho}{z_c} \frac{\partial x_c}{\partial t} + \frac{u}{z_c} \frac{\partial z_c}{\partial t} \right) & \left(\frac{\rho}{z_c} \frac{\partial y_c}{\partial t} + \frac{v}{z_c} \frac{\partial z_c}{\partial t} \right) \end{bmatrix}^T.$$

Substituting (16) into (34) results in

$$d\boldsymbol{\xi} = \mathbf{J}_{im}.\mathbf{J}.d\boldsymbol{\theta} - \boldsymbol{\zeta} dt, \tag{35}$$

where $\boldsymbol{\theta} = \begin{bmatrix} x_M & y_M & \theta_3 & \theta_4 & \theta_5 \end{bmatrix}^T$. Now, dividing both the sides of (35) by differential of time, dt, forms the following derivative equation of the image feature

$$\dot{\boldsymbol{\xi}} = \mathbf{J}_{im}.\mathbf{J}.\dot{\boldsymbol{\theta}} - \boldsymbol{\zeta}. \tag{36}$$

3. DESIGNING CONTROL LAW

3.1. Problem statement and proposition

The requirement of the visual servoing for tracking a flying target is to control the angular velocities of the pan-tilt joints so that the image feature of the target (Figure 6) tends asymptotically to the center of the image plane (see Figure 4) even though the motion trajectories of both the WMR and the flying target are unknown and independent each other.

To solve this control problem, we propose a scheme of the overall system as Figure 7. This scheme consists of two closed-loops. The outer loop includes a kinematic controller shown in Subsection 3.2. The inner loop involves a dynamic controller represented in Subsection 3.3.

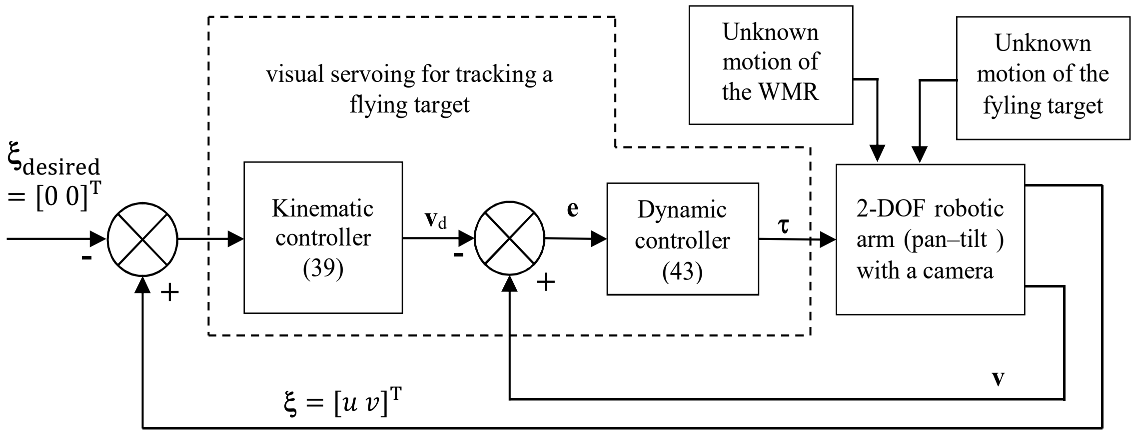

Figure 7. Scheme of the proposed visual servoing for tracking a flying target

3.2. Kinematic control law

We can rearrange (36) as follows

$$\dot{\boldsymbol{\xi}} = \mathbf{A}\begin{bmatrix}\dot{\theta}_4\\ \dot{\theta}_5\end{bmatrix} + \begin{bmatrix} v \\ -u\end{bmatrix} s_5\dot{\theta}_4 + \boldsymbol{\psi}, \tag{37}$$

where,

$$\mathbf{A} = \begin{bmatrix} \dfrac{(\rho^2+u^2)\,c_5}{\rho} & \dfrac{uv}{\rho} \\ \dfrac{uvc_5}{\rho} & \dfrac{\rho^2+v^2}{\rho}\end{bmatrix},$$

$$\boldsymbol{\psi} = \mathbf{H}\theta_3 + \mathbf{K}\begin{bmatrix}\dot{x}_M \\ \dot{y}_M\end{bmatrix} - \boldsymbol{\zeta}, \quad \mathbf{H} = \frac{1}{\rho}\begin{bmatrix}(\rho^2+u^2)\,c_5 + \rho v s_5 \\ uvc_5 - \rho u s_5\end{bmatrix},$$

$$\mathbf{K} = \frac{1}{z_c}\begin{bmatrix} (-\rho s_{34} + uc_{34}c_5) & (-\rho c_{34} + us_{34}c_5) \\ (-\rho c_{34}s_5 + vc_{34}c_5) & (-\rho s_{34}s_5 + vs_{34}c_5)\end{bmatrix},$$

$\boldsymbol{\psi}$ describes the variation of the image feature error $\boldsymbol{\xi}$ because of the unknown motion of the flying target.

In (37), depending on the unknown motion of both the WMR and the flying target, and above all, the depth, z_{Tc}, of target, $\boldsymbol{\psi}$ is unknown. However, when the sampling interval of signals is tiny enough for real-time property to be guaranteed, $\boldsymbol{\psi}$ may be estimated as follows [14]

$$\hat{\boldsymbol{\psi}} = \dot{\boldsymbol{\xi}}^{pre} - \mathbf{A}\begin{bmatrix}\dot{\theta}_4^{pre}\\ \dot{\theta}_5^{pre}\end{bmatrix} - \begin{bmatrix} v \\ -u\end{bmatrix} s_5\dot{\theta}_4^{pre}, \tag{38}$$

where $\hat{\boldsymbol{\psi}}$ is the estimated vector of $\boldsymbol{\psi}$. Furthermore, $\dot{\boldsymbol{\xi}}^{pre}$, $\dot{\theta}_4^{pre}$, and $\dot{\theta}_5^{pre}$ are the latest discrete data of $\dot{\boldsymbol{\xi}}$, $\dot{\theta}_4$, and $\dot{\theta}_5$, respectively.

Since the desired position of the image feature is the center of the image plane, the desired vector of $\boldsymbol{\xi}$ is $\boldsymbol{\xi}_d = [0,0]^T$. Hence, the image error is also $\boldsymbol{\xi}$.

Because of $\det(\mathbf{A}) = (\rho^2 + u^2 + v^2)\, c_5$, there is an undeniable fact that $\mathbf{A}$ is an invertible matrix if $|\theta_5| < \frac{\pi}{2}$. As a result, if $|\theta_5| < \frac{\pi}{2}$, then in order to remove the image error $\boldsymbol{\xi}$, we can choose the desired angular velocities for the pan-tilt joints as follows

$$\begin{bmatrix} \dot{\theta}_{4d} \\ \dot{\theta}_{5d} \end{bmatrix} = \mathbf{A}^{-1}\left(-\mathbf{N}\boldsymbol{\xi} - n\frac{\boldsymbol{\xi}}{\|\boldsymbol{\xi}\|} - \hat{\boldsymbol{\psi}}\right), \tag{39}$$

where $\mathbf{N}$ is a positive-definite diagonal constant matrix, n is a positive constant. Both $\mathbf{N}$ and n can be chosen arbitrarily.

Replacing $\begin{bmatrix} \dot{\theta}_4 & \dot{\theta}_5 \end{bmatrix}^T$ in (37) by $\begin{bmatrix} \dot{\theta}_{4d} & \dot{\theta}_{5d} \end{bmatrix}^T$ in (39), we get the following equation

$$\dot{\boldsymbol{\xi}} = -\mathbf{N}\boldsymbol{\xi} - n\frac{\boldsymbol{\xi}}{\|\boldsymbol{\xi}\|} + \begin{bmatrix} v \\ -u \end{bmatrix} s_5\dot{\theta}_{4d} + \tilde{\boldsymbol{\psi}}, \tag{40}$$

where $\tilde{\boldsymbol{\psi}} = \boldsymbol{\psi} - \hat{\boldsymbol{\psi}}$.

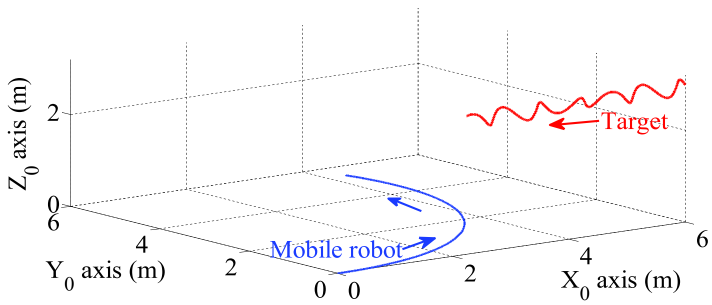

Figure 8. Trajectories of both the WMR (blue) and the flying target (red) in 3D space

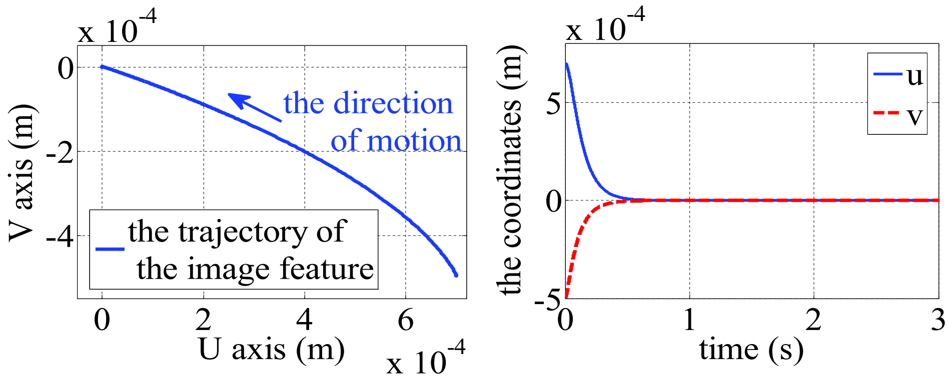

Figure 9. a) Trajectory of the image feature in the image plane
b) Evolution of its coordinates with time

3.3. Dynamic control law

The dynamic model of the platform of the pan-tilt is expressed as follows

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\,\dot{\mathbf{v}} + \mathbf{B}(\mathbf{q}, \mathbf{v})\,\mathbf{v} + \mathbf{g}(\mathbf{q}), \tag{41}$$

where $\mathbf{q} = \begin{bmatrix} \theta_4 & \theta_5 \end{bmatrix}^T$, $\mathbf{v} = \begin{bmatrix} \dot{\theta}_4 & \dot{\theta}_5 \end{bmatrix}^T$, $\boldsymbol{\tau} = [\tau_4, \tau_5]$, τ_4 is the torque at the pan joint, τ_5 is the torque at the tilt joint (see Fig. 3). All $\mathbf{M}(\mathbf{q})$, $\mathbf{B}(\mathbf{q},\mathbf{v})$, and $\mathbf{g}(\mathbf{q})$ are shown specifically in the appendix.

Remark 1. $\mathbf{M}(\mathbf{q})$ is always a symmetric and positive-definite matrix.

Remark 2. $\dot{\mathbf{M}}(\mathbf{q}) - 2\mathbf{B}(\mathbf{q},\mathbf{v})$ is a skew-symmetric matrix, that is,

$$\boldsymbol{\varphi}^T \left[\dot{\mathbf{M}}(\mathbf{q}) - 2\mathbf{B}(\mathbf{q},\mathbf{v})\right] \boldsymbol{\varphi} = 0, \ \forall \boldsymbol{\varphi} \in R^{2\times 1}. \tag{42}$$

To design the dynamic control law, the torque vector is selected as follows

$$\boldsymbol{\tau} = -\boldsymbol{\Gamma}\mathrm{e} + \mathbf{M}(\mathbf{q})\dot{\boldsymbol{v}}_d + \mathbf{B}(\mathbf{q},\mathbf{v})\mathbf{v}_d + \mathbf{g}(\mathbf{q}), \tag{43}$$

where $\mathbf{v}_d = \begin{bmatrix} \dot{\theta}_{4d} & \dot{\theta}_{5d} \end{bmatrix}^T$, $\mathbf{e} = \mathbf{v} - \mathbf{v}_d$, $\boldsymbol{\Gamma}$ is a constant, positive-definite, diagonal gain matrix and can be chosen arbitrarily.

Substituting (43) into (41), it leads to

$$\mathbf{M}(\mathbf{q})\dot{\boldsymbol{e}} = -\mathbf{B}(\mathbf{q},\mathbf{v})\mathbf{e} - \boldsymbol{\Gamma}\mathrm{e}. \tag{44}$$

3.4. Stability

A positively definite Lyapunov candidate function is chosen as follows

$$L = \frac{1}{2}\mathbf{e}^T\mathbf{M}(\mathbf{q})\mathbf{e} + \frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{\xi}. \tag{45}$$

Taking the first derivative of (45), we have

$$\dot{L} = \frac{1}{2}\mathbf{e}^T\dot{\mathbf{M}}(\mathbf{q})\mathbf{e} + \mathbf{e}^T\mathbf{M}(\mathbf{q})\dot{\mathbf{e}} + \boldsymbol{\xi}^T\dot{\boldsymbol{\xi}}. \tag{46}$$

Substituting both (40) and (44) into (46) and combining with (42) results in

$$\dot{L} = -\mathbf{e}^T\Gamma\mathbf{e} - \boldsymbol{\xi}^T\mathbf{N}\boldsymbol{\xi} - n\|\boldsymbol{\xi}\| + \boldsymbol{\xi}^T \begin{bmatrix} v \\ -u \end{bmatrix} s_5\dot{\theta}_{4d} + \boldsymbol{\xi}^T\tilde{\boldsymbol{\psi}}. \tag{47}$$

It is noticeable that $\boldsymbol{\xi}^T \begin{bmatrix} v \\ -u \end{bmatrix} = 0$, so (47) is reduced to

$$\dot{L} = -\mathbf{e}^T\Gamma\mathbf{e} - \boldsymbol{\xi}^T\mathbf{N}\boldsymbol{\xi} - n\|\boldsymbol{\xi}\| + \boldsymbol{\xi}^T\tilde{\boldsymbol{\psi}}. \tag{48}$$

It is assumed that $\tilde{\boldsymbol{\psi}}$ is bounded and there exists an upper bound as Ψ. It means that $\left\|\tilde{\boldsymbol{\psi}}\right\| \le \Psi$. Therefore, $\boldsymbol{\xi}^T\tilde{\boldsymbol{\psi}} \le \|\boldsymbol{\xi}\| . \left\|\tilde{\boldsymbol{\psi}}\right\| \le \Psi\|\boldsymbol{\xi}\|$.

Now, we can illustrate an inequality as follows

$$\dot{L} \le -\mathbf{e}^T\boldsymbol{\Gamma}\mathbf{e} - \boldsymbol{\xi}^T\mathbf{N}\boldsymbol{\xi} - n\|\boldsymbol{\xi}\| + \boldsymbol{\Psi}\|\boldsymbol{\xi}\|. \tag{49}$$

If $n = \Psi + \Omega$ is chosen where Ω is a positive constant, then (49) is rewritten as follows

$$\dot{L} \le -\mathbf{e}^T\boldsymbol{\Gamma}\mathbf{e} - \boldsymbol{\xi}^T\mathbf{N}\boldsymbol{\xi} - \Omega\|\boldsymbol{\xi}\|. \tag{50}$$

It is clear that $\dot{L} \leq 0$ for all $\mathbf{e}$, $\boldsymbol{\xi}$. Particularly, "=" occurs when and only when both $\mathbf{e}$ and $\boldsymbol{\xi}$ equal to zero vectors at the same time. It infers that $\dot{L}$ is a negatively definite function. Consequently, according to Lyapunov theory, $\dot{L} \to 0$ asymptotically. As a result, both $\mathbf{e}$ and $\boldsymbol{\xi}$ tend to zero asymptotically.

Clearly, the trend in which L converges to zero, $L \to 0$, does not depend on time history. It means that this trend has uniformity.

In summary, the stability of the entire control system is uniformly asymptotically stable.

4. SIMULATION RESULTS

Without loss of generality, suppose that the trajectories of the WMR and the target were shown in Table 1. These trajectories were illustrated in Figure 8.

In order to implement simulation by Matlab/Simulink software, the parameters of the pan-tilt's platform (see the APPENDIX) and the camera were assumed as Table 2. The parameters of the controller were chosen as follows $\mathbf{N} = \boldsymbol{\Gamma} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$, $n = 0.25$, the sampling interval $T = 0.001$ (s).

In the initial condition, it is assumed that the target had been in the field of view of the camera.

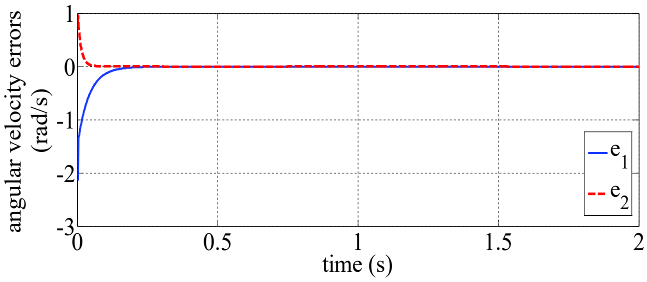

Figure 10. Evolution of $\mathbf{e} = \mathbf{v} - \mathbf{v}_d$ with time

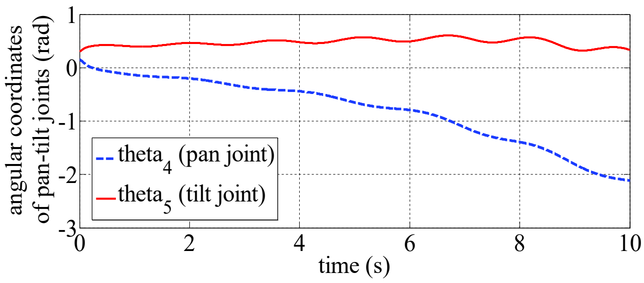

Figure 11. Coordinates of pan-tilt joints with time

Figure 9a expressed the trajectory of the image feature in the image plane. The evolution of the image coordinates with time was represented in Figure 9b. It is obvious that this image trajectory converged asymptotically to the center of the image plane. This implies that the

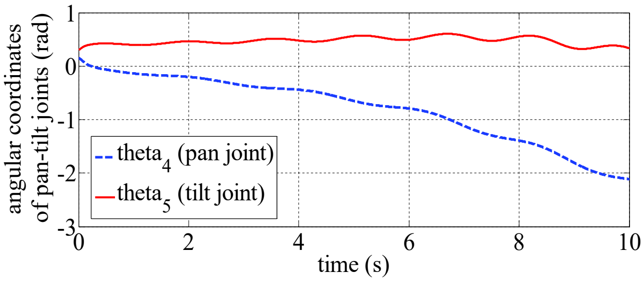

Figure 12. Evolution of the torques

control objective, or, more precisely, the requirement of the control problem in Subsection 3.1, has been satisfied.

It is interesting that combining Figure 9 and Figure 10, one can see that both $\mathbf{e}$ and $\boldsymbol{\xi}$ have converged asymptotically to the zero vectors. Therefore, what has been discussed after (50) is fully exact.

Figure 11 represented the evolution of the angular coordinates of the pan-tilt joints. It is noticeable that θ_5 always satisfies the condition $|\theta_5| < \dfrac{\pi}{2}$. This means that $\mathbf{A}$ in (39) is always an invertible matrix. For this reason, the kinematic control law in (39) is reasonable.

Next, Figure 12 described the torques at the pan-tilt joints. They are smooth and finite.

To conclude, the proposed image-based visual servoing is feasible and correct.

Table 1. The trajectories of both mobile robot and target

Coordinates of objects inthe base frame	Mobile robot (WMR)	Target
X_0(m)	$x_M = 5\sin(0.2t)$,	$x_T = 6$,
Y_0 (m)	$y_M = -5\cos(0.2t)$,	$y_T = -0.1 + 0.5t + 0.1\cos(3t)$,
Z_0(m)	$z_M = 0$,	$z_T = 3 - 0.2t + 0.15\sin(4t)$,
Direction (rad)	$\theta_3 = 0.2t$,	Undetermined

5. CONCLUSION

This article has shown a process for modelling the differential motion of a mobile manipulator by using Paul's algorithm. Subsequently, a fully novel visual servoing for tracking a flying target is designed with the purpose of making the target's image feature tend asymptotically to the center of the image plane when both the mobile robot and the target are moving with unknown trajectories. As opposed to other methods, the advantages of the visual servoing comprise two strong points. The first strong point is that this method has not used the pseudo-inverse of the interaction matrix. The second one is that it has also not estimated the depth of the target. Therefore, this visual servoing method gets the more robustness in performance than other ones. The uniform asymptotic stability of the whole system is ensured by Lyapunov criteria. Simulation results executed by Matlab/Simulink certify the correctness and performance of our proposed control method.

APPENDIX

The terms of the dynamic model (43)

$$\mathbf{M}(\mathbf{q}) = \begin{bmatrix} M_{11} & 0 \\ 0 & I_{Xc} \end{bmatrix}, \quad M_{11} = I_P + I_{Yc}\frac{1+\cos(2\theta_5)}{2} + I_{Zc}\frac{1-\cos(2\theta_5)}{2},$$

$$\mathbf{B}(\mathbf{q},\dot{\mathbf{q}}) = \begin{bmatrix} B_{11} & B_{12} \\ -B_{12} & 0 \end{bmatrix}, \quad B_{11} = \frac{1}{2}(I_{Zc} - I_{Yc})\sin(2\theta_5)\dot{\theta}_5,$$

$$B_{12} = \frac{1}{2}(I_{Zc} - I_{Yc})\sin(2\theta_5)\dot{\theta}_4.$$

The gravity vector $\mathbf{g}(\mathbf{q}) = \begin{bmatrix} 0 \\ 9.8 m_b \eta \cos\theta_5 \end{bmatrix}$.

I_P is the moment of inertia of the link pan's platform about its rotational axis.

I_{Xc}, I_{Yc}, and I_{Zc} respectively are the moments of inertia of the body including both the camera and the link tilt (see Figure 3 and Figure 4).

$\begin{pmatrix} 0 & 0 & -\eta \end{pmatrix}^T$, with $\eta > 0$, is the position of the center of mass of the body in $\mathrm{O}_C\mathrm{X}_C\mathrm{Y}_C\mathrm{Z}_C$.

m_b is the mass of this body.

Table 2. Parameters of the pan tilt platform and camera

$I_P = 0.025\text{kg.m}^2$	$I_{Xc} = 0.015\text{kg.m}^2$
$I_{Yc} = 0.005\text{kg.m}^2$	$I_{Zc} = 0.004\text{kg.m}^2$
$m_b = 0.5\text{kg}$	$\eta = 0.01\text{m}$
$\rho = 0.005\text{m}$	

REFERENCES

[1] M. Galicki, "Task space control of mobile manipulators," *Robotica*, vol. 29, pp. 221–232, 2011.

[2] M. Galicki, "Collision-free control of mobile manipulators in task space," *Mech. Syst. Signal Process*, vol. 25, no. 7, pp. 2766–2784, 2011.

[3] A. Mazur, "Trajectory tracking control in workspace-defined tasks for nonholonomic mobile manipulators," *Robotica*, vol. 28, pp. 57–68, 2010.

[4] N. T. Phuong, V. H. Duy, J. H. Jeong, H. K. Kim, and S. B. Kim. "Adaptive control for welding mobile manipulator with unknown dimensional parameters," *Proc. of the IEEE international Conf on Mechatronics*, pp. 1–6, 2007.

[5] B. W. Chi, and F. X. Ke, "Robust control of mobile manipulator service robot using torque compensation," *Proc. of the IEEE International Conf On Information Technology And Computer Science*, pp. 69–72, 2009.

[6] M. Galicki, "An adaptive non-linear constraint control of mobile manipulators," *Mechanism and Machine Theory*, vol. 88, pp. 63–85, 2015.

[7] A. Muis, Ohnishi, "Eye-to-hand approach on eye-in-hand configuration within real-time visual servoing", *IEEE/ASME Trans. Mechatronics*, vol. 10, pp. 404–410, 2005.

[8] Y. Wang, H. Lang, C. de Silva, "A hybrid visual servo controller for robust grasping by wheeled mobile robots", *IEEE/ASME Trans. Mechatronics*, vol. 15, pp. 757–769, 2009.

[9] A. D. Luca, G. Oriolo, and P. R. Giordano. "Image-based visual servoing schemes for nonholonomic mobile manipulators", *Robotica*, vol. 25, no. 2, pp. 131–145. 2007.

[10] Hideaki Tai, Toshiyuki Murakami, "A control of two wheels driven redundant mobile manipulator using a monocular camera system", *Int. J. Intell. Syst. Technol. Appl*, vol. 8 pp. 361–381, 2009.

[11] HB. Wang, Lv L, Li P. "Study on estimating Jacobian matrix on-line visual servo control algorithm", *Journal of System Simulation*, vol. 22, pp. 2934–2937, 2010.

[12] C. Hua, Y. Wang, Y Guan, "Visual tracking control for an uncalibrated robot system with unknown camera parameters", *Robotics and Computer-Integrated Manufacturing*, vol. 30, pp. 19 – 24, 2014.

[13] W. J. Wilson, C. C. Williams, and G. S. Bell, "Relative end-effector control using cartesian position based visual servoing", *IEEE Trans. Robot. Autom*, vol. 12, no. 5, pp. 684–696, 1996.

[14] E. Malis and P. Rives, "Robustness of image-based visual servoing with respect to depth distribution errors", *Proc. of the 2003 IEEE International Conf on Robotics and Automation*, pp. 1056–1061, 2003.

[15] V. Andaluz, R. Carelli, L. Salinas, J. M. Toibero, F. Roberti, "Visual control with adaptive dynamical compensation for 3D target tracking by mobile manipulators", *Mechatronics* , vol. 22, pp. 491–502, 2012.

[16] S. Hutchinson, G. D. Hager, P. I. Corke, "A tutorial on visual servo control", *IEEE Trans on Robot and Auto,* vol. 12, no. 5, pp. 651–670, 1996.

[17] F. Chaumette, S. Hutchinson, "Visual servo control. Part I: Basic approaches", *IEEE Robotics and Auto Magaz,* vol. 13, no. 4, pp. 82–90, 2006.

[18] F. Chaumette, S. Hutchinson, "Visual servo control. Part II: Advanced approaches", *IEEE Robotics and Auto Magaz,* vol. 14, no. 1, pp. 109–118, 2007.

[19] F. Bensalah, F. Chaumette, "Compensation of abrupt motion changes in target tracking by visual servoing", *Proc. 1995 IEEE/RSJ Inter Conf on Intel Robots and Syst*, pp. 181–187, Aug. 1995.

[20] N. V. Tinh, P. T. Cat, P. M. Tuan, B. T. Quyen, "Visual control of integrated mobile robot – pan tilt – camera system for tracking a moving target", *Proc. of the 2014 IEEE International Conf on Robotics and Biomimetics,* pp. 1566–1571, 2014.

[21] V. Andaluz, F. Roberti, L. Salinas, J. Toibero, R. Carelli, "Passivity-based visual feedback control with dynamic compensation of mobile manipulators: Stability and L_2-gain performance analysis", *Robotics and Autonomous Systems*, vol. 66, pp. 64–74, 2015.

[22] P. McKerrow, *Introduction to Robotics*, Addison-Wesley, 1998.

Received on April 27, 2017
Revised on March 07, 2018