The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Vision-based Robotic Tracking of Moving Object with Dynamic Uncertainty

Hesheng Wang, *Member, IEEE*, Yun-Hui Liu, *Fellow, IEEE,* Weidong Chen, *Member IEEE*

***Abstract*—This paper presents a new controller for locking a moving object in 3-D space at a particular position (for example the center) on the image plane of a camera mounted on a robot by actively moving the camera. The controller is designed to cope with both unknown robot dynamics parameters and unknown motion of the object. Based on the fact that the unknown position of the moving object appears linearly in the closed-loop dynamics of the system if the depth-independent image Jacobian is used, we developed a nonlinear observer to estimate the 3-D motion of the object on-line and an adaptive algorithm to estimate the robot dynamic parameters. With a full consideration of dynamic responses of the robot, we employed the Lyapunov method to prove asymptotic convergence of the image errors. Experimental results are presented to support the approach in this paper.**

***Index Terms*—Moving Object, Tracking, Nonlinear Observer.**

I. INTRODUCTION

IMAGE-BASED visual servoing is a problem of realizing motion control of robots by using the visual feedback from a camera. In many applications, robots need to track moving objects using visual feedback from cameras mounted on them. Technically, robotic vision-based tracking is a problem of locking the projection of the target object on the image plane to a particular position by controlling robot motion. For image-based visual servoing, many kinematics-based methods [1][2] have been developed. However, kinematics-based methods cannot yield high performance and guarantee stability because the nonlinear forces in robot dynamics are neglected. The nonlinear forces not only decline the control accuracy but also cause instability of the system. Dynamic visual servoing were studied by Hashimoto *et al*. [3] provided that the camera intrinsic and extrinsic parameters are calibrated. To cope with uncalibrated problems, Kelly [4] proposed to employ an adaptive algorithm to estimate the unknown camera parameters on-line. Deluca et al [5] developed an observer for depth feature in image-based visual servoing. To solve 3-D uncalibrated visual servoing with robot dynamics, new adaptive controllers [6][7] were proposed in our recent work. In [8][9], we extend our work to dynamic tracking of manipulators with uncalibrated fixed camera.

However, the approaches mentioned above all assumed that the objects to be tracked or the camera are stationary, and hence cannot be employed to tracking moving objects by active camera. The controller developed by Hashimoto *et al.*[10] incorporated the nonlinear robot dynamics in controller design and employed a motion estimator to estimate motion of an object in a plane. Dixon *et al.*[11] presented a Lyapunov-based method for designing dynamic controller for tracking objects in planar motion. However, those methods cannot be employed for tracking objects in the 3-D general motion. To track objects in 3-D motion, Hashimoto [12] proposed a nonlinear observer for estimating its motion velocity with a strong assumption that the acceleration of the target object is zero. In real applications, the target object can rarely move at a constant velocity.

Our objective is to develop a controller for locking a moving object in 3-D space at the center on the image plane of a camera mounted on a robot by actively moving the camera. To achieve this objective, we developed a new controller to cope with both unknown motion of the object and the unknown robot dynamics parameters. In this paper, the first technical issue to be addressed is on-line motion estimation of the moving object, which is the most crucial part in this work. Based on the fact that the kinematics and dynamics of the system can be linearly parameterized by the unknown position of the moving object if the depth-independent image Jacobian is used, we developed a nonlinear observer to estimate the 3-D motion of the object on-line. The second technical issue we investigated is design of a dynamic tracking controller. The controller is designed based on the nonlinear observer and the depth-independent image Jacobian. By using the

Manuscript received February 23, 2010. This work was supported in part by the Hong Kong Research Grants Council under Grant 414406 and Grant 414707, in part by the Natural Science Foundation of China under Grant 60334010, Grant 60475029, Grant 60775062 and Grant 60934006, in part by the National High Technology Research and Development Program of China under Grant 2006AA040203, in part by the Program for New Century Excellent Talents in University under Grant NCET-07-0538, and in part by the State Key Laboratory of Robotics and Systems (HIT)

H. Wang and W. Chen are with Department of Automation, Shanghai Jiao Tong University, Shanghai, China. H. Wang and W. Chen are also with the State Key Laboratory of Robotics and System, Harbin Institute of Technology, China.{wanghesheng,wdchen}@sjtu.edu.cn

Y.-H. Liu is with Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong. Y.-H. Liu is also with Joint Center for Intelligent Sensing and Systems, National University of Defense Technology, Hunan, China

depth-independent image Jacobian, it is possible to eliminate the nonlinear effect of the depth in the perspective projection so that the unknown position of the moving object appears linearly in the closed-loop dynamics of the system. The third technical issue is to design an adaptive algorithm to estimate the unknown robot dynamic parameters on-line. The Lyapunov method is used to prove the stability and convergence of the image errors. Finally, we implement the controller in robotic hand-eye system. Experiments results verified the performance of the proposed control method.

II. Kinematic and Dynamic Modeling

A. Problem Statement

Fig. 1 illustrates the robotic hand-eye system in which a camera mounted on a robot manipulator is to trace an object with unknown 3D motion. Suppose that the camera is a pin-hole camera with perspective projection. Furthermore, assume that the camera intrinsic and extrinsic parameters are calibrated. The problem addressed is defined as follows:

Problem 1: Assume the motion of the object and the robot dynamic parameters are both unknown, how to control motion of the robot so that the projection of the moving object is fixed at the center on the image plane.

B. Kinematics

Denote the joint angle of the manipulator by $\mathbf{q}(t)$, and the homogeneous coordinates of the feature point w.r.t. the robot base frame by $\mathbf{x}(t)$. Denote the homogenous transform matrix of the end-effector with respect to the base frame by $\mathbf{T_e}(\mathbf{q}(t))$. Under the perspective projection model, the image position is

$$\mathbf{y}(t) = \frac{1}{z(t)} \mathbf{P} \mathbf{T}_e^{-1}(\mathbf{q}(t))\mathbf{x}(t) \qquad (1)$$

where $\mathbf{P}$ is the matrix consisting of the first two rows of the matrix $\mathbf{M}$, which represents the known and constant 3×4 *perspective projection matrix*. Let $\mathbf{m}_i^T$ denotes the *i*-th row vector of the matrix $\mathbf{M}$, the depth *z(t)* of the feature point is

$$z(t) = \mathbf{m}_3^T \mathbf{T}_e^{-1}(\mathbf{q}(t))\mathbf{x}(t) \qquad (2)$$

By differentiating eq. (2),

$$\dot{z}(t) = \frac{\partial\left(\mathbf{m}_3^T \mathbf{T}_e^{-1}(\mathbf{q}(t))\mathbf{x}(t)\right)}{\partial \mathbf{q}(t)} \dot{\mathbf{q}}(t) + \mathbf{m}_3^T \mathbf{T}_e^{-1}(\mathbf{q}(t))\dot{\mathbf{x}}(t) \qquad (3)$$

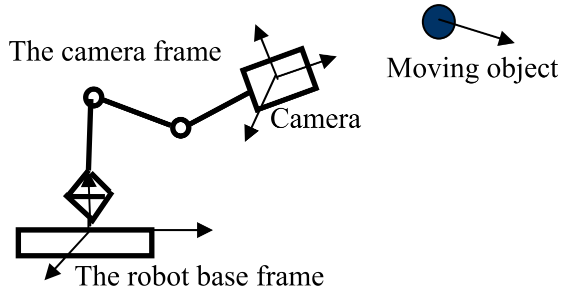


Fig. 1. Tracking a moving object using a robotic eye-in-hand system.

By differentiating eq. (1) and combining with (3), we obtain the following velocity relation:

$$\begin{aligned} z(t)\dot{\mathbf{y}}(t) &= \underbrace{(\mathbf{P} - \mathbf{y}(t)\mathbf{m}_3^T)\frac{\partial(\mathbf{T}_e^{-1}(\mathbf{q}(t))\mathbf{x}(t))}{\partial \mathbf{q}}}_{\mathbf{D}(\mathbf{q}(t),\mathbf{y}(t),\mathbf{x}(t))}\dot{\mathbf{q}}(t) \\ &+ \underbrace{(\mathbf{P} - \mathbf{y}(t)\mathbf{m}_3^T)\mathbf{T}_e^{-1}(\mathbf{q}(t))}_{\mathbf{B}(\mathbf{q}(t),\mathbf{y}(t))}\dot{\mathbf{x}}(t) \end{aligned} \qquad (4)$$

$\mathbf{D}(\mathbf{q}(t), \mathbf{y}(t), \mathbf{x}(t))$ is called depth-independent image Jacobian [11] and $\mathbf{B}(\mathbf{q}(t), \mathbf{y}(t))$ is called depth-independent motion Jacobian. Notice the following property [16].

Property 1: The Depth-independent image Jacobian matrix $\mathbf{D}(\mathbf{q}(t), \mathbf{y}(t), \mathbf{x}(t))$ has a rank of 2.

Note that (2) can be revised as

$$\underbrace{(\mathbf{P} - \mathbf{y}(t)\mathbf{m}_3^T)\mathbf{T}_e^{-1}(\mathbf{q}(t))}_{\mathbf{B}(\mathbf{q}(t),\mathbf{y}(t))}\mathbf{x}(t) = 0 \qquad (5)$$

Therefore, by differentiating above equation,

$$\mathbf{B}(\mathbf{q}(t), \mathbf{y}(t))\dot{\mathbf{x}}(t) + \dot{\mathbf{B}}(\mathbf{q}(t), \mathbf{y}(t))\mathbf{x}(t) = 0 \qquad (6)$$

Then (4) can be rewritten as follows:

$$z(t)\dot{\mathbf{y}}(t) - \mathbf{D}(\mathbf{q}(t), \mathbf{y}(t), \mathbf{x}(t))\dot{\mathbf{q}}(t) = -\dot{\mathbf{B}}(\mathbf{q}(t), \mathbf{y}(t))\mathbf{x}(t) \qquad (7)$$

Note that the unknown position $\mathbf{x}(t)$ of the feature point appears linearly in the matrix. This property will play an important role in our controller design and can be described as:

Property 2: For any parameters vector $\boldsymbol{\rho}$, the vector $z(\mathbf{q}(t), \boldsymbol{\rho}(t))\dot{\mathbf{y}}(t) - \mathbf{D}(\mathbf{q}(t), \mathbf{y}(t), \boldsymbol{\rho}(t))\dot{\mathbf{q}}(t)$ can be written as a linear form of the parameters:

$$z(\mathbf{q}(t), \boldsymbol{\rho}(t))\dot{\mathbf{y}}(t) - \mathbf{D}(\mathbf{q}(t), \mathbf{y}(t), \boldsymbol{\rho}(t))\dot{\mathbf{q}}(t) = -\dot{\mathbf{B}}(\mathbf{q}(t), \mathbf{y}(t))\boldsymbol{\rho}(t) \qquad (8)$$

where $\dot{\mathbf{B}}(\mathbf{q}(t), \mathbf{y}(t))$ does not depend on unknown parameters.

C. Robot Dynamics

The dynamic equation of the manipulator has the form:

$$\mathbf{H}(\mathbf{q}(t))\ddot{\mathbf{q}}(t) + [\frac{1}{2}\dot{\mathbf{H}}(\mathbf{q}(t)) + \mathbf{C}(\mathbf{q}(t), \dot{\mathbf{q}}(t))]\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}(t)) = \boldsymbol{\tau} \qquad (9)$$

where $\mathbf{H}(\mathbf{q}(t))$ is the $n\times n$ inertia matrix. The term $\mathbf{g}(q)$ represents the gravitational force, and $\boldsymbol{\tau}$ is the $n\times 1$ joint input. $\mathbf{C}(\mathbf{q}(t), \dot{\mathbf{q}}(t))$ is a skew-symmetric matrix and for any vector $\mathbf{s}$:

$$\mathbf{s}^T \mathbf{C}(\mathbf{q}(t), \dot{\mathbf{q}}(t))\mathbf{s} = 0 \qquad (10)$$

The robot dynamical equation enjoys one property that will be of great use to us in parameter estimation:

Property 3: The dynamics of the robots are linear in the physical parameters φ as

$$\begin{aligned} &\mathbf{H}(\mathbf{q}(t))\ddot{\mathbf{q}}(t) + (\frac{1}{2}\dot{\mathbf{H}}(\mathbf{q}(t)) + \mathbf{C}(\mathbf{q}(t), \dot{\mathbf{q}}(t)))\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}(t)) \\ &= \mathbf{F}(\mathbf{q}(t), \dot{\mathbf{q}}(t), \dot{\mathbf{q}}(t), \ddot{\mathbf{q}}(t))\varphi \end{aligned} \qquad (11)$$

where $\mathbf{F}(\mathbf{q}(t), \dot{\mathbf{q}}(t), \dot{\mathbf{q}}(t), \ddot{\mathbf{q}}(t))$ *is the regressor matrix.*

III. Image-based Visual Tracking

The control objective is to lock the projection of the feature point at a constant desired position $\mathbf{y}_d$ on the image plane.

A. Controller Design

Since the desired position on the image plane $\mathbf{y}_d$ is constant, the desired velocity and acceleration are both zero. For convenience, introduce the following nominal reference:

$$\dot{\mathbf{y}}_{\mathbf{r}}(t) = \dot{\mathbf{y}}_d - \lambda\Delta\mathbf{y}(t) = -\lambda\Delta\mathbf{y}(t) \tag{12}$$

where λ is a positive scalar, $\Delta\mathbf{y}$ is the position error of the feature point on the image plane, i.e.

$$\Delta\mathbf{y}(t) = \mathbf{y}(t) - \mathbf{y}_d \tag{13}$$

The error of the feature point to the defined nominal reference is given by

$$\mathbf{s}_y(t) = \dot{\mathbf{y}}(t) - \dot{\mathbf{y}}_r(t) = \Delta\dot{\mathbf{y}}(t) + \lambda\Delta\mathbf{y}(t) \tag{14}$$

Note that the trajectory errors $\Delta\mathbf{y}$ and $\Delta\dot{\mathbf{y}}$ are convergent to zero if the error vector $\mathbf{s}_y$ is convergent to zero.

Denote an estimation of the position of the feature point by $\hat{\mathbf{x}}(t)$, and the corresponding estimated depth by $\hat{z}(t)$. Then, the estimation error is given by

$$\Delta\mathbf{x}(t) = \hat{\mathbf{x}}(t) - \mathbf{x}(t) \tag{15}$$

Introduce the following nominal velocity $\dot{\mathbf{q}}_r(t)$:

$$\dot{\mathbf{q}}_r(t) = -\hat{\mathbf{D}}^{+}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t))(\hat{z}(t)\lambda\Delta\mathbf{y}(t) - \dot{\mathbf{B}}(\mathbf{q}(t),\mathbf{y}(t))\hat{\mathbf{x}}(t)) \tag{16}$$

where $\hat{\mathbf{D}}^{+}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t))$ is the pseudo-inverse of the estimated depth-independent image Jacobian matrix, which is calculated using the estimated motion parameters as follows:

$$\begin{aligned}&\hat{\mathbf{D}}^{+}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t))\\ &= \hat{\mathbf{D}}^{T}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t))[\hat{\mathbf{D}}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t))\hat{\mathbf{D}}^{T}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t))]^{-1}\end{aligned} \tag{17}$$

It should be noted that $\hat{\mathbf{D}}^{T}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t))$ must have a rank of 2 in order to guarantee the existence of the pseudo-inverse. By Property 1, we can conclude the existence of the pseudo-inverse.

The error vectors on the joint space are given by $\mathbf{s}_q(t)$:

$$\mathbf{s}_{\mathbf{q}}(t) = \dot{\mathbf{q}}(t) - \dot{\mathbf{q}}_r(t) \tag{18}$$

The nominal acceleration can be represented as:

$$\begin{aligned}\ddot{\mathbf{q}}_r(t) &= -\frac{d}{dt}\left\{\hat{\mathbf{D}}^{+}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t))\right\}(\hat{z}(t)\lambda\Delta\mathbf{y}(t) - \dot{\mathbf{B}}(\mathbf{q}(t),\mathbf{y}(t))\hat{\mathbf{x}}(t))\\ &\quad + \hat{\mathbf{D}}^{+}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t))(\dot{\mathbf{B}}(\mathbf{q}(t),\mathbf{y}(t))\dot{\hat{\mathbf{x}}}(t) + \ddot{\mathbf{B}}(\mathbf{q}(t),\mathbf{y}(t))\hat{\mathbf{x}}(t))\\ &= \bar{\ddot{\mathbf{q}}}_r(t) + \hat{\mathbf{D}}^{+}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t))\ddot{\mathbf{B}}(\mathbf{q}(t),\mathbf{y}(t))\hat{\mathbf{x}}(t)\end{aligned} \tag{19}$$

where only the last term depends on joint and visual accelerations. $\bar{\ddot{\mathbf{q}}}_r(t)$ includes all the terms without joint and visual accelerations. It is reasonable to assume that the acceleration of the object is bounded. Furthermore, suppose that we can estimate joint acceleration $\hat{\ddot{\mathbf{q}}}(t)$ and visual acceleration $\hat{\ddot{\mathbf{y}}}(t)$ from the joint velocity and visual velocity with bounded errors whose upper bounds are denoted by ε_1 and ε_2, respectively. Replacing the joint and visual acceleration by the estimated ones, we obtain

$$\hat{\ddot{\mathbf{q}}}_r(t) = \bar{\ddot{\mathbf{q}}}_r(t) + \hat{\mathbf{D}}^{+}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t))\hat{\ddot{\mathbf{B}}}(\mathbf{q}(t),\mathbf{y}(t))\hat{\mathbf{x}}(t) \tag{20}$$

Multiplying the $\mathbf{H}(\mathbf{q}(t))$ from the left to (20) and using (19), we obtain

$$\begin{aligned}\mathbf{H}(\mathbf{q}(t))\hat{\ddot{\mathbf{q}}}_r(t) &= \mathbf{H}(\mathbf{q}(t))\ddot{\mathbf{q}}_r(t)\\ &\quad + \mathbf{H}(\mathbf{q}(t))\hat{\mathbf{D}}^{+}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t))(\hat{\ddot{\mathbf{B}}}(\mathbf{q}(t),\mathbf{y}(t)) - \ddot{\mathbf{B}}(\mathbf{q}(t),\mathbf{y}(t)))\hat{\mathbf{x}}(t)\\ &= \mathbf{H}(\mathbf{q}(t))\ddot{\mathbf{q}}_r(t) + \mathbf{O}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t),\Delta\ddot{\mathbf{q}}(t),\Delta\ddot{\mathbf{y}}(t))\end{aligned} \tag{21}$$

where $\Delta\ddot{\mathbf{q}}(t)$ and $\Delta\ddot{\mathbf{y}}(t)$ are estimate joint and visual acceleration errors, respectively. And the last term in (21) are bounded by

$$\left\|\mathbf{O}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t),\Delta\ddot{\mathbf{q}}(t),\Delta\ddot{\mathbf{y}}(t))\right\| \le c \tag{22}$$

where c is a positive constant

Based on the nominal references defined, we propose the following controller:

$$\begin{aligned}\boldsymbol{\tau} &= \mathbf{F}(\mathbf{q}(t),\dot{\mathbf{q}}(t),\dot{\mathbf{q}}_r(t),\hat{\ddot{\mathbf{q}}}_r(t))\hat{\varphi}(t)\\ &\quad - \mathbf{K}_1\mathbf{s}_{\mathbf{q}}(t) - \hat{\mathbf{D}}^{T}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t))\mathbf{K}_2\hat{z}(t)\mathbf{s}_{\mathbf{y}}(t) - \mathbf{K}_3\,\mathrm{sgn}(\mathbf{s}_{\mathbf{q}}(t))\end{aligned} \tag{23}$$

where $\mathbf{K}_1$, $\mathbf{K}_2$ and $\mathbf{K}_3$ are positive-definite symmetric gain matrices. The first three terms are to compensate for the inertia forces, the nonlinear centrifugal and Coriolis forces, and the gravitational force, respectively. The fourth term represents a feedback in the joint space. The fifth term is the image error feedback. The last term is to compensate for the error caused by introducing the new term $\hat{\ddot{\mathbf{q}}}_r(t)$.

Submitting this controller into the robot dynamics results in the following closed-loop dynamics:

$$\begin{aligned}&\mathbf{H}(\mathbf{q}(t))\ddot{\mathbf{q}}(t) + (\frac{1}{2}\dot{\mathbf{H}}(\mathbf{q}(t)) + \mathbf{C}(\mathbf{q}(t),\dot{\mathbf{q}}(t)))\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}(t))\\ &= \mathbf{F}(\mathbf{q}(t),\dot{\mathbf{q}}(t),\dot{\mathbf{q}}_r(t),\hat{\ddot{\mathbf{q}}}_r(t))\hat{\varphi}(t) - \mathbf{K}_1\mathbf{s}_{\mathbf{q}}(t)\\ &\quad - \hat{\mathbf{D}}^{T}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t))\mathbf{K}_2\hat{z}(t)\mathbf{s}_{\mathbf{y}}(t) - \mathbf{K}_3\,\mathrm{sgn}(\mathbf{s}_{\mathbf{q}}(t))\end{aligned} \tag{24}$$

By property 3,

$$\begin{aligned}&\mathbf{H}(\mathbf{q}(t))\hat{\ddot{\mathbf{q}}}_r(t) + [\frac{1}{2}\dot{\mathbf{H}}(\mathbf{q}(t)) + \mathbf{C}(\mathbf{q}(t),\dot{\mathbf{q}}(t))]\dot{\mathbf{q}}_r(t)\\ &= \mathbf{F}(\mathbf{q}(t),\dot{\mathbf{q}}(t),\dot{\mathbf{q}}_r(t),\hat{\ddot{\mathbf{q}}}_r(t))\varphi(t)\end{aligned} \tag{25}$$

Equation (25) can be rewritten as

$$\begin{aligned}&\mathbf{H}(\mathbf{q}(t))\hat{\ddot{\mathbf{q}}}_r(t) - \mathbf{H}(\mathbf{q}(t))\ddot{\mathbf{q}}_r(t) + \mathbf{H}(\mathbf{q}(t))\ddot{\mathbf{q}}_r(t)\\ &\quad + [\frac{1}{2}\dot{\mathbf{H}}(\mathbf{q}(t)) + \mathbf{C}(\mathbf{q}(t),\dot{\mathbf{q}}(t))]\dot{\mathbf{q}}_r(t)\\ &= \mathbf{F}(\mathbf{q}(t),\dot{\mathbf{q}}(t),\dot{\mathbf{q}}_r(t),\hat{\ddot{\mathbf{q}}}_r(t))\varphi(t)\end{aligned} \tag{26}$$

From (21), we can obtain

$$\mathbf{H}(\mathbf{q}(t))\hat{\ddot{\mathbf{q}}}_r(t) - \mathbf{H}(\mathbf{q}(t))\ddot{\mathbf{q}}_r(t) = \mathbf{O}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t),\Delta\ddot{\mathbf{q}}(t),\Delta\ddot{\mathbf{y}}(t)) \tag{27}$$

Submitting (27) into (26) give us

$$\mathbf{H}(\mathbf{q}(t))\ddot{\mathbf{q}}_r(t)+[\frac{1}{2}\dot{\mathbf{H}}(\mathbf{q}(t))+\mathbf{C}(\mathbf{q}(t),\dot{\mathbf{q}}(t))]\dot{\mathbf{q}}_r(t)$$
$$+\mathbf{O}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t),\Delta\ddot{\mathbf{q}}(t),\Delta\ddot{\mathbf{y}}(t))=\mathbf{F}(\mathbf{q}(t),\dot{\mathbf{q}}(t),\dot{\mathbf{q}}_r(t),\hat{\dot{\mathbf{q}}}_r(t))\varphi(t) \quad (28)$$

Though equation (24) minus equation (28), we can obtain the following equation

$$\begin{aligned}&\mathbf{H}(\mathbf{q}(t))\dot{\mathbf{s}}_{\mathbf{q}}(t)+[\frac{1}{2}\dot{\mathbf{H}}(\mathbf{q}(t))+\mathbf{C}(\mathbf{q}(t),\dot{\mathbf{q}}(t))]\mathbf{s}_{\mathbf{q}}(t)\\&=\mathbf{F}(\mathbf{q}(t),\dot{\mathbf{q}}(t),\dot{\mathbf{q}}_r(t),\hat{\dot{\mathbf{q}}}_r(t))\Delta\hat{\varphi}(t)\\&-\mathbf{K}_1\mathbf{s}_{\mathbf{q}}(t)-\hat{\mathbf{D}}^T(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t))\mathbf{K}_2\hat{z}(t)\mathbf{s}_{\mathbf{y}}(t)\\&-\mathbf{K}_3\,\mathrm{sgn}(\mathbf{s}_{\mathbf{q}}(t))+\mathbf{O}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t),\Delta\ddot{\mathbf{q}}(t),\Delta\ddot{\mathbf{y}}(t))\end{aligned} \quad (29)$$

where $\Delta\varphi(t)=\hat{\varphi}(t)-\varphi(t)$ is the robot physical parameters estimation errors.

B. Observer Design

Since the robots are to track moving objects whose motions are unknown, the crucial issue in the controller (23) is how to estimate the unknown trajectories $(\hat{\mathbf{x}}(t),\dot{\hat{\mathbf{x}}}(t),\ddot{\hat{\mathbf{x}}}(t))$ of the feature point moving in the 3-D space. We propose the following observer for on-line motion estimation of the feature point:

$$\dot{\hat{\mathbf{x}}}(t)=\mathbf{B}^+(\mathbf{q}(t),\mathbf{y}(t))\{-\dot{\mathbf{B}}(\mathbf{q}(t),\mathbf{y}(t))\hat{\mathbf{x}}-\mathbf{K}_4\xi(t)\} \quad (30)$$

where $\mathbf{K}_4$ are positive-definite symmetric gain matrix. $\mathbf{B}^+(\mathbf{q}(t),\mathbf{y}(t))$ is the pseudo-inverse of the matrix $\mathbf{B}(\mathbf{q}(t),\mathbf{y}(t))$ defined in equation (5). $\dot{\mathbf{B}}(\mathbf{q}(t),\mathbf{y}(t))$ is the time-derivative of matrix $\mathbf{B}(\mathbf{q}(t),\mathbf{y}(t))$. And

$$\xi(t)=\mathbf{B}(\mathbf{q}(t),\mathbf{y}(t))\Delta\mathbf{x}(t)=\mathbf{B}(\mathbf{q}(t),\mathbf{y}(t))\hat{\mathbf{x}}(t) \quad (31)$$

By differentiating eq. (31), we obtain the following relation:

$$\dot{\xi}(t)=\dot{\mathbf{B}}(\mathbf{q}(t),\mathbf{y}(t))\hat{\mathbf{x}}(t)+\mathbf{B}(\mathbf{q}(t),\mathbf{y}(t))\dot{\hat{\mathbf{x}}}(t) \quad (32)$$

The nonlinear observer (30) gives the first order derivative of the estimation position, which can be used to calculate the estimated position by numerical integration. Then the estimation is applied in the controller equation (23).

As for the robot dynamics parameters, we design the following adaptive rule similar to the Slotine-Li algorithm to estimate on-line:

$$\frac{d}{dt}\hat{\varphi}(t)=-\frac{1}{\mathbf{K}_5}\mathbf{F}^T(\mathbf{q}(t),\dot{\mathbf{q}}(t),\dot{\mathbf{q}}(t),\hat{\dot{\mathbf{q}}}(t))\mathbf{s}_q(t) \quad (33)$$

where $\mathbf{K}_5$ are positive-definite symmetric gain matrix.

C. Stability Analysis

We here analyze the stability of the robot manipulator under the control of the proposed controller and observer algorithm. The Lyapunov method is used to prove the stability and convergence of the image errors. Barbalat Lemma is adopted to prove the stability of the system. We first consider the continuous property, which is necessary condition for Barbalat Lemma.

Lemma 1: $\mathbf{s}_q(t)$, $\xi(t)$ and $\hat{z}(t)\mathbf{s}_y(t)$ are uniformly continuous if $\mathbf{s}_q(t)$, $\xi(t)$ and $\Delta\varphi(t)$ are all bounded.

Proof: Note that

$$\begin{aligned}&\hat{\mathbf{D}}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t))\mathbf{s}_q\\&=\hat{\mathbf{D}}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t))(\dot{\mathbf{q}}(t)-\dot{\mathbf{q}}_r(t))\\&=\hat{\mathbf{D}}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t))\dot{\mathbf{q}}(t)+\hat{z}(t)\lambda\Delta\mathbf{y}(t)-\dot{\mathbf{B}}(\mathbf{q}(t),\mathbf{y}(t))\hat{\mathbf{x}}(t)\end{aligned} \quad (34)$$

From Property 2,

$$\hat{z}(t)\dot{\mathbf{y}}(t)-\hat{\mathbf{D}}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t))\dot{\mathbf{q}}(t)=-\dot{\mathbf{B}}(\mathbf{q}(t),\mathbf{y}(t))\hat{\mathbf{x}}(t) \quad (35)$$

Substituting (35) into (34), we obtain

$$\begin{aligned}\hat{\mathbf{D}}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t))\mathbf{s}_q&=\hat{z}(t)\dot{\mathbf{y}}(t)+\hat{z}(t)\lambda\Delta\mathbf{y}(t)\\&=\hat{z}(t)\mathbf{s}_y(t)\end{aligned} \quad (36)$$

If $\mathbf{s}_q(t)$ and $\xi(t)$ are all bounded, by the definition of $\xi(t)$ in (31), $\hat{\mathbf{x}}$ is bounded. From (36), we can conclude $\mathbf{s}_y(t)$ is bounded, which directly leads to $\dot{\mathbf{y}}(t)$ is bounded. Consequently, from (4), $\dot{\mathbf{q}}(t)$ is also bounded. Then, we can claim the boundedness of the $\dot{\mathbf{s}}_q(t)$ and $\dot{\hat{\mathbf{x}}}(t)$ from the closed-loop dynamics (29) and (30), respectively. Consequently, $\mathbf{s}_q(t)$ are uniformly continuous.

The boundedness of $\dot{\xi}(t)$ can be concluded by the boundedness of $\dot{\mathbf{q}}(t)$ and $\dot{\hat{\mathbf{x}}}(t)$ from eq.(32). Consequently, $\xi(t)$ are uniformly continuous.

Multiplying the $\hat{z}(t)\mathbf{s}_y^T(t)\mathbf{K}_2$ from the left to (36), we obtain

$$\hat{z}(t)\mathbf{s}_y^T(t)\mathbf{K}_2\hat{\mathbf{D}}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t))\mathbf{s}_q=\hat{z}^2(t)\mathbf{s}_y^T(t)\mathbf{K}_2\mathbf{s}_y(t) \quad (37)$$

By differentiating equation (36), we have

$$\frac{d}{dt}(\hat{z}(t)\mathbf{s}_y(t))=\dot{\hat{\mathbf{D}}}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t))\mathbf{s}_q+\hat{\mathbf{D}}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t))\dot{\mathbf{s}}_q \quad (38)$$

Since all the terms in the right-hand side of (38) are bounded, $\frac{d}{dt}(\hat{z}(t)\mathbf{s}_y(t))$ are also bounded. Consequently, $\hat{z}(t)\mathbf{s}_y(t)$ is uniformly continuous.

Following is the main result of this paper:

Theorem 1: Under the control of the proposed controller (23), the observer algorithm (30) for motion parameters estimation and the adaptive algorithm (33) for robot dynamic parameters estimation, the image error of the feature point is convergent to zero, i.e.

$$\lim_{t\to\infty}\Delta\mathbf{y}(t)=\mathbf{0} \quad (39)$$

$$\lim_{t\to\infty}\Delta\dot{\mathbf{y}}(t)=\mathbf{0} \quad (40)$$

Proof: Consider the following non-negative function:

$$V(t)=\frac{1}{2}\{\mathbf{s}_q^T(t)\mathbf{H}(\mathbf{q}(t))\mathbf{s}_q(t)+\xi^T(t)\xi(t)+\Delta\varphi^T(t)\mathbf{K}_5\Delta\varphi(t)\} \quad (41)$$

Multiplying the $\mathbf{s}_q^T(t)$ from the left to the closed-loop dynamic results in

$$\begin{aligned}\mathbf{s}_q^T(t)\mathbf{H}(\mathbf{q}(t))\dot{\mathbf{s}}_{\mathbf{q}}(t)+\frac{1}{2}\mathbf{s}_q^T(t)\dot{\mathbf{H}}(\mathbf{q}(t))\mathbf{s}_q(t) &= -\mathbf{s}_q^T(t)\mathbf{K}_1\mathbf{s}_q(t) \\ &-\mathbf{K}_3\left\|\mathbf{s}_{\mathbf{q}}(t)\right\|-\mathbf{s}_q^T(t)\hat{\mathbf{D}}^T(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t))\mathbf{K}_2\hat{z}(t)\mathbf{s}_{\mathbf{y}}(t) \\ &+\mathbf{s}_q^T(t)\mathbf{O}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t),\Delta\ddot{\mathbf{q}}(t),\Delta\ddot{\mathbf{y}}(t)) \\ &+\mathbf{s}_q^T(t)\mathbf{F}(\mathbf{q}(t),\dot{\mathbf{q}}(t),\dot{\mathbf{q}}_r(t),\hat{\ddot{\mathbf{q}}}_r(t))\Delta\hat{\varphi}(t)\end{aligned} \tag{42}$$

From (37), we have

$$\begin{aligned}\mathbf{s}_q^T(t)\mathbf{H}(\mathbf{q}(t))\dot{\mathbf{s}}_q(t)+\frac{1}{2}\mathbf{s}_q^T(t)\dot{\mathbf{H}}(\mathbf{q}(t))\mathbf{s}_q(t) &= \\ -\mathbf{s}_q^T(t)\mathbf{K}_1\mathbf{s}_q(t)-\hat{z}^2(t)\mathbf{s}_y^T(t)\mathbf{K}_2\mathbf{s}_y(t)&-\mathbf{K}_3\left\|\mathbf{s}_{\mathbf{q}}(t)\right\| \\ +\mathbf{s}_q^T(t)\mathbf{O}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t),\Delta\ddot{\mathbf{q}}(t),\Delta\ddot{\mathbf{y}}(t)) \\ +\mathbf{s}_q^T(t)\mathbf{F}(\mathbf{q}(t),\dot{\mathbf{q}}(t),\dot{\mathbf{q}}_r(t),\hat{\ddot{\mathbf{q}}}_r(t))\Delta\hat{\varphi}(t)\end{aligned} \tag{43}$$

Differentiating $V(t)$ result in

$$\begin{aligned}\dot{V}(t) &= \mathbf{s}_q^T(t)(\mathbf{H}(\mathbf{q}(t))\dot{\mathbf{s}}_q(t)+\frac{1}{2}\dot{\mathbf{H}}(\mathbf{q}(t)))\mathbf{s}_q(t) \\ &+\dot{\boldsymbol{\xi}}^T(t)\boldsymbol{\xi}(t)+\Delta\varphi^T(t)\mathbf{K}_5\Delta\dot{\varphi}(t)\end{aligned} \tag{44}$$

By multiplying the $\Delta\varphi^T(t)$ from the left to the adaptive rule of robot physical parameters (33), we obtain

$$\Delta\varphi^T(t)\mathbf{K}_5\Delta\dot{\varphi}(t) = -\Delta\varphi^T(t)\mathbf{F}(\mathbf{q}(t),\dot{\mathbf{q}}(t),\dot{\mathbf{q}}_r(t),\hat{\ddot{\mathbf{q}}}_r(t))\mathbf{s}_q(t) \tag{45}$$

By combining the nonlinear observer (30), (32), (44) and (45), we have

$$\begin{aligned}\dot{V}(t) &= -\mathbf{s}_{\mathbf{q}}^T(t)\mathbf{K}_1\mathbf{s}_{\mathbf{q}}(t)-\hat{z}^2(t)\mathbf{s}_y^T(t)\mathbf{K}_2\mathbf{s}_y(t) \\ &-\boldsymbol{\xi}^T(t)\mathbf{K}_4\boldsymbol{\xi}(t)-\mathbf{K}_3\left\|\mathbf{s}_{\mathbf{q}}(t)\right\| \\ &+\mathbf{s}_q^T(t)\mathbf{O}(\mathbf{q}(t),\mathbf{y}(t),\hat{\mathbf{x}}(t),\Delta\ddot{\mathbf{q}}(t),\Delta\ddot{\mathbf{y}}(t)) \\ &\le -\mathbf{s}_{\mathbf{q}}^T(t)\mathbf{K}_1\mathbf{s}_{\mathbf{q}}(t)-\hat{z}^2(t)\mathbf{s}_y^T(t)\mathbf{K}_2\mathbf{s}_y(t) \\ &-\boldsymbol{\xi}^T(t)\mathbf{K}_4\boldsymbol{\xi}(t)-\left\|\mathbf{s}_{\mathbf{q}}(t)\right\|(\mathbf{K}_3-c)\end{aligned} \tag{46}$$

Here $\mathbf{K}_3$ is selected by $\mathbf{K}_3 \ge c$, then $\dot{V}(t)$ is never positive. From equation (41) the function *V(t)* never increases its value so that it is upper bounded. From the definition (41), bounded *V(t)* directly implies that the joint velocity error $\mathbf{s}_q(t)$ and $\boldsymbol{\xi}(t)$ are all bounded.

By Lemma 1, $\mathbf{s}_q(t)$, $\boldsymbol{\xi}(t)$ and $\hat{z}(t)\mathbf{s}_y(t)$ are uniformly continuous. Therefore, from the Barbalat Lemma [17], we can conclude the convergence of the error $\mathbf{s}_y(t)$ as well as the image error $\Delta\mathbf{y}(t)$ to zero.

Remark 1: Note that from (24), the differential equations of the closed-loop dynamics has a discontinuous right-hand side. Let ζ denote system states and $\mathbf{F}(\zeta,t)$ denote the right-hand side of closed-loop system. Since the above proof requires that a solution exist for $\dot{\zeta}=\mathbf{F}(\zeta,t)$, we will discuss the existence of Filippov's generalized solution [13] to (24), similar to the arguments in [14]. Note that $\mathbf{F}(\zeta,t)$ is continuous almost everywhere except in the set $\{(\zeta,t)\,|\,\mathbf{s}_q=0\}$. Let $\mathcal{F}\ (\zeta,t)$ be an upper semi-continuous set-valued map which embeds the discontinuous differential equation $\dot{\zeta}=\mathbf{F}(\zeta,t)$ into the corresponding differential inclusions $\dot{\zeta}\in\mathcal{F}\ (\zeta,t)$. One of the most obvious choices of $\mathcal{F}\ (\zeta,t)$ is given by the closed convex hull of $\mathbf{F}(\zeta,t)$ (an explicit proof can be found in [15]). Then, we know that an absolute continuous solution exists to $\dot{\zeta}\in\mathcal{F}\ (\zeta,t)$ that is a generalized solution to $\dot{\zeta}=\mathbf{F}(\zeta,t)$.

IV. Experiments

We have implemented the proposed controller in a 3 DOF robot manipulator (Fig. 2) at our laboratory. The moment inertia about its vertical axis of the first link of the manipulator is 0.005kgm^2, the masses of the second and third links are 0.167 kg, and 0.1 kg, respectively. The lengths of the second and third links are, 0.145m and 0.1285m, respectively. A Prosilica camera connecting to IEEE 1394 card installed in a PC with Intel Pentium IV CPU acquires the video signal. The frame rate is 100fps.

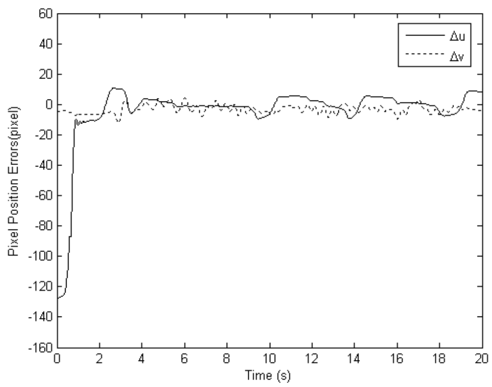
The calibration values of the intrinsic parameters of the camera are $\alpha = 2182$ pixels, $\gamma = 2186$ pixels, $u_0 = 340$ pixels and $v_0 = 199$ pixels. The initial estimations of the robot physical parameters are:

$$\begin{aligned}\hat{\varphi}(0) = (&0.0022\quad 0\quad 0.0032\quad 0\quad 0.0005\quad 0.0009\quad -0.0027 \\ &0.0005\quad 0.0035\quad 0.0005\quad 0.0012\quad -0.0001\quad 0.002 \\ &0.0004\quad 0.0259\quad 0.0061)^T\end{aligned}$$

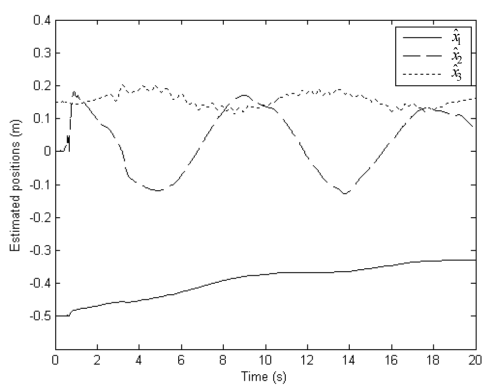
The control gains used are $\mathbf{K}_1 = 0.3$, $\mathbf{K}_2 = 0.0008$, $\mathbf{K}_3 = 0.5$, $\mathbf{K}_4 = 0.5$ and $\mathbf{K}_5 = 0.2$. The camera is mounted on the end-effector, the robot manipulator is to trace a moving object with an arbitrary motion in the 3D world. The objective is to continuously keep the projection of the target object on the image plane stationary.

Fig. 2. The experiment setup.

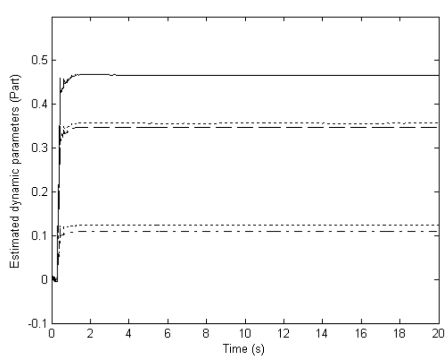
The position errors on the image plane as well as the estimated 3D position of the moving object are shown in Fig. 3, which confirms expected asymptotic convergence of the trajectory error to the zero. Fig. 3(b) plots the estimated 3D position of the moving object. Fig. 3(c) plots the estimated robot physical parameters. The nonlinear observer is only to guarantee that the tracking trajectory error converge to zero. The estimated parameters may not converge to the real ones. The sampling time used in the experiment is 17 ms.



(a)



(b)



(c)

Fig. 3. The experimental results: (a) The position errors on the image plane. (b) The estimated position of the moving object.(c) Estimated robot dynamic parameters (part).

V. CONCLUSIONS

This paper presents a new controller for visual tracking with eye-in-hand system. The objective is to lock the projection of the target object on the image plane to a particular position by controlling robot motion. The controller is designed to cope with both the nonlinear robot dynamics and unknown motion of the object. Based on the fact that the unknown position/velocity of the moving object appears linearly in the closed-loop dynamics of the system if the depth-independent image Jacobian is used, we developed a nonlinear observer to estimate the 3-D motion of the object on-line. We employed the Lyapunov method to prove asymptotic convergence of the image errors. Experiments have been conducted to demonstrate the performance of the proposed controller.

REFERENCES

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

[2] G. Hu, W. Mackunis, N. Gans, W. E. Dixon, J. Chen, A. Behal, D.M. Dawson, "Homography-Based Visual Servo Control with Imperfect Camera Calibration," *IEEE Trans. on Automatic Control*, Vol. 54, No. 6, pp. 1318-1324, 2009

[3] K. Hashimoto, T. Ebine and H. Kimura, "Dynamic visual feedback control for a hand-eye manipulator", *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 1863-1869, 1992.

[4] R. Kelly, R. Carelli, O. Nasisi, B. Kuchen and F. Reyes "Stable visual servoing of camera-in-hand robotic systems," *IEEE/ASME Trans. on Mechatronics,* vo. 5, no. 1, pp. 39-48, 2000.

[5] De Luca, A., G. Oriolo and P. Robuffo Giordano, "Feature depth observation for image-based visual servoing: theory and experiments," *Int. Journal of Robotics Research* Vol. 27***, No. 10***, pp. 1093-1116, 2008.

[6] H. Wang, Y. H. Liu, and D. Zhou, "Adaptive visual servoing using point and line features with an uncalibrated eye-in-hand camera," *IEEE Tran. on Robotics*, Vol. 24, no. 4, 2008

[7] Y. H. Liu, H. Wang, and K. K. Lam, "Uncalibrated visual servoing of robots using a depth-independent image Jacobian matrix", *IEEE Tran. on Robotics,* Vol.22, No. 4, 2006.

[8] H. Wang, Y. H. Liu and D. Zhou, "Dynamic visual tracking for manipulators using an uncalibrated fixed camera," *IEEE Trans. on Robotics,* vol. 23, no. 3, pp. 610-617, 2007.

[9] H. Wang, Y. H. Liu and W. Chen, "Uncalibrated visual tracking control without visual velocity," *IEEE Trans. on Control Systems Technology,* to appear, 2010.

[10] K. Hashitmoto, K. Nagahama and T. Noritsugu, "A mode switching estimator for visual servoing", *Proc. of IEEE Int. Conf. on Robotics and Automation,* pp. 1610- 1615, 2002.

[11] W. E. Dixon, E. Zergeroglu, Y. Fang, and D. M. Dawson, Object tracking by a robot manipulator,: A robust cooperative visual servoing approach, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 211-216, 2002.

[12] K. Hashimoto and H. Kimura, "Visual servo with nonlinear observer," *Proc. of IEEE Int. Conf. on Robotic and Automation,* pp. 484-496, 1995.

[13] A. F. Filippov, "Differential equations with discontinuous right-hand sides," *Amer. Math. Soc. Translations*, vol. 42, pp. 199-231, 1964.

[14] M. M. Polycarpou and P. A. Ioannou, "On the existence and uniqueness of solutions in adaptive control system," *IEEE Trans. on Automatic Control,* vol. 38, no. 3, pp. 474-479, 1993.

[15] S. Gutman, "Uncertain dynamical systems – a Lyapunov min-max approach," *IEEE Trans. on Automatic Control*, vol. 24, no. 3, pp. 437-443, 1979.

[16] D. A Forsyth and J. Ponce, "Computer vision: a modern approach," Prentice Hall, 2003.

[17] J. J. Slotine and W. Li, "Applied nonlinear control," Prentice Hall, 1991.