

# A NEW CONTROL METHOD FOR STEREO VISUAL SERVOING SYSTEM WITH PAN-TILT PLATFORM

LE VAN CHUNG<sup>1</sup> AND PHAM THUONG CAT<sup>2</sup>

<sup>1</sup>*Thai Nguyen University of Information and Communication Technology;  
chunglv84@gmail.com*

<sup>2</sup>*Institute of Information Technology, Vietnam Academy of Science and Technology;  
ptcat@ioit.ac.vn*



**Abstract.** This paper proposes a new control method for Pan-Tilt stereo camera system to track a moving object when there are many uncertainties in the parameters of both camera and Pan-Tilt platform. If a pair of cameras placed on the Pan-Tilt robot, it is unnecessary for its installation location to be determined accurately. Assuming that the optical parameters like focal length of two cameras in the simulation are the same. A two degree of freedom Pan-Tilt platform holding camera has many uncertain parameters such as inertial moment, Jacobian matrix, the friction and noise impact, etc. The proposed control algorithm is highly adaptive and robust due to the use of a compensating neural network with on-line learning rule. The asymptotic stability of overall control system is proved by Lyapunov's stability method.

**Keywords.** Target tracking, pan/tilt, stereo camera, neural network, Lyapunov stability.

## 1. INTRODUCTION

Tracking moving targets is mainly applied in the security and military. Recently, this research topic attracts many researchers. Many visual servoing systems have been studied and developed. Most of all use one [3, 6] or two cameras to track moving targets. With stereo visual servoing system [7, 8, 13] the posture of the target can be determined in 3D Cartesian coordinates. From control methods point of view one can classify the visual servoing systems into kinematic [8] and dynamic controls. With the kinematics control method, the controllers of the systems have to calculate the necessary speeds of robot joints so that the tracking errors must reach 0 [1, 7, 10]. But, the most important condition is that the joints of robot can be controlled exactly at any desired speed. The dynamic controller uses dynamic equations to calculate the necessary torque of robot joints [5, 13]. The result of this control method is more stable than that of the kinematics controller. Due to nonlinearities and many uncertainties of these systems, finding asymptotic stable control algorithms with good performance is challenging. In [10, 11] stereo visual servoing systems with PD controller and Kalman filter are proposed to track a fast moving target. The control performance of the system is analyzed when the feedback gains of motion control and the image capturing frequency are tuned. To get better results, the stereo tracking system [12] uses adaptive pan-tilt zoom camera and particle filter methods for fast target detection. In this case, the control parameters are calculated from images of two cameras and other setup parameters at the same time.

With the group of eye-to-hand camera system, the system is also tracking target very well, even when the target and the robot move [7] or mobile obstructions appear on the way [2]. From capturing

images, the image velocity [8] can be estimated to track the target by using classical controller. However, it doesn't consider nonlinearities and uncertainties of the parameters in the system.

In this paper, a neural control method is proposed for Pan-Tilt stereo camera system to track a moving object when there are many uncertainties in the parameters of both camera and Pan-Tilt platform.

The paper is organized as follows. Section 2 presents how to obtain the kinematic model of moving target tracking stereo robot system. Section 3 gives a new neural control algorithm for the uncertain visual servo system. Section 4 shows several simulations on PC to check the validity of proposed control algorithm. Finally, the main results and some conclusions are summarized in Section 5.

## 2. KINEMATIC MODEL OF STEREO VISUAL SERVOING SYSTEM WITH UNCERTAIN PARAMETERS

A pan/tilt platform, that is a two degree of freedom robot, can rotate simultaneously in pan and tilt directions. Two cameras are mounted on the robot arm. The target moves in the cameras space and its image obtained from cameras depends on the movement of pan/tilt robots. The image collected from each camera is processed to find the center of the targets image. However, image processing algorithms will be presented in another report. The objective of this paper is finding the controller to control the pan-tilt camera system, tracking moving targets so that the tracking error converges on zero.

In Figure 1, the coordinates are assigned to robot joints and cameras. It is going to determine the kinematic equation of this stereo visual servoing system.

### 2.1. Determination of Image Jacobian matrix

Parameters such as distance, coordinates of target in 3D space can be determined by using two cameras. But in the moving target tracking problem, the coordinate parameters without distance parameter to target are just used because it is needed to control the camera system toward target only. Figure 2 shows the relationship between the camera coordinate system and its image.

Notations: Left camera coordinate system is  $O_L X_L Y_L Z_L$  with the origin located at the focal point of left camera, right camera coordinate system is  $O_R X_R Y_R Z_R$ , with the origin located at the focal point of right camera and camera coordinate system is  $O_C X_C Y_C Z_C$  with the origin located at the midpoint of origin of two cameras. The photo frame is specified in the front and perpendicular to the Y-axis at the center, the axis  $U, V$  parallel to the axis  $Z, X$  of camera, respectively.

**Assumption 1.** Pan angle is  $\theta_1$ , its rotation around the axis  $z_0$  in the original coordinate of the platform pan/tilt, Tilt angle is  $\theta_2$ , its rotation around the axis  $z_1$  in the coordinate system  $O_1 X_1 Y_1 Z_1$  of the platform pan/tilt.

The feature point coordinates of the target obtained from left cameras image is named as  $(U_L, V_L)$  and right cameras is  $(U_R, V_R)$  on two axes  $(U, V)$ . Following the assumption 1, two cameras have the same height so the coordinates of obtained images are the same on  $V$  axis or  $V_R = V_L$ . From Figure 2, the coordinates of feature point on the left image frame  $(U_L, V_L)$  and the right image frame  $(U_R, V_R)$  with  $V_L = V_R = V$  are transformed to  $(Z, Y)$  and  $(X, Y)$  plane as shown in Figures 3 and 4. There are geometrical relations in the coordinate  $O_C X_C Y_C Z_C$ :

$$U_L = \frac{f}{Y} \left( \frac{K}{2} + Z \right); \quad V_L = \frac{f}{Y} X; \quad U_R = -\frac{f}{Y} \left( \frac{K}{2} - Z \right) \quad (1)$$

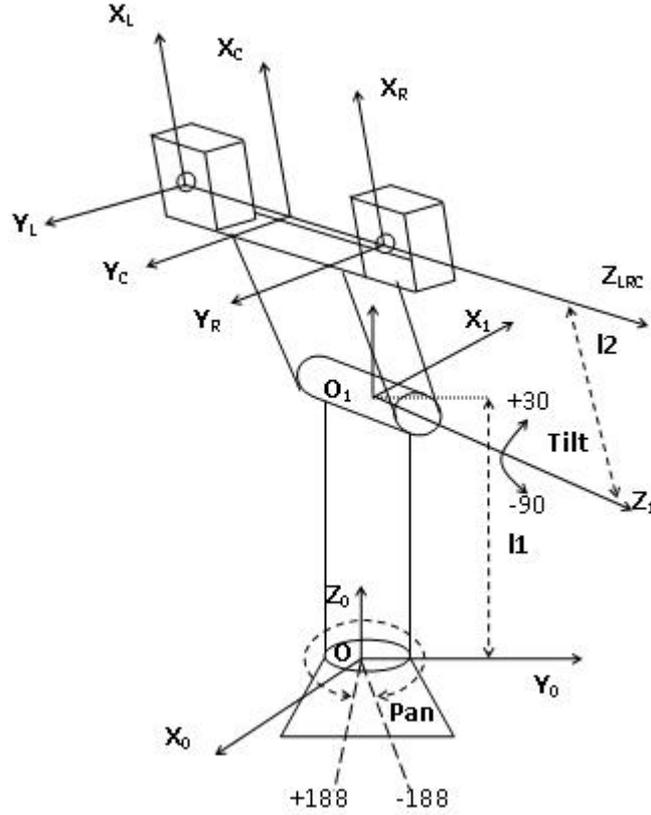


Figure 1: Robot-camera coordinates.

with:

$$\frac{1}{Y} = \frac{U_L - U_R}{f.K} \tag{2}$$

From Eqs. (1) , (2), the coordinates of the target point  $Q(X, Y, Z)$  [2] are calculated in OC coordinates:

$$Q = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{2}{U_R - U_L} \begin{bmatrix} \frac{K}{2} V_L \\ \frac{K}{2} f \\ \frac{K}{4} (U_L + U_R) \end{bmatrix}. \tag{3}$$

$K$  is the distance between the optical axis of two cameras;  $f_L = f_R = f$  is the focal length of the camera lens.

When the target moves, the Pan-Tilt platform must be controlled in order to follow the target. If the translational velocity vector of the camera's origin OC is denoted by

$${}^C \mathbf{v} = [ T_{XC} \quad T_{YC} \quad T_{ZC} ]^T$$

and angular velocity vector is denoted by  ${}^C \boldsymbol{\Omega} = [ \omega_{XC} \quad \omega_{YC} \quad \omega_{ZC} ]^T$ , the velocity vector of

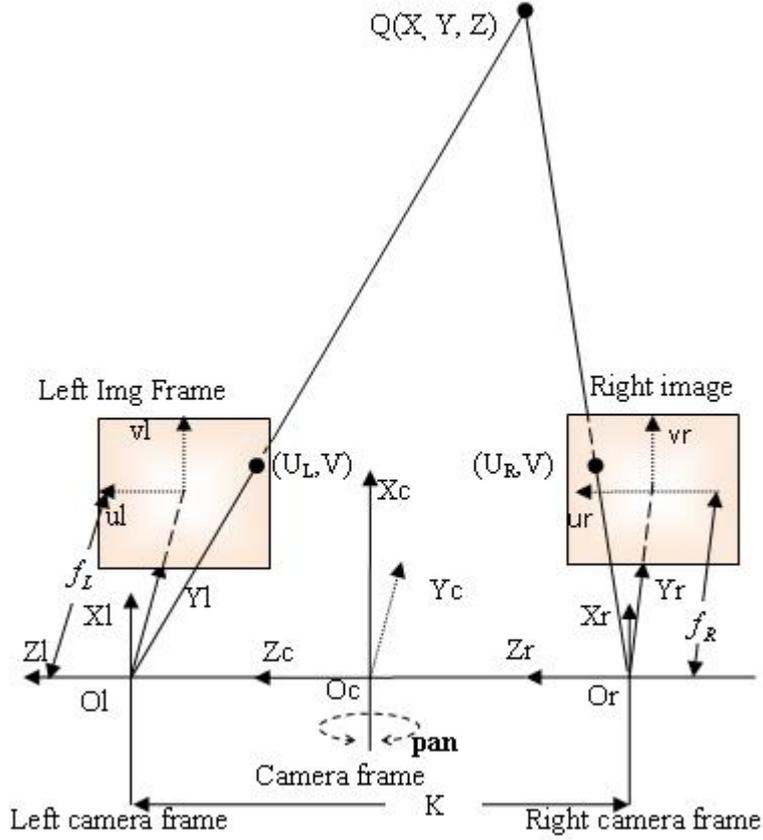


Figure 2: Camera system model.

point  $Q = [ X \ Y \ Z ]^T$  [8] in the camera coordinate is:  $\dot{Q} = {}^C\mathbf{v} + {}^C\boldsymbol{\Omega} \times Q$  or:

$$\dot{X} = T_{XC} + Z\omega_{YC} - Y\omega_{ZC}, \quad (4)$$

$$\dot{Y} = T_{YC} - Z\omega_{XC} + X\omega_{ZC}, \quad (5)$$

$$\dot{Z} = T_{ZC} + Y\omega_{XC} - X\omega_{YC}. \quad (6)$$

The velocity relationships between the movements seen in the camera frame  ${}^C\mathbf{v}, {}^C\boldsymbol{\Omega}$  and the movements seen in the target frame  ${}^T\mathbf{v}, {}^T\boldsymbol{\Omega}$  are:

$${}^C\mathbf{v} = [ T_X \ T_Y \ T_Z ] = -{}^T\mathbf{v},$$

$${}^C\boldsymbol{\Omega} = [ \omega_X \ \omega_Y \ \omega_Z ] = -{}^T\boldsymbol{\Omega}.$$

Take the derivative of Eq. (1) and substitute into Eqs. (4), (5) and (6) to obtain the velocity rela-

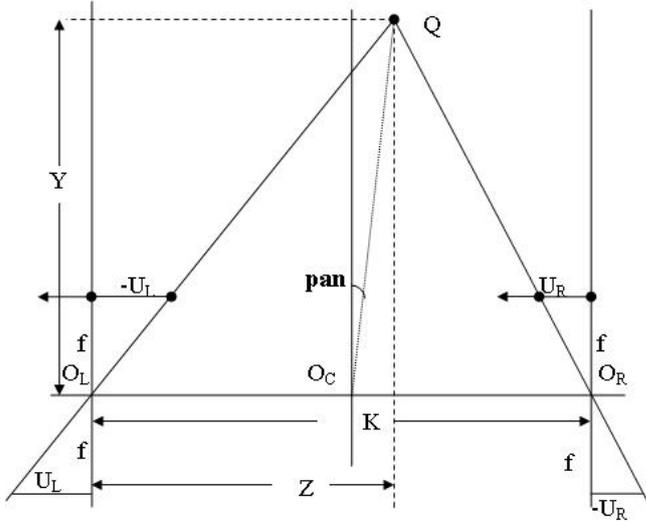


Figure 3: Coordinates of feature point on Z, Y axis.

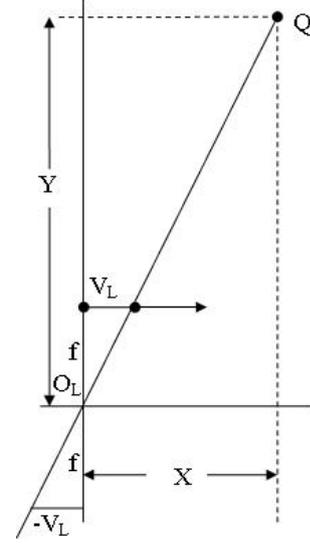


Figure 4: Coordinates of feature point on X, Y axis.

tionship Eq. (7):

$$\begin{aligned} \begin{bmatrix} \dot{U}_L \\ \dot{V}_L \\ \dot{U}_R \end{bmatrix} &= \begin{bmatrix} \frac{d}{dt} \left( \frac{f}{Y} \left( \frac{K}{2} + Z \right) \right) \\ \frac{d}{dt} \left( \frac{f}{Y} X \right) \\ \frac{d}{dt} \left( -\frac{f}{Y} \left( \frac{K}{2} - Z \right) \right) \end{bmatrix} \\ &= \begin{bmatrix} \frac{-f}{Y} & 0 & \frac{U_L}{Y} & \frac{U_L V_L}{f} & -\frac{f^2 + U_L^2}{f} + \frac{K U_L}{2Y} & V_L \\ 0 & \frac{-f}{Y} & \frac{V_L}{Y} & \frac{f^2 + V_L^2}{f} & -\frac{U_L V_L}{f} + \frac{K V_L}{2Y} & -U_L + \frac{K f}{2Y} \\ \frac{-f}{Y} & 0 & \frac{U_R}{Y} & \frac{U_R V_L}{f} & -\frac{f^2 + U_R^2}{f} - \frac{K U_R}{2Y} & V_L \end{bmatrix} \begin{bmatrix} T_Z \\ T_X \\ T_Y \\ \omega_Z \\ \omega_X \\ \omega_Y \end{bmatrix}. \end{aligned} \quad (7)$$

Via substituting Eq. (2) into Eq. (7) yields:

$$\begin{bmatrix} \dot{U}_L \\ \dot{V}_L \\ \dot{U}_R \end{bmatrix} = \begin{bmatrix} \frac{U_R - U_L}{K} & 0 & \frac{U_L(U_L - U_R)}{fK} & \frac{U_L V_L}{f} & -\frac{2f^2 + U_L^2 + U_L U_R}{2f} & V_L \\ 0 & \frac{U_R - U_L}{K} & \frac{V_L(U_L - U_R)}{fK} & \frac{f^2 + V_L^2}{f} & -\frac{V_L(U_L + U_R)}{2f} & -\frac{U_L + U_R}{2f} \\ \frac{U_R - U_L}{K} & 0 & \frac{U_R(U_L - U_R)}{fK} & \frac{U_R V_L}{f} & -\frac{2f^2 + U_R^2 + U_L U_R}{2f} & V_L \end{bmatrix} \begin{bmatrix} T_Z \\ T_X \\ T_Y \\ \omega_Z \\ \omega_X \\ \omega_Y \end{bmatrix} \quad (8)$$

Therefore, Eq. (8) describes the relationship between the velocity of targets image on 2 cameras and the velocity of the target. Rewriting Eq. (8) in matrix form, results in:

$$\dot{\mathbf{m}} = \mathbf{J}_{imag}(\mathbf{m})\mathbf{u}, \quad (9)$$

where:  $\mathbf{m} = [U_L V_L U_R]^T$  is the image feature vector consisting three components (converted into coordinate  $O_C X_C Y_C Z_C$ ),  $\dot{\mathbf{m}} = [\dot{U}_L \dot{V}_L \dot{U}_R]^T$  is the velocity of the image feature vector,  $\mathbf{J}_{imag}(\mathbf{m})$

is the image Jacobi matrix:

$$\mathbf{J}_{imag}(\mathbf{m}) = \begin{bmatrix} \frac{U_R - U_L}{K} & 0 & \frac{U_L(U_L - U_R)}{fK} & \frac{U_L V_L}{f} & -\frac{2f^2 + U_L^2 + U_L U_R}{2f} & V_L \\ 0 & \frac{U_R - U_L}{K} & \frac{V_L(U_L - U_R)}{fK} & \frac{f^2 + V_L^2}{f} & -\frac{V_L(U_L + U_R)}{2f} & -\frac{U_L + U_R}{2f} \\ \frac{U_R - U_L}{K} & 0 & \frac{U_R(U_L - U_R)}{fK} & \frac{U_R V_L}{f} & -\frac{2f^2 + U_R^2 + U_L U_R}{2f} & V_L \end{bmatrix}, \quad (10)$$

$\mathbf{u} = [T_X \ T_Y \ T_Z \ \omega_X \ \omega_Y \ \omega_Z]^T$  is the velocity vector of the camera system.

## 2.2. Kinematic equations of Pan-Tilt platform

At first, kinematic equations of Pan-Tilt platform are identified based on the homogeneous transformation between coordinate frames attached to the joints of platform [9]. The Homogeneous transformation matrix transforming coordinates from  $O_C$  to  $O_0$  in Figure 1 is:

$${}^0R_C = {}^0R_1 {}^1R_C = \begin{bmatrix} C_1 C_2 & -C_1 S_2 & S_1 & l_2 C_1 C_2 \\ S_1 C_2 & -S_1 S_2 & -C_1 & l_2 S_1 C_2 \\ S_2 & C_2 & 0 & l_1 + l_2 S_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (11)$$

The second, internal variable vector of the pan-tilt platform is  $\mathbf{q} = [\theta_1, \theta_2]^T$  and extrinsic variable vector of grip head is  $\mathbf{x} = [x \ y \ z \ \phi_X \ \phi_Y \ \phi_Z]^T$ , in which  $\phi_X, \phi_Y, \phi_Z$  are the rotation angle in three axes X, Y, Z. Hence, vector  $\mathbf{x}$  describes the position and orientation of the coordinate  $O_C X_C Y_C Z_C$  in robot coordinate  $O_0 X_0 Y_0 Z_0$ . So, the velocity vector follows:

$$\begin{aligned} \dot{\mathbf{x}} &= [\dot{x} \ \dot{y} \ \dot{z} \ \omega_X \ \omega_Y \ \omega_Z]^T = \mathbf{u}, \\ &= [T_X \ T_Y \ T_Z \ \omega_X \ \omega_Y \ \omega_Z]^T. \end{aligned} \quad (12)$$

Denote  $\mathbf{J}_{robot}$  the Jacobian of the Pan-Tilt platform yields:

$$\dot{\mathbf{x}} = \mathbf{J}_{robot}(\mathbf{q})\dot{\mathbf{q}}, \quad (13)$$

where  $\mathbf{J}_{robot}$  can be determined as follows [9]:

$$\mathbf{J}_{robot} = \begin{bmatrix} l_2 C_1 C_2 & 0 \\ l_2 S_1 C_2 & 0 \\ l_2 S_2 & l_1 \\ 0 & S_1 \\ 0 & -C_1 \\ 1 & 0 \end{bmatrix}. \quad (14)$$

## 2.3. Formulation of stereo visual servoing problem with uncertain parameters

It is to calculate the image feature error from the measured image feature vector  $\mathbf{m}$  and the desired image feature vector  $\mathbf{m}_d$ :

$$\varepsilon = \mathbf{m} - \mathbf{m}_d. \quad (15)$$

The kinematic control problem of stereo visual servoing is to find control law  $\dot{\mathbf{q}} = \mathbf{K}(\varepsilon)$  to control the system tracking the moving target so the tracking error  $\varepsilon$  converges on zero. The closed-loop of the system shown as:

$$\begin{aligned}\dot{\mathbf{m}} &= \mathbf{J}_{imag}(\mathbf{m})\mathbf{u}, \\ \mathbf{u} = \dot{\mathbf{x}} &= \mathbf{J}_{robot}(\mathbf{q})\dot{\mathbf{q}},\end{aligned}\quad (16)$$

$$\dot{\mathbf{q}} = \mathbf{K}(\varepsilon). \quad (17)$$

Assuming that the parameters of the robot and the camera are not known exactly, now the Jacobian matrices are described as:

$$\begin{aligned}\mathbf{J}_{imag}(\mathbf{m}) &= \hat{\mathbf{J}}_{imag}(\mathbf{m}) + \Delta\mathbf{J}_{imag}(\mathbf{m}), \\ \mathbf{J}_{robot}(\mathbf{q}) &= \hat{\mathbf{J}}_{robot}(\mathbf{q}) + \Delta\mathbf{J}_{robot}(\mathbf{q}),\end{aligned}\quad (18)$$

where  $\hat{\mathbf{J}}_{imag}(\mathbf{m})$ ,  $\hat{\mathbf{J}}_{robot}(\mathbf{q})$  are the estimated values,  $\Delta\mathbf{J}_{imag}(\mathbf{m})$ ,  $\Delta\mathbf{J}_{robot}(\mathbf{q})$  are the difference between estimated values and actual values. The exact values of their parameters are not known. Substitute Eq. (16) into Eq. (9) to obtain:

$$\begin{aligned}\dot{\mathbf{m}} &= \mathbf{J}_{imag}(\mathbf{m})\mathbf{u} = \mathbf{J}_{imag}(\mathbf{m})\mathbf{J}_{robot}(\mathbf{q})\dot{\mathbf{q}}, \\ \dot{\mathbf{m}} &= \mathbf{J}\dot{\mathbf{q}}.\end{aligned}\quad (19)$$

The combined Jacobian matrix is:

$$\mathbf{J} = \mathbf{J}_{imag}(\mathbf{m})\mathbf{J}_{robot}(\mathbf{q}). \quad (20)$$

Via substituting the Jacobian matrix of image Eq. (10) and Jacobian matrix of robot Eq. (14) into Eq. (20) results in Eq. (21)

$$\mathbf{J} = \begin{bmatrix} \frac{U_R - U_L}{K} l_2 C_1 C_2 + \frac{U_L(U_L - U_R)}{fK} l_2 S_2 + V_L & \frac{U_L(U_L - U_R)}{fK} l_1 + \frac{U_L V_L}{f} S_1 + \frac{2f^2 + U_L^2 + U_L U_R}{2f} C_1 \\ \frac{U_R - U_L}{K} l_2 S_1 C_2 + \frac{V_L(U_L - U_R)}{fK} l_2 S_2 - \frac{U_L + U_R}{2f} & \frac{V_L(U_L - U_R)}{fK} l_1 + \frac{f^2 + V_L^2}{f} S_1 + \frac{V_L(U_L + U_R)}{2f} C_1 \\ \frac{U_R - U_L}{K} l_2 C_1 C_2 + \frac{U_R(U_L - U_R)}{fK} l_2 S_2 + V_L & \frac{U_R(U_L - U_R)}{fK} l_1 + \frac{U_R V_L}{f} S_1 + \frac{2f^2 + U_R^2 + U_L U_R}{2f} C_1 \end{bmatrix} \quad (21)$$

Thus the dimension of combined Jacobian matrix  $J$  is  $(3 \times 2)$ . By substituting Eq. (18) into Eq. (20), following some transformation, we get:

$$\begin{aligned}\mathbf{J} &= \mathbf{J}_{imag}(\mathbf{m})\mathbf{J}_{robot}(\mathbf{q}) = \hat{\mathbf{J}} + \Delta\mathbf{J}, \\ \hat{\mathbf{J}} &= \hat{\mathbf{J}}_{imag}(\mathbf{m})\hat{\mathbf{J}}_{robot}(\mathbf{q}).\end{aligned}\quad (22)$$

$\Delta\mathbf{J} = \hat{\mathbf{J}}_{imag}(\mathbf{m})\Delta\mathbf{J}_{robot}(\mathbf{q}) + \Delta\mathbf{J}_{imag}(\mathbf{m})\hat{\mathbf{J}}_{robot}(\mathbf{q}) + \Delta\mathbf{J}_{imag}(\mathbf{m})\Delta\mathbf{J}_{robot}(\mathbf{q})$  with

$$\hat{\mathbf{J}}$$

is known,  $\Delta\mathbf{J}$  is the uncertain part of  $J$ . Substitute Eq. (22) into Eq. (19):

$$\dot{\mathbf{m}} = \hat{\mathbf{J}}\dot{\mathbf{q}} + \Delta\mathbf{J}\dot{\mathbf{q}}. \quad (23)$$

Set  $f = \Delta\mathbf{J}\dot{\mathbf{q}}$  yields:

$$\dot{\mathbf{m}} = \hat{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{f}. \quad (24)$$

Now, it is to find control law  $\dot{\mathbf{q}} = \mathbf{K}(\varepsilon)$  applied to equation (23) in order to  $\varepsilon = (\mathbf{m} - \mathbf{m}_d) \rightarrow 0$ .

### 3. CONTROL LAW DESIGN

Assuming that, robot operates in working area where Jacobian matrix  $\mathbf{J}$  is not degenerate, so it is:

$$\|f\| = \|\Delta\mathbf{J}\dot{\mathbf{q}}\| \leq f_0. \quad (25)$$

Therefore,  $f$  is a continuous and bounded function. Now, the task is to find the joints velocity  $\dot{q}$  in order to make the image feature error vector reach to 0. Control law is chosen as follows:

$$\dot{\mathbf{q}} = \hat{\mathbf{J}}^+[(\dot{\mathbf{m}}_d - \mathbf{K}\varepsilon) + \mathbf{u}_1]. \quad (26)$$

where  $\hat{\mathbf{J}}^+ = \hat{\mathbf{J}}^T(\hat{\mathbf{J}}\hat{\mathbf{J}}^T)^{-1}$  is the pseudo-inverse matrix of the combined Jacobian matrix  $\mathbf{J}$ ;  $\mathbf{u}_1$  is a control component to compensate the effect of uncertain parameters. It will be determined later. Take the derivatives at the two sides of the Eq. (5) and substitute Eqs. (24) and (26) into it to get:

$$\begin{aligned} \dot{\varepsilon} &= \dot{\mathbf{m}} - \dot{\mathbf{m}}_d = \hat{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{f} - \dot{\mathbf{m}}_d, \\ \dot{\varepsilon} &= -\mathbf{K}\varepsilon + \mathbf{f} + \mathbf{u}_1. \end{aligned} \quad (27)$$

Because the continuous and bounded function  $\mathbf{f}$  is unknown, it can be approximated by a neural network yielding  $\hat{f}$  with suitable learning algorithm, and the control vector  $\mathbf{u}_1$  can be so determined that the system in Eq. (26) becomes asymptotically stable. The structure of the chosen artificial neural network is of RBF type as shown in Figure 5. It has three layers: the input layer includes the three components of error  $\varepsilon$ , the output layer includes 3 linear neurons and hidden layer contains neurons with Gaussian output function:

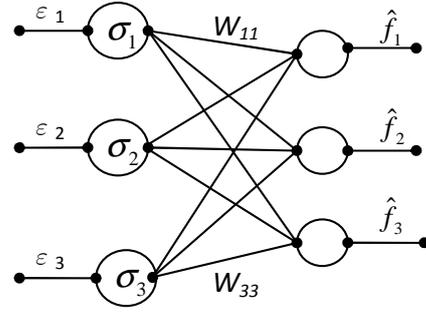


Figure 5: RBF network approximating function  $\mathbf{f}$

$$\sigma_j = \exp\left(-\frac{(\varepsilon_j - c_j)^2}{\lambda_j^2}\right); \quad j = 1, 2, 3, \quad (28)$$

where  $c_j, \lambda_j$  are selective parameters of Gauss function. The output values of the network are approximate function  $\hat{f}$  as shown in Figure 5.

According to the Stone–Weierstrass theorem [4], the structure of the above RBF network above can approximate bounding and continuous unknown  $\mathbf{f} \in R^3$ , and  $\mathbf{f}$  is described by the equation:

$$\mathbf{f} = \hat{\mathbf{f}} + \beta, \quad (29)$$

$$\hat{\mathbf{f}} = \mathbf{W}\sigma, \quad (30)$$

where  $\beta$  denotes bounded approximation error,

$$\|\beta\| \leq \beta_0. \quad (31)$$

$\mathbf{W}$  is the weight matrix of the neural network, updated on-line from the real-time measurement value of error  $\varepsilon$ .

**Theorem 1.** *The stereo camera system described by the model in Esq. (9), (16) with uncertain parameters controlled by the neural network defined by Eqs. (28), (29) and (30) will track moving targets with the error  $\varepsilon, \dot{\varepsilon} \rightarrow \mathbf{0}$  if the speed of the Pan-Tilt joints is determined by the Eqs. (32), (33), (34) and learning rules (35):*

$$\dot{\mathbf{q}} = \hat{\mathbf{J}}^+[(\dot{\mathbf{m}}_d - \mathbf{K}\varepsilon) + \mathbf{u}_1] = \hat{\mathbf{J}}^+(\dot{\mathbf{m}}_d - \mathbf{K}\varepsilon) + \hat{\mathbf{J}}^+\mathbf{u}_1 = \mathbf{u}_0 + \mathbf{u}_1, \quad (32)$$

$$\mathbf{u}_0 = \hat{\mathbf{J}}^+ [\dot{\mathbf{m}}_d - \mathbf{K}\varepsilon], \quad (33)$$

$$\mathbf{u}_1 = \left[ (\eta - 1)\mathbf{W}\sigma - \delta \frac{\varepsilon}{\|\varepsilon\|} \right], \quad (34)$$

$$\begin{aligned} \mathbf{u}_1 &= \hat{\mathbf{J}}^+\mathbf{u}_1, \\ \dot{\mathbf{W}} &= -\eta\varepsilon\sigma^T, \end{aligned} \quad (35)$$

where  $\mathbf{K}$  is a positive definite symmetric matrix,  $\mathbf{K} = \mathbf{K}^T > 0$ , the coefficients  $\eta > 1$ ,  $\delta > 0$ .

The block diagram of overall control system can be seen on Figure 6. The signal  $\dot{\mathbf{q}}$  consists of two components:  $\mathbf{u}_0$  is a feedback component,  $\mathbf{u}_1$  a neural network component with on-line learning algorithm to compensate uncertain parameters. This theorem can be proved by Lyapunovs stability method that guarantees the global asymptotic stability of system:

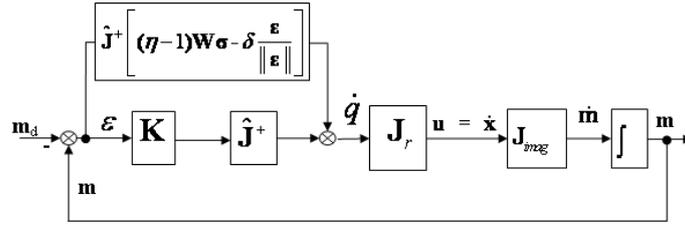


Figure 6: Structure of proposed visual tracking system with many uncertain parameters.

*Proof.* Chose a positive definite candidate Lyapunov function as follows:

$$V = \frac{1}{2} \left( \varepsilon^T \varepsilon + \sum_{i=1}^3 \mathbf{w}_i^T \mathbf{w}_i \right), \quad (36)$$

where  $\mathbf{w}_i$  is  $i_{th}$  column vector of weight matrix  $\mathbf{W}$ . And  $V > 0$  when  $\varepsilon, \mathbf{w}_i \neq \mathbf{0}$ ;  $V = 0$  if and only if  $\varepsilon, \mathbf{w}_i = \mathbf{0}$ ,  $i = 1, 2, 3$ ;  $V \rightarrow \infty$  if  $\varepsilon, \mathbf{w}_i \rightarrow \infty$ . Taking the derivative of  $V$  along time yields:

$$\dot{V} = \varepsilon^T \dot{\varepsilon} + \sum_{i=1}^3 \mathbf{w}_i^T \dot{\mathbf{w}}_i, \quad (37)$$

By substituting Eq. (27) into Eq. (37), the derivative  $\dot{V}$  takes the form:

$$\dot{V} = -\varepsilon^T \mathbf{K}\varepsilon + \varepsilon^T (\mathbf{u}_1 + \mathbf{f}) + \sum_{i=1}^3 \mathbf{w}_i^T \dot{\mathbf{w}}_i, \quad (38)$$

With on-line learning algorithm Eq. (35) to get:

$$\dot{\mathbf{w}}_i = -\eta\varepsilon\sigma_i; \quad i = 1, 2, 3; \quad (39)$$

We can identify that:

$$\sum_{i=1}^3 \mathbf{w}_i^T \dot{\mathbf{w}}_i = \sum_{i=1}^3 -\eta \mathbf{w}_i^T \varepsilon \sigma_i = -\eta \varepsilon^T \sum_{i=1}^3 \mathbf{w}_i \sigma_i = -\eta \varepsilon^T \mathbf{W} \sigma. \quad (40)$$

Substituting Eq. (29), Eq. (31) and Eq. (40) into Eq. (38) results in:

$$\dot{V} = -\varepsilon^T \mathbf{K} \varepsilon + \varepsilon^T (\mathbf{u}_1 - (\eta - 1) \mathbf{W} \sigma + \beta). \quad (41)$$

Via Substituting  $\mathbf{u}_1 = (\eta - 1) \mathbf{W} \sigma - \delta \frac{\varepsilon}{\|\varepsilon\|}$  from Eq. (34) into Eq. (41) yields:

$$\begin{aligned} \dot{V} &= -\varepsilon^T \mathbf{K} \varepsilon + \varepsilon^T \left( -\delta \frac{\varepsilon}{\|\varepsilon\|} + \beta \right) \leq -\varepsilon^T \mathbf{K} \varepsilon, \\ -\delta \|\varepsilon\| + \|\varepsilon\| \cdot \|\beta\| &\leq -\varepsilon^T \mathbf{K} \varepsilon - \delta \|\varepsilon\| + \beta_0 \|\varepsilon\|. \end{aligned} \quad (42)$$

Because  $\delta$  is a positive optional parameter of the neural network, we choose  $\delta = \beta_0 + \mu > 0$ ; where  $\mu > 0$ , then:

$$-\delta \|\varepsilon\| + \|\varepsilon\| \beta_0 \leq -\mu \|\varepsilon\|. \quad (43)$$

Substitution of Eq. (43) to Eq. (42) results in:

$$\dot{V} \leq -\varepsilon^T \mathbf{K} \varepsilon - \mu \|\varepsilon\| \leq 0. \quad (44)$$

$\dot{V}$  is the negative-semi definite function. This implies that  $V(t) \leq V(0)$ , and therefore, that  $\varepsilon$  and  $\mathbf{W}$  are bounded. Because the system is non- autonomous, it cannot conclude that  $\varepsilon$  converge on zero. Barbalat's lemma is used to prove the asymptotic stability of the system. Let us check the uniform continuity of  $\dot{V}$ . The derivative of  $\dot{V}$  is:

$$\begin{aligned} \ddot{V} &= -2\varepsilon^T \mathbf{K} \dot{\varepsilon} + \dot{\varepsilon}^T \beta - \frac{\delta \varepsilon \dot{\varepsilon}}{\|\varepsilon\|} \\ &= -\left( 2\varepsilon^T \mathbf{K} - \beta^T + \frac{\delta \varepsilon}{\|\varepsilon\|} \right) \dot{\varepsilon} \\ &= -\left( 2\varepsilon^T \mathbf{K} + \beta^T - \frac{\delta \varepsilon}{\|\varepsilon\|} \right) \left( \eta \mathbf{W} \sigma - \frac{\delta \varepsilon}{\|\varepsilon\|} + \beta - \mathbf{K} \varepsilon \right). \end{aligned} \quad (45)$$

It is found that  $\ddot{V}$  is bounded because  $\varepsilon$ ,  $\beta$  and  $\mathbf{W}$  are bounded,  $\frac{\varepsilon}{\|\varepsilon\|}$  is the unit vector of  $\varepsilon$  and always bounded,  $\mathbf{K}$  is the positive-definite constant matrix and  $\eta, \delta > 0$ . Thus  $\dot{V}$  is uniformly continuous. According to Barbalat's lemma, it results in  $\varepsilon \rightarrow \mathbf{0}$  as  $t \rightarrow \infty$ . Hence, the system represented in the Eqs. (9), (16) is asymptotically stable and the camera will track moving targets with error that converges on zero as the time approaches the infinity. Theorem 1 as well as the global asymptotic stability of the overall stereo camera visual tracking system using neural network described in Figure 6 has been proven.  $\square$

#### 4. SIMULATION

Simulations are carried out in Matlab, Simulink by use of the pseudo inverse matrix of  $\hat{\mathbf{J}}$  as  $\hat{\mathbf{J}}^+$ .  $\hat{\mathbf{J}}$  is estimated matrix of Jacobian matrix  $\mathbf{J}$  in Eq. (16b). Assuming that, only 80% value of  $\mathbf{J}$  are

estimated as follows:

$$\hat{\mathbf{J}} = 0.8 \begin{bmatrix} \frac{U_R - U_L}{K} l_2 C_1 C_2 + \frac{U_L(U_L - U_R)}{fK} l_2 S_2 + V_L & \frac{U_L(U_L - U_R)}{fK} l_1 + \frac{U_L V_L}{f} S_1 + \frac{2f^2 + U_L^2 + U_L U_R}{2f} C_1 \\ \frac{U_R - U_L}{K} l_2 S_1 C_2 + \frac{V_L(U_L - U_R)}{fK} l_2 S_2 - \frac{U_L + U_R}{2f} & \frac{V_L(U_L - U_R)}{fK} l_1 + \frac{f^2 + V_L^2}{f} S_1 + \frac{V_L(U_L + U_R)}{2f} C_1 \\ \frac{U_R - U_L}{K} l_2 C_1 C_2 + \frac{U_R(U_L - U_R)}{fK} l_2 S_2 + V_L & \frac{U_R(U_L - U_R)}{fK} l_1 + \frac{U_R V_L}{f} S_1 + \frac{2f^2 + U_R^2 + U_L U_R}{2f} C_1 \end{bmatrix}.$$

By using the notation  $\hat{\mathbf{J}} = \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{bmatrix}$ , the pseudo inverse matrix is calculated as:

$$\hat{\mathbf{J}}^+ = (\hat{\mathbf{J}}^T \hat{\mathbf{J}})^{-1} \hat{\mathbf{J}}^T = \begin{bmatrix} \frac{a_1 C - b_1 B}{M} & \frac{a_2 C - b_2 B}{M} & \frac{a_3 C - b_3 B}{M} \\ -\frac{a_1 B + b_1 A}{M} & -\frac{a_2 B + b_2 A}{M} & -\frac{a_3 B + b_3 A}{M} \end{bmatrix}$$

where  $A = a_1^2 + a_2^2 + a_3^2$ ,  $B = a_1 b_1 + a_2 b_2 + a_3 b_3$ ,  $C = b_1^2 + b_2^2 + b_3^2$ ,  $M = (a_1 b_2 - a_2 b_1)^2 + (a_1 b_3 - a_3 b_1)^2 + (a_2 b_3 - a_3 b_2)^2$ . Choose the optional parameters:

$$\eta = 1.5; \delta = 0.2; \sigma_1 = \exp\left(-\frac{(\varepsilon_1 - 1)^2}{25}\right); \sigma_2 = \exp\left(-\frac{(\varepsilon_2 - 2)^2}{9}\right); \sigma_3 = \exp\left(-\frac{(\varepsilon_3 - 0.5)^2}{16}\right),$$

$$\mathbf{K} = \text{diag}\langle 0.4 \quad 0.7 \quad 1 \rangle.$$

The parameters of pan/tilt camera system are:  $l_1 = 0.22$  m;  $l_2 = 0.1$  m;  $K = 0.2$  m;  $f = 12.5$  mm  $\sim 360$  pixels  $U_L, V_L, U_R$ , and  $V_R$  are left and right image coordinates. The unit of these parameters is a pixel. The coordinates of the initial image vector are:  $m = [U_L, V_L, U_R] = [0, 0, 0]$ .

**Simulation 1** (Fixed target). Camera center at the initial time:  $m(0) = [-40, 30, 0]$  (pixel); Image coordinates of the target stand still at  $mt = [-20, 0, 20]$  (pixel); Simulation results are given in Figures 7 and 8.

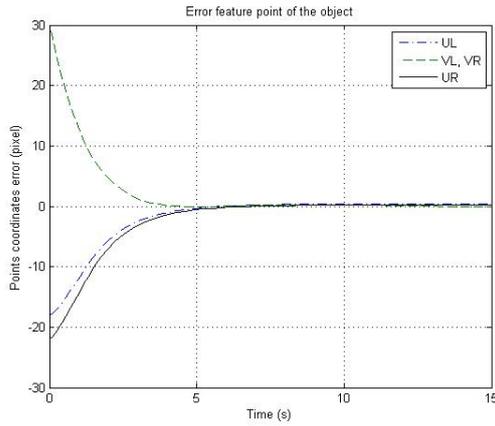


Figure 7: Image feature error when using neural control  $\mathbf{u}_1 \neq 0$ .

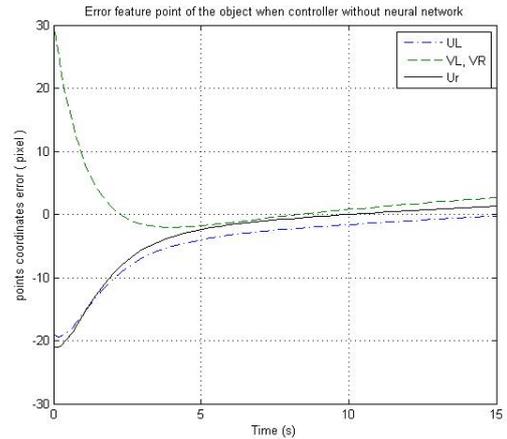


Figure 8: Image feature error when no neural control is used  $\mathbf{u}_1 = 0$ .

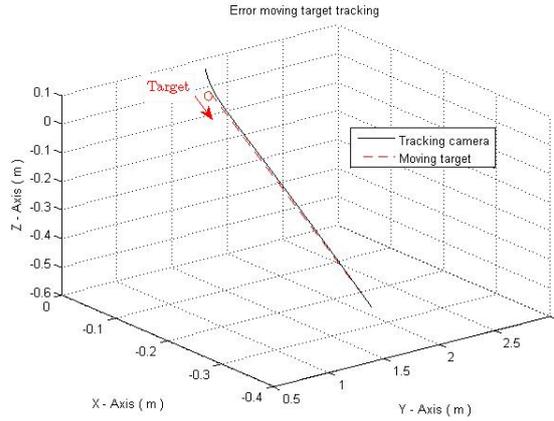


Figure 9: Tracking error coordinates when the target moves along a straight line.

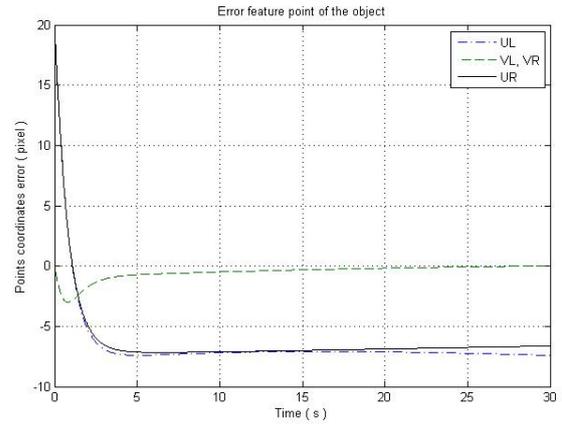


Figure 10: Image feature error coordinates when the target moves along a straight line.

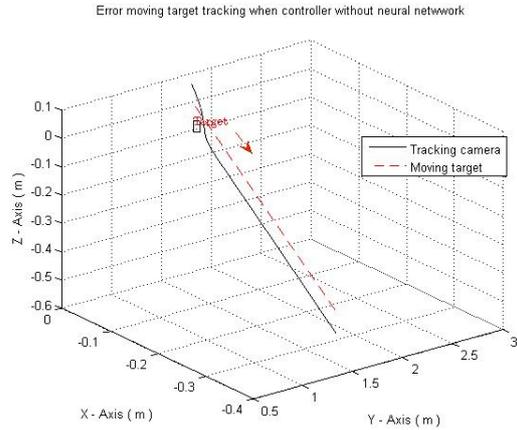


Figure 11: Tracking error coordinates when the target moves along a straight line and no neural control is used.

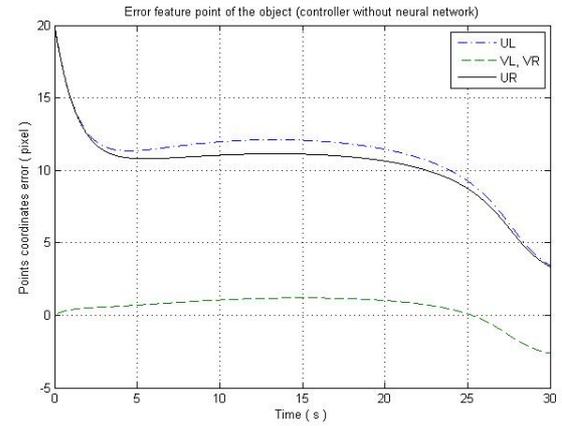


Figure 12: Image feature error coordinates when the target moves along a straight line and no neural control is used.

**Simulation 2** (Moving target in a straight line). Moving target from point  $A(0 \text{ m}, 1.8 \text{ m}, 0 \text{ m})$  to  $B(-0.3 \text{ m}, 1.8 \text{ m}, -0.5 \text{ m})$  on the plane  $Z_C O_C X_C$  and far away from coordinate origin  $Y_C = 1.8 \text{ m}$  in the camera coordinate system  $O_C X_C Y_C Z_C$ . Targets moving time  $T = 30 \text{ s}$  with moving speed is  $v \sim 2 \text{ cm/s}$ . Simulation results are given in Figures 9 and 10:

**Simulation 3** (Moving target in an arc). The target follows the circular arc with center coordinates at the origin  $O(0,0)$ , radius  $r = 1$  on the plane  $Z_C O_C X_C$  and far away from coordinate origin  $Y_C = 1.8 \text{ m}$  in the camera coordinate system  $O_C X_C Y_C Z_C$ . Target moves from point  $A(0 \text{ m}, 1.8 \text{ m}, 1 \text{ m})$  to  $B(1 \text{ m}, 1.8 \text{ m}, 0 \text{ m})$  following arc in time  $T = 30 \text{ s}$  with speed  $v \sim 3 \text{ cm/s}$ . Simulation results are given in Figures 13-16.

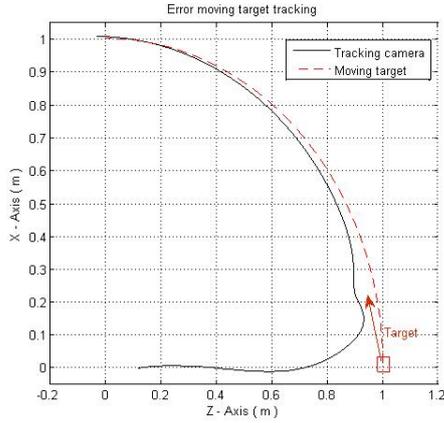


Figure 13: Tracking error coordinates when the target moves along an arc.

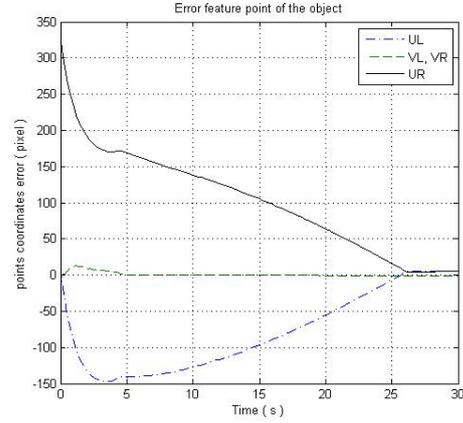


Figure 14: Image feature error coordinates when the target moves along an arc.

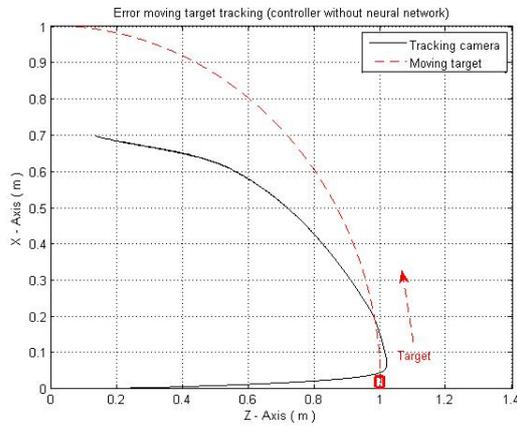


Figure 15: Tracking error coordinates when the target moves along an arc and no neural control is used.

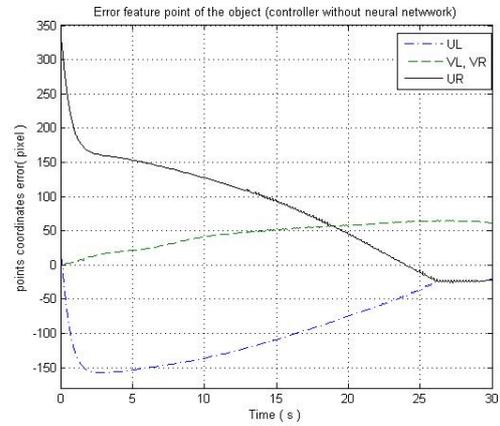


Figure 16: Image feature error coordinates when the target moves along an arc and no neural control is used

**Simulation 4** (Moving target in arcs with acceleration). The target follows the circular arc similar to the arc of simulation 3. However, in the first 1/6 arc, the velocity increases steadily with acceleration  $1 \text{ cm/s}^2$ , after 3 seconds it will move with the constant speed  $3 \text{ cm/s}$ . In 1/6 end of the arc, its moving speed reduces with the deceleration  $-1 \text{ cm/s}^2$ . Simulation results are given in Figures 17 and 18.

**Simulation 5** (Moving target in a semicircle including arc and diameter with changing velocity). Target moves in a semicircle including arc. The arc is the same as simulation 4. After 1/6 arc in the first 1/4 circle, target moves with steady speed  $2 \text{ cm/s}^2$  then it changes speed to  $10 \text{ cm/s}$ . In the 1/4 end of the circle it moves with steady speed  $2 \text{ cm/s}^2$ . Simulation results are given in Figures 19 and 20.

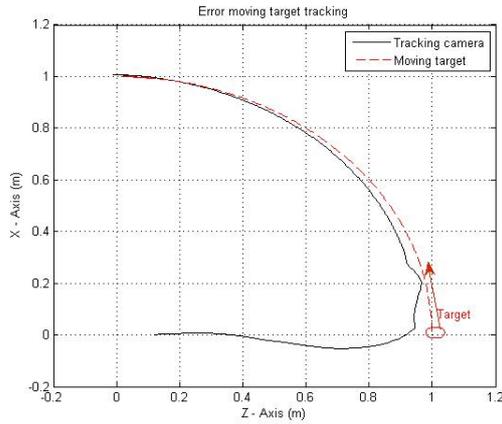


Figure 17: Tracking error coordinates when the target moves along an arc with changing acceleration.

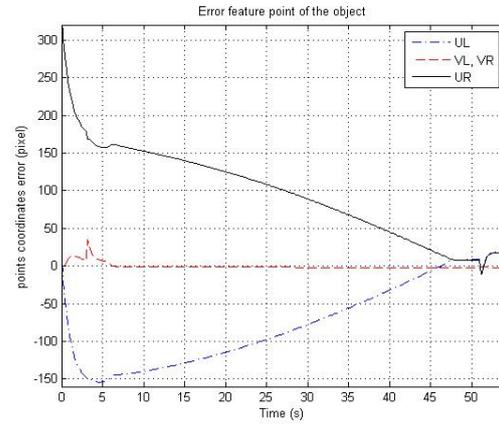


Figure 18: Image feature error coordinates when the target moves along an arc with changing acceleration.

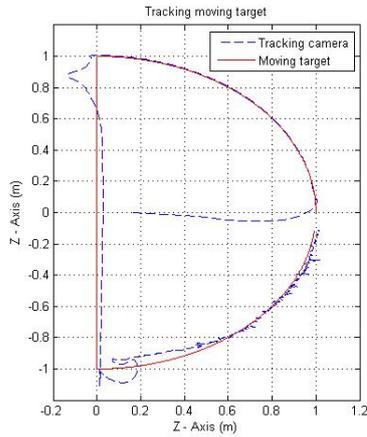


Figure 19: Tracking error coordinates when the target moves along a semicircle with changing velocity.

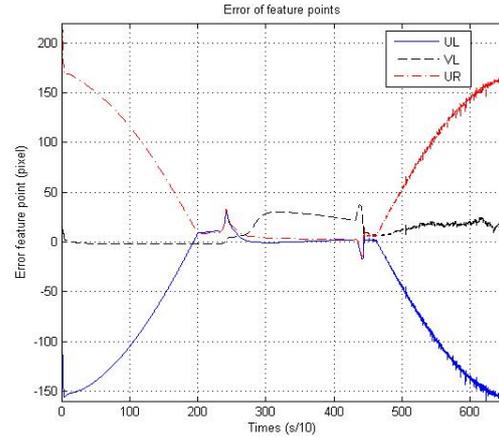


Figure 20: Image feature error coordinates when the target moves along a semicircle with changing velocity.

## Comparison and evaluation

The simulation results show that the system with neural network controller successfully compensates the effects of uncertain parameters. It has a response time faster and less error when there is no neural control  $u_1 = 0$ .

In simulation 1 result—fixed target Figure 7 and simulation 2 - moving target in a straight line Figures 9 and 10 it is found that the error converges on zero. While target follows circular arc, the system with neural control tracks target with better quality than without using neural network as shown in Figures 13 and 13. Because the effects of uncertain parameters are well compensated by on-line learning neural network continuously, the impact of noise and uncertain parameters is reduced. Besides, it keeps the system steady and suppressive error tracking. The system still tracks well when

the target moves with variant velocity (Figures 17 and 19). Quality of moving target tracking is acceptable. When the target changes to steady motion, the initial error of system increases, but it still tracks target rapidly. At the end of a circular arc, the target moves steadily slow but the absolute error of tracking system does not change.

The controller without using neural network can track fixed target Figure 8 or moving target in a straight line Figures 11 and 12 with acceptable error. However, when moving target follows circular arcs Figures 15 and 16, the result of tracking has greater error, even loss of tracking in the final stage of circular arcs may happen.

## 5. CONCLUSION

This paper proposes a new control method for a stereo visual servoing system using neural networks with on-line learning rules to compensate for the impacts of uncertain parameters such as inertial torque, Jacobian matrix, friction in the joints, noise effects, etc. The proposed method guarantees the stability of overall system and eliminates tracking error. Control algorithm is highly adaptable and is able to resist the noise impact on the system. The global asymptotic stability of the whole system is proved by the Lyapunov stability theory. The simulation results with uncertainties up to 20% in the case of fixed target, moving target in a straight line or circular arc, show that the tracking error converges on zero. These results are consistent with the theoretical.

## REFERENCES

- [1] G. Antonelli, "Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems," in *IEEE International Conference on Robotics and Automation, 2008. ICRA 2008*. IEEE, 2008, pp. 1993–1998.
- [2] W. Budiharto, A. Santoso, D. Purwanto, and A. Jazidie, "A navigation system for service robot using stereo vision and kalman filtering," in *11th International Conference on Control, Automation and Systems (ICCAS), 2011*. IEEE, 2011, pp. 1771–1776.
- [3] P. T. Cat and N. T. Minh, "Robust neural control of robot-camera visual tracking," in *IEEE International Conference on Control and Automation, 2009. ICCA 2009*. IEEE, 2009, pp. 1825–1830.
- [4] N. Cotter, "The stone-weierstrass theorem and its application to neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 4, pp. 290–295, 1990.
- [5] S.-T. Kao, C.-Y. Yen, and M.-T. Ho, "Design and implementation of a stereo vision-guided omnidirectional mobile robot for real-time object tracking," in *Proceedings of SICE Annual Conference 2010*. IEEE, 2010, pp. 2968–2974.
- [6] R. Kelly, R. Carelli, O. Nasisi, B. Kuchen, and F. Reyes, "Stable visual servoing of camera-in-hand robotic systems," *IEEE/ASME Transactions on Mechatronics*, vol. 5, no. 1, pp. 39–48, 2000.
- [7] T.-I. Kim, W. Bahn, C. Lee, T.-j. Lee, M. M. Shaikh, and K.-s. Kim, "Vision system for mobile robots for tracking moving targets, based on robot motion and stereo vision information," in *International Symposium on System Integration (SII), 2011 IEEE/SICE*. IEEE, 2011, pp. 634–639.

- [8] H. Li, M. Jin, and L. Zou, “A new binocular stereo visual servoing model,” in *Computational Intelligence and Industrial Application, 2008. PACIIA'08. Pacific-Asia Workshop on*, vol. 1. IEEE, 2008, pp. 461–465.
- [9] L. Sciavicco and B. Siciliano, *Modelling and control of robot manipulators*. McGraw Hill Inc., 1996.
- [10] M. Shibata, H. Eto, and M. Ito, “Image-based visual tracking to fast moving target for active binocular robot,” in *IECON 2010-36th Annual Conference on IEEE Industrial Electronics Society*. IEEE, 2010, pp. 2727–2732.
- [11] —, “Visual tracking control for stereo vision robot with high gain controller and high speed cameras,” in *1st International Symposium on Access Spaces (ISAS), 2011*. IEEE, 2011, pp. 288–293.
- [12] H. Ukida, Y. Terama, and H. Ohnishi, “Object tracking system by adaptive pan-tilt-zoom cameras and arm robot,” in *Proceedings of SICE Annual Conference (SICE), 2012*. IEEE, 2012, pp. 1920–1925.
- [13] B. Yang, H. Li, Z. Kang, and H. Jiang, “Hamiltonian-based binocular visual servoing of camera-in-hand robotic systems,” in *Proceedings of International Conference on Modelling, Identification & Control (ICMIC), 2012*. IEEE, 2012, pp. 388–393.

*Received on October 28 - 2014*

*Revised on June 11 - 2015*