

# Robust Visual Tracking Control of Pan-tilt-Stereo Camera System

Chung Le Van<sup>1</sup>

<sup>1</sup> Faculty of Automation Technology - Thai Nguyen University of Information and Communication Technology,  
P 211, C1, Z115, Quyet Thang (Thai Nguyen);  
Email: lvchung@ictu.edu.vn

## ABSTRACT

*In this paper, a new virtual 3D stereo camera model is presented to get a full rank of the image Jacobian matrix ( $J_{img}$ ) for an eye-in-hand stereo camera system. Some problems of the classical image Jacobian are local minima and singularities in image space solving by  $J_{img}$ . Furthermore, a robust visual control method is also proposed for a pan-tilt platform with stereo camera-in-hand using a neural network to compensate the effect of uncertainties in the dynamics of the overall system. An online learning neural network is used in the closed loop control system to compensate the uncertainties of both the Jacobian matrix and the dynamics of pan-tilt robot. The asymptotic stability of the overall visual control system is proved by Lyapunov stability method. The performance of the proposed control method is demonstrated by simulations in Matlab.*

**Keywords:** Visual servoing, pan-tilt robot, neural network, dynamic, tracking target.

**Mathematics Subject Classification:** 65D19, 70Q05, 70E60

**Computing Classification System:** I.2.9, I.4, I.6

## 1. INTRODUCTION

Visual control systems usually use one camera (Cat and Minh, 2009), (Kelly et al., 2000) or two cameras (Bo et al., 2012), (Li et al., 2009), (Tae-Il et al., 2011), (Widodo et al., 2011) to track moving targets. In such system, the image Jacobian matrices ( $J_{img}$ ) (Bo et al., 2012), (Oscar and Ricardo, 2003), (Shibata et al., 2010), (Shibata et al., 2011) play an important role in visual tracking tasks. However, they are not full-rank resulting in some problems as local minima and singularities in image space. Recently, some authors have developed a camera model to obtain a full-rank of the Jacobian matrix in a fixed stereo camera system (Cai et al., 2013). This paper proposes a 3D camera model to get a full rank of the image Jacobian matrix in an eye-in-hand camera.

Other problems are finding the appropriate control method and control algorithms. Numbers of authors use kinematic methods (Antonelli, 2008), (Siciliano et al., 2009), (Tae-Il et al., 2011) while the others use dynamic method (Cai et al., 2013), (Sho-Tsung et al., 2010). In recent years, there were many types of control algorithms in visual servoing systems to track moving targets including a classic control method as a PID controller (Li et al., 2009), a modern control method as sliding mode control (Cai et al., 2013), adaptive method (Oscar and Ricardo, 2003), (Ukida et al., 2012), combined with neural network (Cat and Minh, 2009), (Hashimoto et al., 1992), (Yang et al., 2017, Vol 13), (Yang et al., 2017, Vol 47) and fuzzy controller (Qiu et al., 2017). Nearly, some author integrated with optimization part of parameters or model for better results (Azar et al., 2016), (Vaščák, 2012). This

paper deals with the robust control problem of a pan-tilt-stereo camera system affected by uncertainties in both the Jacobian matrix and the dynamics of pan tilt platform. The RBF neural network with online learning algorithm is used for rapidly respond.

The Lyapunov stability theory as well as Barbalat's lemma are most frequently used to prove the asymptotic stability of the overall closed loop visual tracking system (Cai et al., 2013), (Li et al., 2009). The Hamiltonian method in checking asymptotic stability can be seen in (Bo et al., 2012).

In the previous paper (Chung and Cat, 2015), image Jacobian is not full rank and exits a singularity when take pseudo inverse. That is the cause of the tracking of camera cannot pass through the  $-\pi$  point when moving target follow a circle. Other hand, the controller that proposed in the paper (Chung and Cat, 2015) is only used for kinematic system and not paying attention to the speed error, uncertain in dynamic models also other effect of noise. Also, the image errors in section 4 of paper (Chung and Cat, 2015) are not really converges to zero. In this paper, a visual control scheme to control a pan-tilt-stereo camera system to track moving targets is proposed with some new ideas. Firstly, a 3D stereo camera model having a full-rank of image Jacobian for an eye-in-hand problem of a pan-tilt-stereo camera system is built. Secondly, the dynamic of the system is built. Thirdly, a robust controller with an on-line learning neural-network to compensate the effects of noise as well as uncertainties in the kinematics and dynamics of the overall system is proposed.

The paper is organized as follows. In section 2, a 3D virtual stereo camera model for eye-in-hand stereo camera system with pan-tilt robot (Fig.1) is determined. In section 3, a robust control algorithm with neural network is represented and the asymptotic stability of the overall system is proved by Lyapunov stability method. Section 4 shows simulation results on Matlab-Simulink to demonstrate the performance of the proposed control algorithm. Finally, the conclusion is summarized in the section 5.

## 2. 3D VISUAL MODEL FOR EYE-IN-HAND STEREO CAMERA SYSTEM

The realization of this work supposes the availability of a great number of repetitions of samples responding to the same known theoretical model. In practice, as the theoretical model is unknown, we use the Monte-Carlo method based on the generation of the data by computer according to a fixed theoretical model.

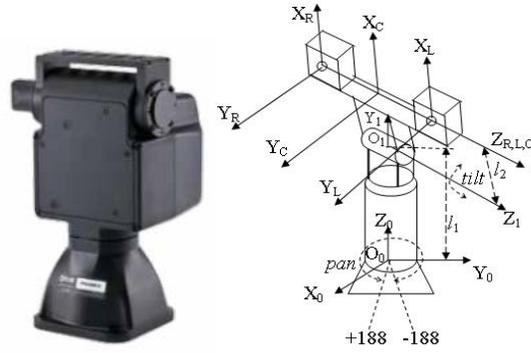
### 2.1. 3D virtual stereo camera model systems

Figure 2 shows the coordinate frames used to construct 3D visual camera model. Left and right camera coordinate frames are  $O_L X_L Y_L Z_L$ ,  $O_R X_R Y_R Z_R$  with the origin located at the focal point of respective camera. The origin of camera frame  $O_c X_c Y_c Z_c$  is located at the midpoint of origin of two cameras.

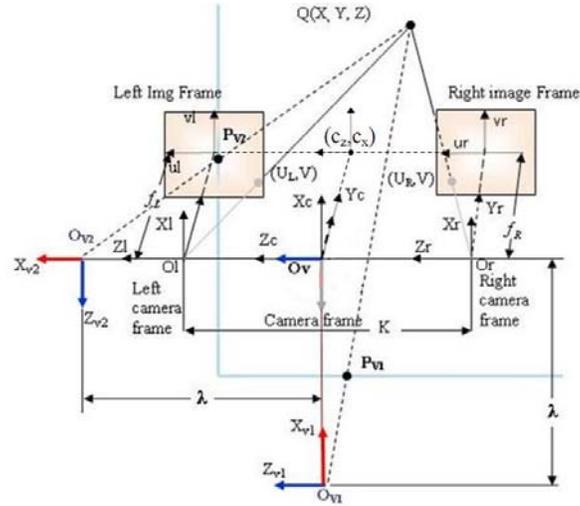
**Assumption 1:** The intrinsic parameters of camera as focal length  $f$ , number of pixels, etc. are the same, placed at the same height and the optical axis of two cameras is parallel together. Pan angle is  $\theta_1$ , its rotation around the axis  $Z_0$  in the original coordinate of platform pan/tilt, 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 platform pan/tilt.

The feature point coordinates of the target are captured from the left camera's image named as  $(U_L, V_L)$  and the right camera  $(U_R, V_R)$  on two axes  $(U, V)$ . Following the above assumption, the coordinates of feature point on two cameras are the same on  $V$  axis or  $V_R = V_L = V$ . From Fig.2, the coordinates of feature point on the left image frame  $(U_L, V_L)$  and the right image frame  $(U_R, V_R)$  are transformed to  $(Z, Y)$  and  $(X, Y)$  planes as shown in Fig.3. Now, a 3D visual space is built according to the following steps:

*First step*, from geometrical relations between the target and that feature images point in Fig.3, the coordinates of the target point  $\mathbf{x}_c = [x \ y \ z]^T$  in the camera coordinate frame  $O_c X_c Y_c Z_c$  are calculated as follows eq.(1) (Chung and Cat, 2015):



**Figure 1.** Coordinates of Pan/Tilt camera system



**Figure 2.** 3D visual stereo camera model

$$\mathbf{x}_c = [x \quad y \quad z]^T = \frac{1}{U_R - U_L} \begin{bmatrix} K.V_L \\ K.f \\ \frac{K}{2}(U_R + U_L) \end{bmatrix}, \quad (1)$$

where  $K$  is distance between two cameras optical  $O_L$  and  $O_R$ .  $f$  is focal length of camera.

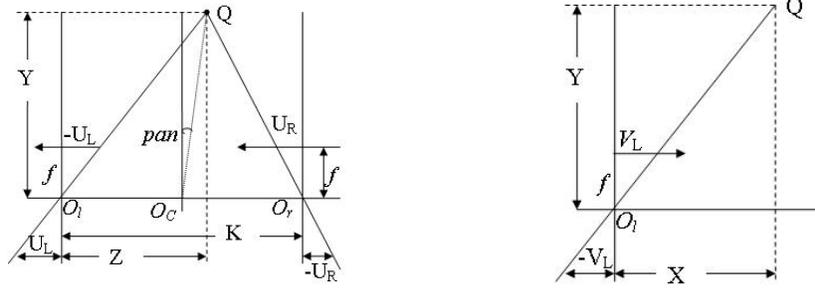
*Second step*, a reference coordinate frame  $O_v X_v Y_v Z_v$  with the origin located at the same position as  $O_c X_c Y_c Z_c$  is defined. In order to transform  $O_c$  to  $O_v$ , the rotation matrix  ${}^c R_v$  (Fig.2) is used. The projection of  $\mathbf{x}_c = [x \quad y \quad z]^T$  in  $O_c X_c Y_c Z_c$  is defined in  $O_v X_v Y_v Z_v$  as  $\mathbf{x}_v = [z_v \quad x_v \quad y_v]^T$  (Cai et al., 2013):

$$\mathbf{x}_v = {}^v R_c \mathbf{x}_c, \quad (2)$$

where

$${}^v R_c = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & -\sin \psi \\ 0 & \sin \psi & \cos \psi \end{bmatrix}; \psi = -90^\circ. \quad (3)$$

The purpose of coordinate frame  $O_v$  is to simplify the virtual camera model in a specific orientation of  $y$  and  $z$  axes when define two virtual camera's frame on the next step.



**Figure 3.** Coordinates of feature point on Z, Y and X, Y axis

*Third step*, the reference coordinate frame  $O_v X_v Y_v Z_v$  is used to define two virtual camera's frame  $O_{v1} X_{v1} Y_{v1} Z_{v1}$ ,  $O_{v2} X_{v2} Y_{v2} Z_{v2}$  associated with stereo cameras. Their location on  $X_v$  and  $Z_v$  axes are far away from  $O_v$  the distances  $\lambda$ .

The coordinates of the origin  $O_{v1}$  and  $O_{v2}$  in  $O_v X_v Y_v Z_v$  respectively are

$$[z_v; x_v - \lambda]^T; [z_v + \lambda; x_v]^T. \quad (4)$$

*Last step*, the virtual camera model is combined with 3D visual camera model to construct a 3D virtual Cartesian space having feature point vector denoted as:  $\mathbf{x}_s = [z_{v1} \ z_{v2} \ x_{v1}]^T$ .

The pinhole camera model (Shibata et al., 2011) is used to calculate the relationship between the coordinates of the target in  $O_v X_v Y_v Z_v$  and its image coordinates in the virtual cameras. Its image coordinates in  $O_{v1} X_{v1} Y_{v1} Z_{v1}$  are  $V_1(z_{v1}, x_{v1})$  and in  $O_{v2} X_{v2} Y_{v2} Z_{v2}$  are  $V_2(z_{v2}, x_{v2})$ :

$$z_{v1} = f_v \frac{z_v}{\lambda - x_v} + c_z, \quad (5)$$

$$z_{v2} = \frac{f_v}{\lambda + z_v} (x_v - \lambda) + c_z, \quad (6)$$

$$x_{v2} = \frac{f_v}{\lambda + z_v} y_v + c_x. \quad (7)$$

$(c_z, c_x)$  are the coordinates of the principal point in the virtual image frame seen in the frame  $O_v X_v Y_v Z_v$ .  $f_v$  is the focal length of virtual camera.  $l$  is the distance from the frame  $O_v X_v Y_v Z_v$  to the coordinate of virtual camera origin  $O_{v1}$ ,  $O_{v2}$  respectively with the axis  $z$ ,  $x$ . Rewriting (5), (6), (7) in another form yields (8)

$$\mathbf{x}_s = \begin{bmatrix} z_{v1} \\ z_{v2} \\ x_{v2} \end{bmatrix} = f_v \begin{bmatrix} \frac{z_v}{\lambda - x_v} \\ \frac{x_v - \lambda}{z_v + \lambda} \\ \frac{y_v}{z_v + \lambda} \end{bmatrix} + \begin{bmatrix} c_z \\ c_z \\ c_x \end{bmatrix} \quad (8)$$

where  $\mathbf{x}_s$  is the coordinates of the target feature point in virtual space.

If the target or pan/tilt joints moves, the features images point in virtual space also move. Assuming the geometric relationship between the virtual and real camera system is still guarantees under the above formulas when the pan/tilt robot turns. Then, taking the derivative (8) with the time results in (9) that describes the speed relations between the target's image on two virtual cameras and the frame  $O_v X_v Y_v Z_v$ :

$$\dot{\mathbf{x}}_s = f_v \mathbf{J}_{vimg} \dot{\mathbf{x}}_v, \quad (9)$$

where  $\mathbf{J}_{vimg}$  is the visual Jacobian matrix:

$$\mathbf{J}_{vimg} = \begin{bmatrix} \frac{1}{\lambda - x_v} & \frac{z_v}{(\lambda - x_v)^2} & 0 \\ -\frac{x_v - \lambda}{(z_v + \lambda)^2} & \frac{1}{z_v + \lambda} & 0 \\ -\frac{y_v}{(z_v + \lambda)^2} & 0 & \frac{1}{z_v + \lambda} \end{bmatrix} \quad (10)$$

$\mathbf{J}_{vimg}$  describes the speed relations between the image in two virtual cameras and the target in the reference frame  $O_v X_v Y_v Z_v$ . The coordinates of the target in the base frame  $O_0 X_0 Y_0 Z_0$  of the pan-tilt robot can be computed as:

$$\mathbf{x}_0 = {}^0 R_C \mathbf{x}_C. \quad (11)$$

Then  $\mathbf{x}_v = {}^v R_0 {}^0 R_C \mathbf{x}_0$ , where  $\mathbf{x}_0$  is the coordinate of target in the origin of pan-tilt coordinate  $O_0$ . The equation (9) can be rewritten as:

$$\dot{\mathbf{x}}_s = f_v \mathbf{J}_{vimg} {}^C R_v {}^0 R_C \dot{\mathbf{x}}_v = \mathbf{J}_{img} \dot{\mathbf{x}}_0 + f_v \mathbf{J}_{vimg} {}^v \dot{R}_C \mathbf{x}_0, \quad (12)$$

where

$${}^0 R_C = \begin{bmatrix} C_1 C_2 & -C_1 S_2 & S_1 \\ S_1 C_2 & -S_1 S_2 & -C_1 \\ S_2 & C_2 & 0 \end{bmatrix}; \quad (13)$$

${}^0 R_C$  is the homogeneous transformation matrix which transforms coordinates from  $O_C X_C Y_C Z_C$  to  $O_0 X_0 Y_0 Z_0$ .  $C_1$  is  $\cos\theta_1$ ;  $C_2$  is  $\cos\theta_2$ ;  $S_1$  is  $\sin\theta_1$ ;  $S_2$  is  $\sin\theta_2$ ;  $\mathbf{J}_{img} = f_v \mathbf{J}_{vimg} {}^C R_v {}^0 R_C$ .

The differential kinematic of pan-tilt pedestal describes the velocity relations between the joint velocity  $\dot{\mathbf{q}} = [\dot{q}_1, \dot{q}_2]^T$  and the target velocity  $\dot{\mathbf{x}}_0$ :

$$\mathbf{J}_{img} = f_v \mathbf{J}_{vimg} {}^C R_v {}^0 R_C \quad (14)$$

where  $\mathbf{J}_{robot}(\mathbf{q})$  is (3x2) and is the Jacobian matrix of the pan/tilt robot see in camera coordinate. It can be determined as (Siciliano et al., 2009):

$$\mathbf{J}_{robot}(\mathbf{q}) = \begin{bmatrix} -l_2 S_1 S_2 & l_2 C_1 C_2 \\ l_2 C_1 S_2 & l_2 C_1 S_2 \\ 0 & -l_2 S_2 \end{bmatrix} \quad (15)$$

Substituting the equation (14) into (11) yields:

$$\begin{aligned} \dot{\mathbf{x}}_s &= \mathbf{J}_{img} \dot{\mathbf{x}}_0 + f_v \mathbf{J}_{vimg} {}^v \dot{R}_C \mathbf{x}_0 = \mathbf{J}_{img} \mathbf{J}_{robot}(\mathbf{q}) \dot{\mathbf{q}} + f_v \mathbf{J}_{vimg} {}^v \dot{R}_C \mathbf{x}_0, \\ &= \mathbf{J} \dot{\mathbf{q}} + f_v \mathbf{J}_{vimg} {}^v \dot{R}_C \mathbf{x}_0 \end{aligned} \quad (16)$$

where  $\mathbf{J} = \mathbf{J}_{img} \mathbf{J}_{robot}(\mathbf{q})$  is the visual Jacobian (3x2) matrix of the pan-tilt camera in the 3D virtual Cartesian space.

## 2.2. Avoid singularity

The homogeneous transformation matrices  ${}^0 R_C$  and  ${}^C R_v$  are non singular. Therefore, the singularity of  $\mathbf{J}_{img}$  depends on only  $\mathbf{J}_{vimg}$ .  $\mathbf{J}_{vimg}$  will be singular only when  $\lambda = x_v$  or  $z_v = \lambda$ . Then, the singularity of  $\mathbf{J}_{vimg}$  that can be avoided by choosing the parameter  $\lambda$  such as  $\lambda > \max(x_v, z_v)$ . Similarly, the

singularity of  $\mathbf{J}_{robot}$  can be avoided by defining the limit of robot workspace. Therefore, the singularities of the system Jacobian matrix  $\mathbf{J}$  only depend on those of the robot Jacobian matrix.

### 2.3. Stereo visual servoing problem with uncertain parameters

Assuming that the parameters of the above visual control system are not known exactly, now the Jacobian matrix is described as:

$$\mathbf{J} = \hat{\mathbf{J}} + \Delta\mathbf{J}, \quad (17)$$

where  $\hat{\mathbf{J}}$ ,  $\Delta\mathbf{J}$  are the known and unknown part of the Jacobian matrix.

Substituting Eq. (17) into Eq. (14) yields:

$$\dot{\mathbf{x}}_s = (\hat{\mathbf{J}} + \Delta\mathbf{J})\dot{\mathbf{q}} + f_v \mathbf{J}_{vimg} {}^v \dot{R}_C \mathbf{x}_0. \quad (18)$$

### 2.4. Dynamics of robot manipulator with uncertainties.

The dynamics of a serial  $n$ -link rigid robot with friction, and uncertainty can be written as follows:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{d}(t) = \boldsymbol{\tau} \quad (19)$$

where  $\mathbf{M}(\mathbf{q})$  is the  $(n \times n)$  symmetric positive definite inertia matrix.  $\mathbf{q}$  is the  $(n \times 1)$  vector of robot joint rotation.  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$  is the  $(n \times n)$  vector of centripetal and Coriolis effects.  $\mathbf{g}(\mathbf{q})$  is the vector of gravitational torque and  $\boldsymbol{\tau}$  is the torque vector applied for robot joints. In this system, pan/tilt robot has two joints,  $n = 2$ .  $\mathbf{d}(t)$  is the effect of all uncertainties in the robot dynamics

Assuming that,  $\mathbf{d}(t)$  is continuous and bounded function, so:

$$\|\mathbf{d}(t)\| \leq d_{\max}. \quad (21)$$

From the equation (19) it derives (Cat and Minh, 2009):

$$\ddot{\mathbf{q}} = \mathbf{M}^{-1}(\mathbf{q})[\boldsymbol{\tau} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q}) - \mathbf{d}(t)]. \quad (22)$$

The tracking image feature error of the system is defined as  $\boldsymbol{\varepsilon} = \mathbf{x}_s - \mathbf{x}_{sd}$ ; where  $\mathbf{x}_s \in R^3$  is a vector of image feature points,  $\mathbf{x}_{sd} \in R^3$  is a desired constant vector of image feature points of the target in the 3D virtual Cartesian space. The variable vector  $\boldsymbol{\varepsilon}$  describes the image feature error in the 3D virtual Cartesian space.

$$\boldsymbol{\varepsilon} = \mathbf{x}_s - \mathbf{x}_{sd}. \quad (23)$$

Taking the derivative of  $\boldsymbol{\varepsilon}$ , the first and the second time, yields:

$$\dot{\boldsymbol{\varepsilon}} = \dot{\mathbf{x}}_s = \hat{\mathbf{J}}\dot{\mathbf{q}} + \Delta\mathbf{J}\dot{\mathbf{q}}, \quad (24)$$

$$\ddot{\boldsymbol{\varepsilon}} = \ddot{\mathbf{x}}_s = \hat{\mathbf{J}}\ddot{\mathbf{q}} + \dot{\hat{\mathbf{J}}}\dot{\mathbf{q}} + \Delta\mathbf{J}\ddot{\mathbf{q}} + \Delta\dot{\mathbf{J}}\dot{\mathbf{q}}. \quad (25)$$

Substituting the Eq. (22) into Eq. (25) and after arranging, results in:

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\hat{\mathbf{J}}^+ \ddot{\boldsymbol{\varepsilon}} - \mathbf{M}(\mathbf{q})\hat{\mathbf{J}}^+ \dot{\hat{\mathbf{J}}}\dot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{f}_u, \quad (26)$$

where  $\hat{\mathbf{J}}^+ = (\hat{\mathbf{J}}^T \hat{\mathbf{J}})^{-1} \hat{\mathbf{J}}^T$  is the pseudo inverse matrix of  $\hat{\mathbf{J}}$ .  $\mathbf{f}_u$  is the uncertainty component:

$$\mathbf{f}_u = \mathbf{d}(t) + \Delta\mathbf{J}\ddot{\mathbf{q}} - \hat{\mathbf{M}}(\mathbf{q})\hat{\mathbf{J}}^+ \dot{\mathbf{J}}\dot{\mathbf{q}}. \quad (27)$$

Denoting:

$$\begin{aligned}\mathbf{A} &= \mathbf{M}(\mathbf{q})\hat{\mathbf{J}}^+ \\ \mathbf{b} &= -\mathbf{A}\hat{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q})\end{aligned}\quad (28)$$

And rewriting the Eq. (26), yields:

$$\boldsymbol{\tau} = \mathbf{A}\ddot{\boldsymbol{\epsilon}} + \mathbf{b} + \mathbf{f}_u. \quad (29)$$

The dynamic control problem of stereo visual servoing is finding the control law  $\boldsymbol{\tau}$  in the (29) to stabilize asymptotically the closed-loop system, so the image features error  $\boldsymbol{\epsilon} \rightarrow \mathbf{0}$ .

### 3. ROBUST NEURAL CONTROL OF STEREO CAMERA SYSTEM WITH PAN TILT ROBOT

#### 3.1. Construction of robust controller

The task is to find the joint torque  $\boldsymbol{\tau}$  to make the image feature error vector  $\boldsymbol{\epsilon}$  reaching to  $\mathbf{0}$ . The control law is chosen as follows:

$$\boldsymbol{\tau} = \mathbf{A}(-\mathbf{K}_D\dot{\boldsymbol{\epsilon}} - \mathbf{K}_P\boldsymbol{\epsilon}) + \mathbf{b} + \boldsymbol{\tau}_{NR} = \boldsymbol{\tau}_0 + \boldsymbol{\tau}_1, \quad (30)$$

$$\boldsymbol{\tau}_0 = \mathbf{A}(-\mathbf{K}_D\dot{\boldsymbol{\epsilon}} - \mathbf{K}_P\boldsymbol{\epsilon}) + \mathbf{b}, \quad (31)$$

where  $\mathbf{K}_D$ ,  $\mathbf{K}_P$  are symmetric positive definite matrices.  $\boldsymbol{\tau}_{NR}$  is a control component used to compensate the effect of uncertain parameters. It will be determined later.

Substitute equation (30) and (31) into (29), the error dynamics can be rewritten as follows:

$$\ddot{\boldsymbol{\epsilon}} + \mathbf{K}_D\dot{\boldsymbol{\epsilon}} + \mathbf{K}_P\boldsymbol{\epsilon} = \mathbf{A}^+(\boldsymbol{\tau}_{NR} - \mathbf{f}_u). \quad (32a)$$

Denoting:  $\boldsymbol{\tau}_1 = \mathbf{A}^+\boldsymbol{\tau}_{NR}$  and  $\mathbf{f}_1 = \mathbf{A}^+\mathbf{f}_u$  where  $\mathbf{A}^+ = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$  is the pseudo inverse matrix of  $\mathbf{A}$ . Rewriting the equation (32a), yields:

$$\ddot{\boldsymbol{\epsilon}} + \mathbf{K}_D\dot{\boldsymbol{\epsilon}} + \mathbf{K}_P\boldsymbol{\epsilon} = \boldsymbol{\tau}_1 - \mathbf{f}_u. \quad (32b)$$

#### 3.2 Layer construction of RBF neural network

Now, it is to construct a neural network to approximate  $\mathbf{f}_1$  and find the component  $\boldsymbol{\tau}_1$  so the equation (32b) is asymptotically stable.  $\mathbf{f}_1$  can be approximated by a neural network with suitable learning law and the control vector  $\boldsymbol{\tau}_{NR}$  such as system (32b) asymptotically stable can be determined. The structure of choosing artificial neural network is a Radial Basis Function (RBFNN) network. It has 3 layers.

##### Input layer

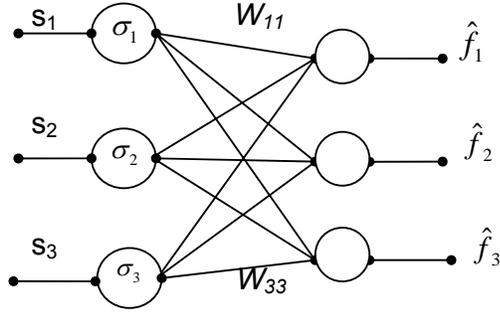
Input layer of the neural network includes three components of the image error  $\boldsymbol{\epsilon}$ . In the neural network the input vector is  $\mathbf{s} = [s_1 \ s_2 \ s_3]^T$  and chosen as follows:

$$\mathbf{s} = \dot{\boldsymbol{\epsilon}} + \mathbf{H}\boldsymbol{\epsilon}, \quad (33)$$

where  $\mathbf{H}$  is a symmetric positive-definite matrix. If  $\mathbf{s} \rightarrow \mathbf{0}$  as  $t \rightarrow \infty$ , will  $\boldsymbol{\epsilon} \rightarrow \mathbf{0}$  as  $t \rightarrow \infty$ .

##### Hidden layer computation

The hidden layer consists of neurons with output function calculated by Gaussian form (Murata et al., 1994):



**Figure 4.** RBF network approximating function  $f$

$$\sigma_j = \exp\left(-\frac{(\varepsilon_j - c_j)^2}{\lambda_j^2}\right) ; j = 1, 2, 3 . \quad (34)$$

where  $\sigma_j$  is the activation function for hidden unit  $j$ .  $c_j, \lambda_j$  are the center and the width of hidden unit of the  $j$  radial basis function.  $(\varepsilon_j - c_j)$  is the Euclidian distance from center. The value of  $\sigma_j$  only acceptable when  $(\varepsilon_j - c_j) < \lambda_j^2$ .

#### Output layer

The output values of the network are approximate function  $f_1$  as shown in Fig.4. that includes 3 linear neurons. According to the Stone – Weierstrass theorem (Cotter et al., 1990), the structure of RBF network above can approximate the unknown function  $f_1(\mathbf{s})$  represented as below:

$$f_1(\mathbf{s}) = \hat{f}(\mathbf{s}) + \boldsymbol{\beta} , \quad (34a)$$

$$\hat{f}(\mathbf{s}) = \mathbf{W}\boldsymbol{\sigma} \quad (34b)$$

where  $\boldsymbol{\beta}$  is bounded approximation errors.

$$\|\boldsymbol{\beta}\| \leq \beta_0 , \quad (35)$$

$\mathbf{W}$  is the weight matrix of the neural network, updated on-line from the real-time measured value of error  $\varepsilon$  following equation (38).

### 3.3 Optimization parameters

In the papers (Azar et al., 2016), (Mostafa et al., 2016), (Precup et al., 2015), (Vaščák, 2012) the authors used some algorithms to optimize the parameters. SOMA algorithm is used for optimization parameters in fuzzy cognitive maps (Vaščák, 2012). In paper (Mostafa et al., 2016) new algorithm MOsDE-Im used for multi-objective optimization. Two nature-inspired optimization algorithms are SA and PSO (Precup et al., 2015) to optimal Takagi-Sugeno-Kang fuzzy models. In this paper, some parameters in the controller as  $\mathbf{G}, \mathbf{H}, c_j, \lambda_j$  are optional but we can optimize them.

The Genetic Algorithms (GA) is used in optimizing parameters as radial basis function center  $c_j$  and base width parameter function  $\lambda_j$  followed steps (Yang et al., 2007):

*Step 1:* randomize initial parameters. The initial of  $c_j, \lambda_j$  shown in the equation (53) in simulation section.

*Step 2:* Train the RBFNN model and calculate the fitness values.

*Step 3:* Check the satisfy of the image error vector  $\mathbf{s} = [s_1 \ s_2 \ s_3]^T$ . if the image error are not in allowed range go to step 4.

*Step 4:* Crossover, mutation and reproduction new  $c_j, \lambda_j$  parameters then go to step 2.

The  $\mathbf{G}, \mathbf{H}$  parameters in this paper are determined by experimental methods. These parameters are used to define  $\mathbf{K}_D, \mathbf{K}_P$  parameters of PD controller.

**Theorem 1:** The image error dynamics (32b) of the uncertain pan-tilt – Stereo camera tracking system (18), (19) will be asymptotically stable with the error  $\boldsymbol{\varepsilon} \rightarrow \mathbf{0}$  if the control torque is chosen by following (36), (37) and online learning rules (38):

$$\boldsymbol{\tau} = \mathbf{A}(-\mathbf{K}_D \dot{\boldsymbol{\varepsilon}} - \mathbf{K}_P \boldsymbol{\varepsilon}) + \mathbf{b} + \boldsymbol{\tau}_{NR}, \quad (36)$$

$$\boldsymbol{\tau}_{NR} = \mathbf{A} \left[ (\eta + 1) \mathbf{W} \boldsymbol{\sigma} - \delta \frac{\boldsymbol{\varepsilon}}{\|\boldsymbol{\varepsilon}\|} \right], \quad (37)$$

$$\dot{\mathbf{W}} = -\eta \mathbf{s} \boldsymbol{\sigma}^T, \quad (38)$$

where  $\mathbf{K}_D = \mathbf{G} + \mathbf{H}$ ,  $\mathbf{K}_P = \mathbf{G}\mathbf{H}$  and  $\mathbf{H} = \mathbf{H}^T > \mathbf{0}$ ,  $\mathbf{G} = \mathbf{G}^T > \mathbf{0}$ .  $\mathbf{K}_P$  and  $\mathbf{K}_D$  are two parameters of PD controller.

**Proof:**

We choose the candidate Lyapunov function as follows:

$$V(\mathbf{s}, \mathbf{W}) = \frac{1}{2} \mathbf{s}^T \mathbf{s} + \frac{1}{2} \sum_{i=1}^3 \mathbf{w}_i^T \mathbf{w}_i. \quad (39)$$

We have  $V > 0$  when  $\mathbf{s} \neq \mathbf{0}, \boldsymbol{\varepsilon} \neq \mathbf{0}$ ,  $V = 0$  if and only if  $\mathbf{s} = \mathbf{0}, \boldsymbol{\varepsilon} = \mathbf{0}$ .  $V \rightarrow \infty$  when  $\mathbf{s}, \mathbf{w}_i \rightarrow \infty$ . Taking the derivative of  $V$  along time, yields:

$$\dot{V} = \frac{1}{2} \mathbf{s}^T \dot{\mathbf{s}} + \frac{1}{2} \sum_{i=1}^3 \mathbf{w}_i^T \dot{\mathbf{w}}_i. \quad (40)$$

From the equation (33) it derives:

$$\dot{\mathbf{s}} = \ddot{\boldsymbol{\varepsilon}} + \mathbf{H} \dot{\boldsymbol{\varepsilon}}. \quad (41)$$

Subtracted  $\ddot{\boldsymbol{\varepsilon}}$  from the equation (32b) and then substituting into the equation (41), results in:

$$\dot{\mathbf{s}} = \boldsymbol{\tau}_1 - \mathbf{f}_1 - (\mathbf{K}_D - \mathbf{H}) \dot{\boldsymbol{\varepsilon}} - \mathbf{K}_P \boldsymbol{\varepsilon}. \quad (42a)$$

If choosing the optional parameters  $\mathbf{K}_D = \mathbf{G} + \mathbf{H}$  and  $\mathbf{K}_P = \mathbf{G}\mathbf{H}$  where  $\mathbf{G}$  is a symmetric positive-definite matrix, then the Eq.42a can be rewritten as follows:

$$\dot{\mathbf{s}} = \boldsymbol{\tau}_1 - \mathbf{f}_1 - \mathbf{G}\mathbf{s}. \quad (42b)$$

Substituting equation (34a), (34b) and (42b) into (40), yields:

$$\dot{V} = \mathbf{s}^T (\boldsymbol{\tau}_1 - \mathbf{w}\boldsymbol{\sigma} - \boldsymbol{\beta} - \mathbf{G}\mathbf{s}) + \sum_{i=1}^3 \mathbf{w}_i^T \dot{\mathbf{w}}_i. \quad (43)$$

With on-line learning algorithm (38) results in:

$$\dot{\mathbf{w}}_i = -\eta \mathbf{s} \sigma_i; i = 1, 2, 3; \quad (44)$$

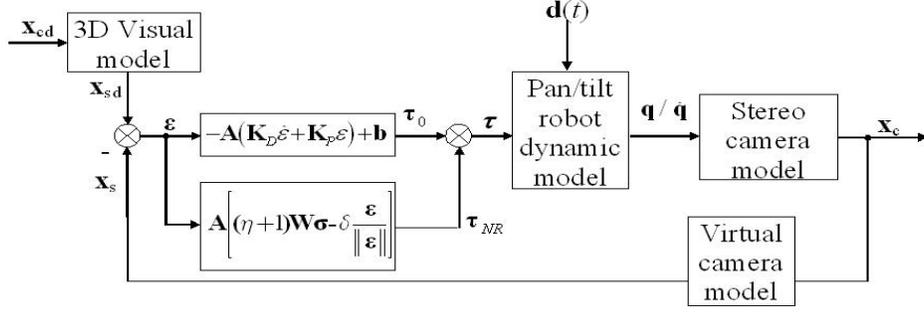
And it is possible to identify:

$$\sum_{i=1}^3 \mathbf{w}_i^T \dot{\mathbf{w}}_i = \sum_{i=1}^3 -\eta \mathbf{w}_i^T \mathbf{s} \sigma_i = -\eta \mathbf{s}^T \sum_{i=1}^3 \mathbf{w}_i \sigma_i = -\eta \mathbf{s}^T \mathbf{W}\boldsymbol{\sigma} \quad (45)$$

Substituting the equation (45) into (43), yields:

$$\dot{V} = -\mathbf{s}^T \mathbf{G}\mathbf{s} + \mathbf{s}^T [\boldsymbol{\tau}_1 - (\eta + 1) \mathbf{W}\boldsymbol{\sigma} - \boldsymbol{\beta}]. \quad (46)$$

Selecting  $\boldsymbol{\tau}_1 = (\eta + 1) \mathbf{W}\boldsymbol{\sigma} - \delta \frac{\mathbf{s}}{\|\mathbf{s}\|}$ ;  $\delta > 0$  and substituting  $\boldsymbol{\tau}_1$  into (46) results in the following:



**Figure 5.** Structure of proposed visual tracking system

$$\begin{aligned} \dot{V} &= -\mathbf{s}^T \mathbf{G} \mathbf{s} + \mathbf{s}^T \left( \delta \frac{\mathbf{s}}{\|\mathbf{s}\|} + \boldsymbol{\beta} \right) = -\mathbf{s}^T \mathbf{G} \mathbf{s} - \delta \|\mathbf{s}\| - \mathbf{s}^T \boldsymbol{\beta} \\ \dot{V} &\leq -\mathbf{s}^T \mathbf{G} \mathbf{s} - \delta \|\mathbf{s}\| + \|\mathbf{s}\| \|\boldsymbol{\beta}\| \leq -\mathbf{s}^T \mathbf{G} \mathbf{s} - \delta \|\mathbf{s}\| + \|\mathbf{s}\| \beta_0 \end{aligned} \quad (47)$$

If choosing  $\delta = \beta_0 + \mu$ ;  $\mu > 0$  it yields:

$$-\delta \|\mathbf{s}\| + \|\mathbf{s}\| \beta_0 \leq -\mu \|\mathbf{s}\| \quad (48)$$

Substituting (48) into (47) results in:

$$\dot{V} \leq -\mathbf{s}^T \mathbf{G} \mathbf{s} - \mu \|\mathbf{s}\| \leq 0 \quad (49)$$

$\dot{V}$  is negative-semidefinite function. This implies that  $V(t) \leq V(0)$ , and therefore, that  $\mathbf{s}$  and  $\mathbf{W}$  are bounded. Because the dynamic system is non-autonomous, it is impossible to conclude the convergence of  $\mathbf{s}$  to 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:

$$\begin{aligned} \ddot{V} &= -2\mathbf{s}^T \mathbf{G} \dot{\mathbf{s}} - \dot{\mathbf{s}}^T \boldsymbol{\beta} - \delta \frac{\mathbf{s} \dot{\mathbf{s}}}{\|\mathbf{s}\|} = -\left( 2\mathbf{s}^T \mathbf{G} - \boldsymbol{\beta}^T + \delta \frac{\mathbf{s}}{\|\mathbf{s}\|} \right) \dot{\mathbf{s}} \\ &= -\left( 2\mathbf{s}^T \mathbf{G} - \boldsymbol{\beta}^T + \delta \frac{\mathbf{s}}{\|\mathbf{s}\|} \right) \left( \eta \mathbf{W} \boldsymbol{\sigma} - \delta \frac{\mathbf{s}}{\|\mathbf{s}\|} - \boldsymbol{\beta} - \mathbf{G} \mathbf{s} \right) \end{aligned} \quad (50)$$

We found that  $\ddot{V}$  is bounded because  $\mathbf{s}, \boldsymbol{\beta}$  and  $\mathbf{W}$  are bounded,  $\frac{\mathbf{s}}{\|\mathbf{s}\|}$  is the unit vector of  $\mathbf{s}$  and always bounded,  $\mathbf{G}$  is the positive-definite constant matrix and  $\eta, \delta > 0$ . Thus  $\dot{V}$  is uniformly continuous. According to Barbalat's lemma, we have  $\mathbf{s} \rightarrow \mathbf{0}$  as  $t \rightarrow \infty$  and it forces  $\boldsymbol{\varepsilon} \rightarrow \mathbf{0}$  as  $t \rightarrow \infty$ . Hence, the system represented in the eq. (32b) is asymptotically stable and the cameras will track moving targets with error converges to zero as the time approaches to the infinity. Theorem 1 as well as the global asymptotic stability of the overall stereo camera visual tracking system using neural network described in Fig. 5 has been proven.

#### 4. SIMULATION RESULTS

There are dynamics of the pan-tilt robot as eq. (19) with  $\mathbf{q} = [q_1, q_2]^T$  is the joint's angular vector of pan/tilt platform.  $\boldsymbol{\tau} = [\tau_1, \tau_2]^T$  Are torques impacted on the joints. The inertia matrix of the pan - tilt robot  $\mathbf{M}(\mathbf{q})$ , the vector of  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  and  $\mathbf{g}(\mathbf{q})$  are:

$$\mathbf{M}(\mathbf{q}) = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}; \quad \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} -m_{13} \dot{q}_2 & -m_{13} \dot{q}_1 \\ -m_{13} \dot{q}_1 & 0 \end{bmatrix}; \quad \mathbf{g}(\mathbf{q}) = \begin{bmatrix} 0 \\ g m_2 l_{c2} C_2 \end{bmatrix} \quad (51)$$

where:

$$\begin{aligned} m_{11} &= l_1 + l_2 S_2^2 + l_1 C_2^2 & m_{22} &= l_2; & C_2 &= \text{Cos}(q_2); S_2 = \text{Sin}(q_2) \\ m_{12} &= m_{21} = 0 & m_{13} &= (l_1 - l_2) S_2 C_2 \end{aligned} \quad (52)$$

The Inertia of robot links  $l_1, l_2$  are presented in table of system parameters below.

Choose the optional parameters:

$$\eta = 1.5; \delta = 0.2; \sigma_1 = \exp\left(-\frac{(s_1 - 2)^2}{9}\right); \sigma_2 = \exp\left(-\frac{(s_2 - 2)^2}{16}\right); \sigma_3 = \exp\left(-\frac{(s_3 - 2)^2}{25}\right) \quad (53)$$

$$\mathbf{G} = \text{diag}\{5, 2.5, 5\} \times 10^{-2}; \mathbf{H} = \text{diag}\{2.5, 5, 2.5\} \times 10^{-2}; \mathbf{K}_D = \mathbf{G} + \mathbf{H}; \mathbf{K}_P = \mathbf{G}\mathbf{H} \quad (54)$$

Assuming that only 80% value of the Jacobian matrix is just estimated:  $\hat{\mathbf{J}} = 0.8\mathbf{J}$

Notation  $\hat{\mathbf{J}} = 0.8 \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{bmatrix}$ , the pseudo inverse matrix is calculated by:

$$\hat{\mathbf{J}}^+ = (\hat{\mathbf{J}}^T \hat{\mathbf{J}})^{-1} \hat{\mathbf{J}}^T = 0.8 \begin{bmatrix} \frac{a_1 C - b_1 B}{M} & \frac{a_2 C - b_2 B}{M} & \frac{a_3 C - b_3 B}{M} \\ \frac{a_1 B - b_1 A}{M} & \frac{a_2 B - b_2 A}{M} & \frac{a_3 B - b_3 A}{M} \\ M & M & M \end{bmatrix} \quad (55)$$

Where:

$$\begin{aligned} A &= a_1^2 + a_2^2 + a_3^2; \quad C = b_1^2 + b_2^2 + b_3^2; \quad B = a_1 b_1 + a_2 b_2 + a_3 b_3 \\ M &= (a_1 b_2 - a_2 b_1)^2 + (a_1 b_3 - a_3 b_1)^2 + (a_2 b_3 - a_3 b_2)^2 \end{aligned} \quad (56)$$

Table: The parameters of the system

Parameter name	Notation	Value
Length of link 1	$l_1$	0.22 m
Length of link 2	$l_2$	0.1 m
Link 1 Center of gravity	$l_{c1}$	0.15 m
Link 2 Center of gravity	$l_{c2}$	0.07 m
Inertia of link 1	$I_1$	0.01 kg m <sup>2</sup>
Inertia of link 1	$I_2$	0.002 kg m <sup>2</sup>
Mass of link 1	$m_1$	4.5 kg
Mass of link 1	$m_2$	1.2 kg
Distance of 2 cameras	K	0.2 m
Focal length of the camera (Eye-RIS 2.1)	$f$	12.5 mm
Virtual camera focal length	$f_v$	10 mm
Image point $U_L, V_L, U_R$	U, V	Pixel

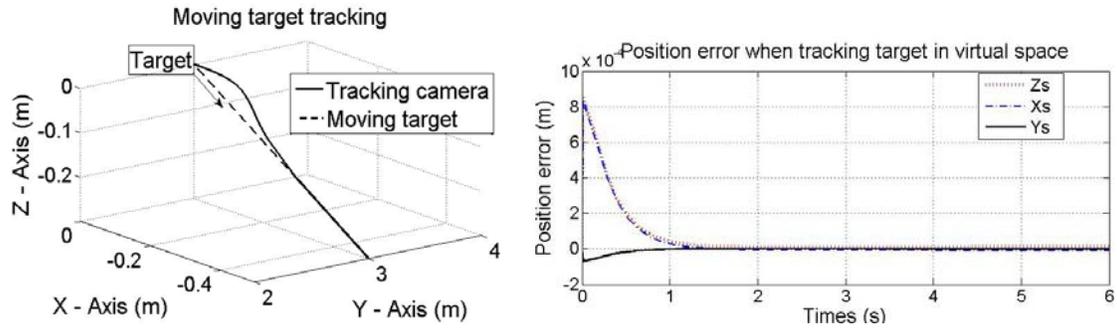
The unknown torque vector chosen as follows

$$\mathbf{d}(t) = 0.01 * \mathbf{S}_1 \quad (55)$$

In simulations, the target moves on workspace limited in 1m then choose  $\lambda > \text{Max}(\text{max}(x_v), \text{max}(y_v))$ ;  $\lambda = 2$ .

**Simulation 1: Moving target in a straight line**

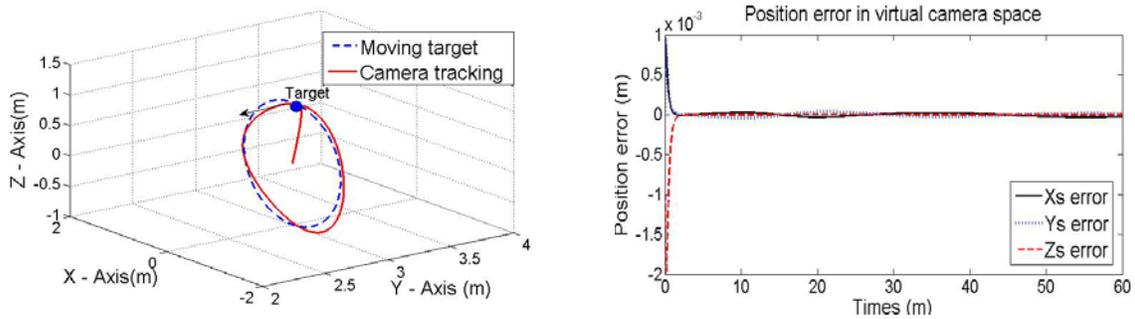
Moving target is from point A (0m,3m,0m) to B (-0.5m, 3m, -0.3m) on the plane  $Z_c O_c X_c$  and far away from coordinate origin  $Y_c = 3m$  in the camera coordinate  $O_c X_c Y_c Z_c$ . Target's moving time  $T = 6$  (s) with moving speed is  $v \sim 10$  (cm/s). Simulation results are given in Fig. 6.



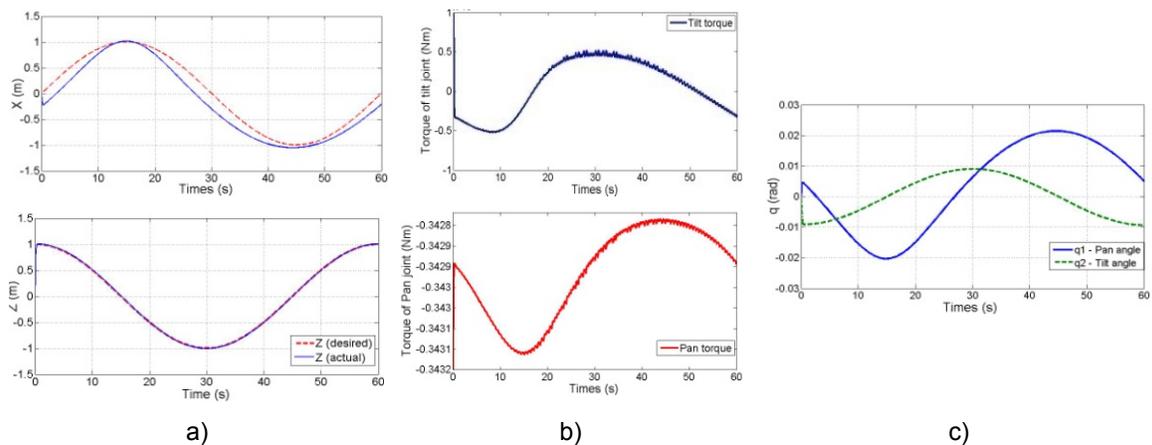
**Figure 6.** Tracking error coordinates when moving target in a straight line

**Simulation 2: Moving target in a circle**

Moving target follows the circle with center coordinates at  $O (0, 0, 0)$ , radius  $r = 1$  on the plane  $Z_c O_c X_c$  and far away from coordinate origin  $Y_c = 3m$  in the camera coordinate  $O_c X_c Y_c Z_c$ . Target moves around a circle in time  $T = 60s$ . Simulation results are given in Figs. 7 and 8.



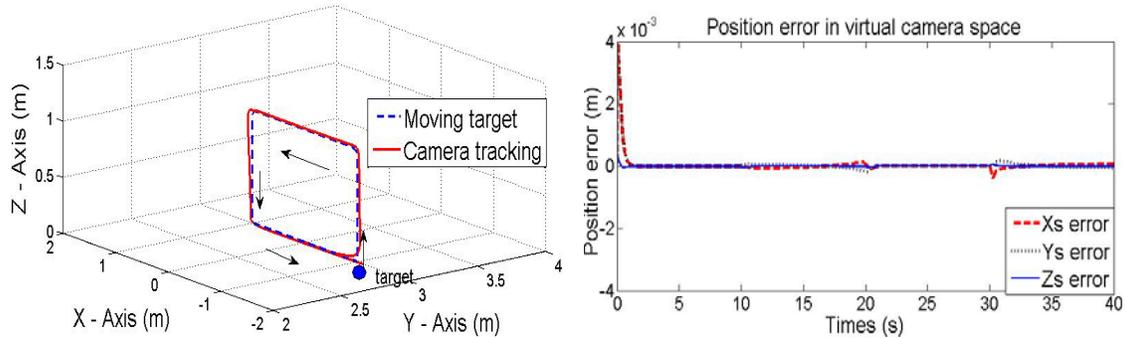
**Figure 7.** Tracking error coordinates when moving target follows a circle.



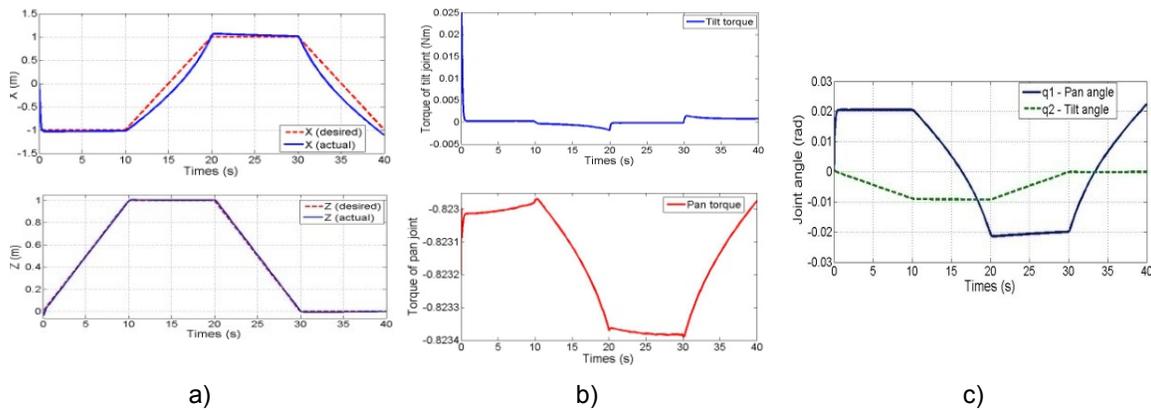
**Figure 8.** a) Tracking error coordinates in X, Z axes. b) Torques of pan and tilt joints. c) Joints angle  $q$

**Simulation 3: Moving target in a rectangle with changing velocity.**

Moving target follows the rectangle as figure 8 from point (-1m, 3m, 0m) to (-1m, 3m, 1m) in 10 seconds. In the next 10 seconds, target moves from point (-1m, 3m, 1m) to (1m, 3m, 1m) and the same to return to the starting point. Simulation results are given in Figs. 9 and 10.



**Figure 9.** Tracking error coordinates when moving target is in a rectangle with changing velocity.



**Figure 10.** a) Tracking error coordinates in X, Z axes. b) Torques of pan and tilt joints. c) Joints angle  $q$

**Simulation 4: Moving target in space with random velocity and direction.**

The target move following trajectory:  $(x, y, z) = (3, 3, 0.5)$  in  $O_0$  coordinates when  $t = 0$ . Process variability of angular and straight velocity with time is:  $y_t = 0.1t$ . The movement of the target following x, z axes is plane-parallel motion:

$$0 \leq t < 5s : v = 0.5 \text{ (m/s)}. \omega = 0 \text{ (rad/s)}.$$

$$5 \leq t < 10s : v = 0.5 + 0.15\sin((t-5)\pi/10) \text{ (m/s)}; \omega = 0.15\sin((t-5)\pi/10) \text{ (rad/s)};$$

$$10 \leq t < 15s : v = 0.75 \text{ (m/s)}. \omega = 0 \text{ (rad/s)};$$

$$15 \leq t < 20s : v = 0.75 + 0.15\sin((t-15)\pi/10) \text{ (m/s)}; \omega = -0.15\sin((t-15)\pi/10) \text{ (rad/s)};$$

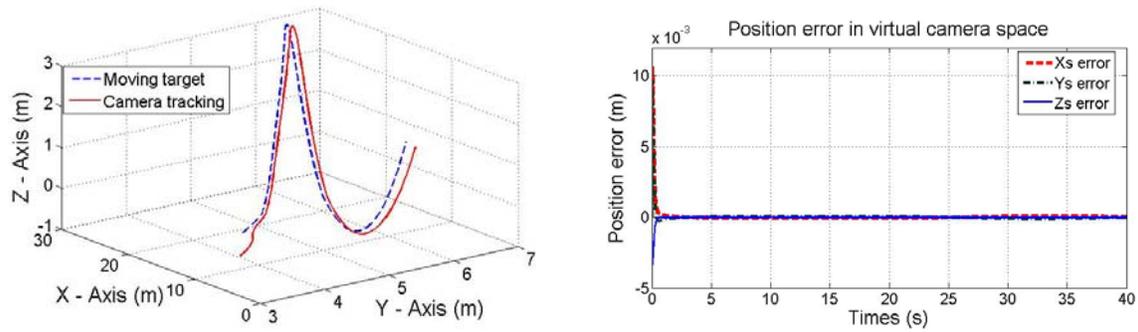
$$20 \leq t < 25s : v = 0.75 \text{ (m/s)}. \omega = -0.15 \text{ (rad/s)};$$

$$25 \leq t < 30s : v = 0.75 \text{ (m/s)}. \omega = -0.15 - 0.15\sin((t-25)\pi/10) \text{ (rad/s)};$$

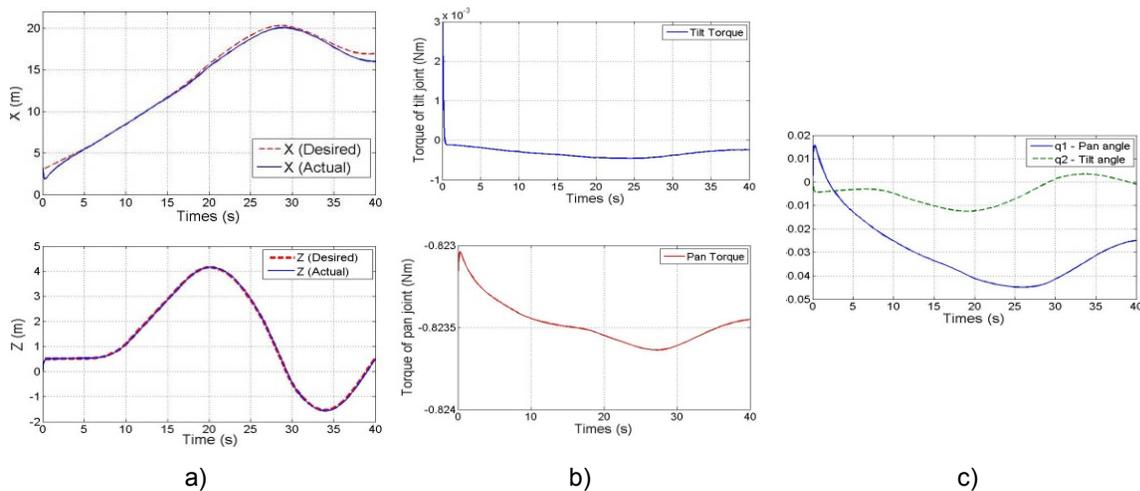
$$t > 30s : v = 0.5 \text{ (m/s)}. \omega = -0.3 \text{ (rad/s)}.$$

$$\lambda = 20.$$

Simulation results are given in Figs. 11 and 12.

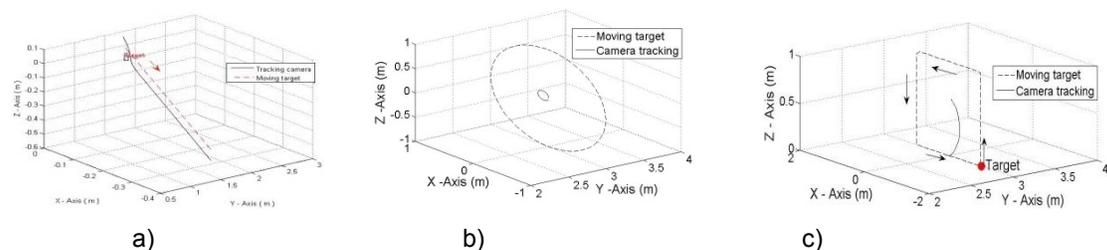


**Figure 11.** Tracking error coordinates when moving target is in space with random velocity and direction.



**Figure 12.** a) Tracking error coordinates in X, Z axes. b) Torques of pan and tilt joints. c) Joints angle  $q$

In the simulation 1 - moving target in a straight line in Fig. 6, it is found out that the error converges to zero. When target follows a circle (simulation 2), the system can track targets, but the error is still great so it needs to improve. When the function  $\cos$  or  $\sin$  is changes from positive to negative and in the opposite at  $\frac{1}{4}$ ,  $\frac{3}{4}$ ,  $0$ ,  $\frac{1}{2}$  circular of the circle, the tracking target error is fluctuations, but it still tracks the target. Different from (Chung and Cat, 2015), it lost tracking at  $\frac{1}{2}$  circular. In simulation 3, 4, when the target moves following a rectangle and random in the space, the system still captures. When the moving target simultaneous changed orientation and velocity, the initial error of the system increases, but it still tracks target rapidly because external effects as well as the effects of uncertain parameters in the system model are well compensated by neural network controller based on-line learning algorithm. The impact of noise and uncertain parameters is reduced. Otherwise, image and robot Jacobian matrices ( $J_{img}$  and  $J_{robot}$ ) have not a singularity. This condition helps the system move follow in the complex way.



**Figure 13.** Tracking error coordinates without part of neural network in the controller. a) moving target in a straight line, b) moving target follows a circle, c) moving target is in a rectangle with changing velocity

The controller without neural network can track a target following a straight line with acceptable error (Chung and Cat, 2015). However, when moving target follows a circle and rectangle, the result of tracking has so great error, even loss of tracking when target follows a rectangle with changing velocity (Fig 11).

## 5. DISCUSSION AND CONCLUSION

The problem of visual tracking control of pan-tilt-stereo camera system is dialled with when there are uncertainties in both the Jacobian matrix and the dynamics of the system. A 3D virtual stereo camera model having a full-rank of image Jacobian is constructed to solve local minima and image space singularity problem of the classical image Jacobian.

Furthermore, the authors propose a robust visual control scheme with online learning RBF neural network to compensate the effects of uncertainties. Due to online learning algorithm of RBF neural network continuously update is done only with some multiplication and integral. Therefore the calculations should not be too large, in accordance with real-time system. The asymptotic stability of the overall system is proved by Lyapunov stability method. Simulations show that the proposed control scheme is relatively effective even in the case of target moving in a circle or rectangle meanwhile the dynamics and the Jacobian of the pan-tilt robot are not known exactly.

In the other hand, all parameter in the controller can be optimized to have better results. in this paper, I only optimized two parameters. Other parameters such as  $\eta$ ,  $\delta$ ,  $\mathbf{G}$ ,  $\mathbf{H}$  also can be optimized. I will consider all effect of the optimization parameters in next step of my research.

## 6. REFERENCES

- Antonelli, G., 2008, Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems, *IEEE International Conference of Robotics and Automation, Pasadena, CA, USA*, 1993-1999.
- Azar, D., Fayad, K., Daoud, C., 2016, A combined ant colony optimization and simulated annealing algorithm to assess stability and fault-proneness of classes based on internal software quality attributes, *International Journal of Artificial Intelligence*, **14**, 2, 137-156.
- Bo, Y., Huiguang, L., Zhenbo, K., Honglei, J., 2012, Hamiltonian-based binocular visual servoing of camera-in-hand robotic systems, *International Conference on Modeling, Identification and Control, Wuhan, China*, 388-393.
- Cai, C., Dean-Leon, W., Mendoza, D., Soman, N., Knoll, A., 2013, Uncalibrated 3D stereo image-based dynamic visual servoing for robot manipulators, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan*, 63-70.
- Cat, P.T., Minh, N.T., 2009, Robust neural control of robot-camera visual tracking, *IEEE International Conference on Control and Automation, Christchurch, New Zealand*, 1825-1830.
- Cotter, N.E., 1990, The Stone-Weierstrass and Its application to neural networks. *IEEE Transactions on Neural Networks*, **1**, 4, 290-295.
- Chung, L.V., Cat, P.T., 2015, A new control method for stereo visual servoing system with pan-tilt platform, *Journal of Computer Science and Cybernetics*, **31**, 2, 107-122.
- Hashimoto, H., Kubota, T., Sato, M., Harashima, F., 1992, Visual control of robotics manipulator based on neural networks, *IEEE Transactions on Industrial Electronics*, **9**, 490-496.
- Kelly, R., Carelli, R., Nasisi, O., Kuchen, B., Reyes, F., 2000, Stable visual servoing of camera-in-hand robotic systems, *IEEE/ASME Transactions on Mechatronics*, **5**, 1, 39-48.
- Li, H., Jin, M., Zou, L., 2009, A New Binocular Stereo Visual Servoing Model, *IEEE International Conference on Computational Intelligence and Industrial Application, Wuhan, China*, 461-465.

- Mostafa, Z.A., Noor, H.A., Rehab, M.D., 2016, Multi-objective differential evolution algorithm with a new improved mutation strategy, *International Journal of Artificial Intelligence*, **14**, 2, 23-41.
- Murata, N., Yoshizawa, S., Amari, S., 1994, Network information criterion-determining the number of hidden units for an artificial neural network model, *IEEE Transactions on Neural networks*, **5**, 6, 856-887.
- Oscar, N., Ricardo, C., 2003, Adaptive servo visual robot control, *Robotics and Autonomous Systems* **43**, 51-78.
- Precup, R.-E., Sabau, M.-C., Petriu, E.M., 2015, Nature-inspired optimal tuning of input membership functions of Takagi-Sugeno-Kang fuzzy models for anti-lock braking systems, *Applied Soft Computing*, **27**, 575-589.
- Qiu, S., Li, Z., He, W., Zhang, L., Yang, C., Su, C.-Y., 2017, Brain machine interface and visual compressive sensing based teleoperation control of an exoskeleton robot, *IEEE Transactions on Fuzzy Systems*, **25**, 1, 58-69.
- Shibata, M., Eto, H., Ito, M., 2010, Image-based visual tracking to fast moving target for active binocular robot, *IEEE International Conference on Industrial Electronics Society, Glendale, AZ, USA*, 2727-2732.
- Shibata, M., Eto, H., Ito, M., 2011, Visual tracking control for stereo vision robot with high gain controller and high speed cameras, *IEEE International Symposium on Access Spaces, Yokohama, Japan*, 288-293.
- Sho-Tsung, K., Chung-Yi, Y., Ming, T.H., 2010, Design and implementation of a stereo vision- guided omnidirectional mobile robot for real-time object tracking, *SICE Annual Conference, Taipei, Taiwan*, 2968 - 2974.
- Siciliano, B., Sciavicco, L., Villani, G., Oriolo, G., 2009, Modeling and control of robot manipulators, *McGraw-Hill International editions*, 614p.
- Tae-Il, K., Wook, B., Changhun, L., Tae-jae, L., Luigi, V., Giuseppe, O., 2011, Vision system for mobile robots for tracking moving targets, based on robot motion and stereo vision information, *SI International IEEE, Kyoto, Japan*, 634-639.
- Ukida, H., Terama, Y., Ohnishi, H., 2012, Object tracking system by adaptive pan-tilt-zoom cameras and arm robot. *SICE Annual Conference, Akita, Japan*, 1920-1925.
- Vašćák, J., 2012, Adaptation of fuzzy cognitive maps by migration algorithms, *Kybernetes*, **41**, 3 - 4, 429-443.
- Widodo, B., Ari, S., Djoko, P., Achmad, J., 2011, A navigation system for service robot using stereo vision and kalman filtering, *International Conference on Control, Automation and Systems, Gyeonggi-do, Korea*, 1771-1776.
- Yang, Z., Che, Y., Cheng, E., K. W., 2007, Genetic algorithm-based RBF neural network load forecasting model, *IEEE Power Engineering Society General Meeting, Tampa, FL, USA*, 1298-1303.
- Yang, C., Wang, X., Li, Z., Li, Y., Su, C.-Y., 2017, Teleoperation control based on combination of wave variable and neural networks, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, **47**, 8, 2125-2136.
- Yang, C., Wang, X., Li, Z., Li, Y., Su, C.-Y., 2017, Neural control of bimanual robots with guaranteed global stability and motion precision, *IEEE Transactions on Industrial Informatics*, **13**, 3, 1162-1171.