Adaptive Tracking Control for Robots with Unknown Kinematic and Dynamic Properties

C. C. Cheah, C. Liu and J.J.E. Slotine

C.C. Cheah and C. Liu are with School of Electrical and Electronic Engineering, Nanyang Technological University, Block S1, Nanyang Avenue, S(639798), Republic of Singapore, Email: ECCCheah@ntu.edu.sg, J.J.E. Slotine is with Nonlinear System Laboratory, Massachusetts Institute of Technology, 77 Massachusetts Ave, Cambridge, MA 02139 USA, Email:JJS@mit.edu.

Abstract

It has been almost two decades since the first globally tracking convergent adaptive controllers were derived for robot with dynamic uncertainties. However, the problem of concurrent adaptation to both kinematic and dynamic uncertainties has never been systematically solved. This is the subject of this paper. We derive a new adaptive Jacobian controller for trajectory tracking of robot with uncertain kinematics and dynamics. It is shown that the robot end effector is able to converge to a desired trajectory with the uncertain kinematics and dynamics parameters being updated online by parameter update laws. The algorithm requires only to measure the end-effector position, besides the robot's joint angles and joint velocities. The proposed controller can also be extended to adaptive visual tracking control with uncertain camera parameters, taking into consideration the uncertainties of the nonlinear robot kinematics and dynamics. Experimental results are presented to illustrate the performance of the proposed controllers. In the experiments, we demonstrate that the robot's shadow can be used to control the robot.

Keywords: Adaptive Control; Tracking control; Adaptive Jacobian control; Visual Servoing.

I. Introduction

Humans do not have an accurate knowledge of the real world but are still able to act intelligently in it. For example, with the help of our eyes, we are able to pick up a large number of new tools or objects with different and unknown kinematic and dynamic properties, and manipulate them skillfully to accomplish a task. We can also grip a tool at different grasping points and orientations, and use it without any difficulty. Other examples include tennis and golf playing, and walking on stilts. In all cases, people seem to extend their self-perception to include the unknown tool as part of the body. In addition, humans can learn and adapt to the uncertainties from previous experience [1], [2]. For example, after using the unknown tool for a few times, we can manipulate it more skillfully. Recent research [3] also suggests that body shadows may form part of the approximate sensorimotor transformation. The way by which humans manipulate an unknown object easily and skillfully shows that we do not need an accurate knowledge of the kinematics and dynamics of the arms and object. The ability of sensing and responding to changes without an accurate knowledge of sensorimotor transformation [4] gives us a high degree of flexibility in dealing with unforseen changes in the real world.

The kinematics and dynamics of robot manipulators are highly nonlinear. While a precisely calibrated model-based robot controller may give good performance [5]-[7], the assumption of having exact models also means that the robot is not able to adapt to any changes and uncertainties in its models and environment. For example, when a robot picks up several tools of different dimensions, unknown orientations or gripping points, the overall kinematics and dynamics of the robot changes and are therefore difficult to derive exactly. Hence, even if the kinematics and dynamics parameters of the robot manipulator can be obtained with sufficiently accuracy by calibrations and identification techniques [8], [9], it is not flexible to do calibration or parameter identification for every object that a robot picks up, before manipulating it. It is also not possible for the robot to grip the tool at the same grasping point and orientation even if the same tool is used again. The behavior from human reaching movement show that we do not first identify unknown mass properties, grasping points and orientations of objects, and only then manipulate them. We can grasp and manipulate an object easily with unknown grasping point and orientation. The development of robot controllers that can similarly cope in a fluid fashion with uncertainties in both kinematics and dynamics is therefore an important step towards dexterous control of mechanical systems.

To deal with dynamic uncertainties, many robot adaptive controllers [11]-[31] have been proposed. A key point in adaptive control is that the tracking error will converge regardless of whether the trajectory is persistently exciting or not [11], [14]. That is, one does not need parameter convergence for task convergence. In addition, the overall stability and convergence of the combined on-line estimation/control (exploit/explore) process can also be systematically guaranteed. However, in these adaptive controllers, the kinematics of the robot is assumed to be known exactly.

Recently, several Approximate Jacobian setpoint controllers [32]-[35] have been proposed to overcome the uncertainties in both kinematics and dynamics. The proposed controllers do not require the exact knowledge of kinematics and Jacobian matrix. However, the results in [32]-[35] are focusing on setpoint control or point-to-point control of robot. The research on robot control with uncertain kinematics and dynamics is just at the beginning stage [36].

In this paper, we present an adaptive Jacobian controller for trajectory tracking control of robot manipulators. The proposed controller does not require exact knowledge of either kinematics or dynamics. The trajectory tracking control problem in the presence of kinematic and dynamic uncertainties is formulated and solved based on a Lyapunov-like analysis. By using sensory feedback of the robot end effector position, it is shown that the end effector is able to follow a desired trajectory with uncertainties in kinematics and dynamics. Novel adaptive laws, extending the capability of the standard adaptive algorithm [14] to deal with kinematics uncertainty, are proposed. A novel dynamics regressor using the estimated kinematics parameters is also proposed. The main new point is the adaptation to kinematic uncertainty in addition to dynamics uncertainty, which is something ”human-like” as in tool manipulation. This gives the robot a high degree of flexibility in dealing with unforseen changes and uncertainties in its kinematics and dynamics. The proposed controller can also be extended to adaptive visual tracking control with uncertain camera parameters, taking the nonlinearity and uncertainties of the robot kinematics and dynamics into consideration. A fundamental benefit of vision based control is to deal with uncertainties in models and much progress has been obtained in the literature of visual servoing (see [39]-[52] and references therein). Though image-based visual servoing techniques are known to be robust to modeling and calibration errors in practice, but as pointed out in [51], only a few theoretical results been obtained for the stability analysis in presence of the uncertain camera parameters [49]-[52]. In addition, these results are focusing on uncertainty in interaction matrix or image Jacobian matrix, and the effects of uncertain robot kinematics and dynamics are not considered. Hence, no theoretical result has been obtained for the stability analysis of visual tracking control with uncertainties in camera parameters, taking into consideration the uncertainties of the nonlinear robot kinematics and dynamics.

Section II formulates the robot dynamic equations and kinematics; Section III presents the adaptive Jacobian tracking controllers; Section IV presents some experimental results and shows that the robot's shadow can be used to control the robot; Section V offers brief concluding remarks.

II. Robot Dynamics and Kinematics

The equations of motion of robot with n degrees of freedom can be expressed in joint coordinates $q = [q_1, \cdots, q_n]^T \in R^n$ as [11], [30]:

$$M(q)\ddot{q} + (\frac{1}{2}\dot{M}(q) + S(q,\dot{q}))\dot{q} + g(q) = \tau \tag{1}$$

where $M(q) \in R^{n \times n}$ is the inertia matrix, $\tau \in R^n$ is the applied joint torque to the robot,

$$S(q,\dot{q})\dot{q} \;\; = \;\; \frac{1}{2}\dot{M}(q)\dot{q} - \frac{1}{2}\{\frac{\partial}{\partial q}\dot{q}^T M(q)\dot{q}\}^T$$

and $g(q) \in R^n$ is the gravitational force. Several important properties of the dynamic equation described by equation (1) are given as follows [11], [14], [30], [37]:

Property 1 The inertia matrix $M(q)$ is symmetric and uniformly positive definite for all $q \in R^n$.

Property 2 The matrix $S(q, \dot{q})$ is skew-symmetric so that $\nu^T S(q, \dot{q})\nu = 0$, for all $\nu \in R^n$.

Property 3 The dynamic model as described by equation (1) is linear in a set of physical parameters $\theta_d = (\theta_{d1}, \cdots, \ \theta_{dp})^T$ as

$$M(q)\ddot{q} + (\frac{1}{2}\dot{M}(q) + S(q, \dot{q}))\dot{q} + g(q) = Y_d(q, \dot{q}, \dot{q}, \ddot{q})\theta_d$$

where $Y_d(\cdot) \in R^{n \times p}$ is called the dynamic regressor matrix. $\diamondsuit$

In most applications of robot manipulators, a desired path for the end-effector is specified in task space, such as visual space or Cartesian space. Let $x \in R^n$ be a task space vector defined by [11], [32],

$$x = h(q)$$

where $h(\cdot) \in R^n \rightarrow R^n$ is generally a non-linear transformation describing the relation between joint space and task space. The task-space velocity $\dot{x}$ is related to joint-space velocity $\dot{q}$ as:

$$\dot{x} = J(q)\dot{q} \qquad (2)$$

where $J(q) \in R^{n \times n}$ is the Jacobian matrix from joint space to task space.

If cameras are used to monitor the position of the end-effector, the task space is defined as image space in pixels. Let r represents the position of the end-effector in Cartesian coordinates and x represents the vector of image feature parameters [39]. The image velocity vector $\dot{x}$ is related to the joint velocity vector $\dot{q}$ as [39]-[42],

$$\dot{x} = J_I(r)J_e(q)\dot{q} \qquad (3)$$

where $J_I(r)$ is the interaction matrix [41] or image Jacobian matrix [39], and $J_e(q)$ is the manipulator Jacobian matrix of the mapping from joint space to Cartesian space. In the presence of uncertainties, the exact Jacobian matrix cannot be obtained. If a position sensor is used to monitor the position of the end-effector, the task space is defined as Cartesian space and hence $J(q) = J_e(q)$.

A property of the kinematic equation described by equation (2) is stated as follows [56]:

Property 4 The right hand side of equation (2) is linear in a set of constant **kinematic parameters** $\theta_k = (\theta_{k1}, \cdots, \ \theta_{kq})^T$, such as link lengths, link twist angles. Hence, equation (2) can be expressed as,

$$\dot{x} = J(q)\dot{q} = Y_k(q, \dot{q})\theta_k \tag{4}$$

where $Y_k(q, \dot{q}) \in R^{n \times q}$ is called the kinematic regressor matrix. ♢

For illustration purpose, an example of a 2-link planar robot with a fixed camera configuration is given. The interaction matrix or image Jacobian matrix for the 2-link robot is given by

$$J_I = \frac{f}{z - f} \begin{bmatrix} \beta_1 & 0 \\ 0 & \beta_2 \end{bmatrix}, \tag{5}$$

where β_1, β_2 denote the scaling factors in pixels/m, z is the perpendicular distance between the robot and the camera, f is the focal length of the camera. The Jacobian matrix $J_m(q)$ from joint space to Cartesian space for the 2-link robot is given by:

$$J_m(q) = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix}, \tag{6}$$

where l_1, l_2 are the link lengths, q_1 and q_2 are the joint angles, $c_1 = \cos q_1$, $s_1 = \sin q_1$, $c_{12} = \cos(q_1 + q_2)$, $s_{12} = \sin(q_1 + q_2)$. The constants l_1, l_2, β_1, β_2, z, and f are all unknown.

The image space velocity $\dot{x}$ can be derived as:

$$\begin{aligned} \dot{x} = J_I J_m(q)\dot{q} &= \frac{f}{z - f} \begin{bmatrix} \beta_1 & 0 \\ 0 & \beta_2 \end{bmatrix} \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \\ &= \begin{bmatrix} -v_1 l_1 s_1 \dot{q}_1 - v_1 l_2 s_{12}(\dot{q}_1 + \dot{q}_2) \\ v_2 l_1 c_1 \dot{q}_1 + v_2 l_2 c_{12}(\dot{q}_1 + \dot{q}_2) \end{bmatrix} \end{aligned} \tag{7}$$

where $v_1 = \frac{f\beta_1}{z-f}, \ v_2 = \frac{f\beta_2}{z-f}$.

Hence $\dot{x} = J_I J_m(q)\dot{q}$ can be written into the product of a known regressor matrix $Y_k(q, \dot{q})$ and an unknown constant vector θ_k where

$$\dot{x} \;=\; \begin{bmatrix} -s_1\dot{q}_1 & -s_{12}(\dot{q}_1+\dot{q}_2) & 0 & 0 \\ 0 & 0 & c_1\dot{q}_1 & c_{12}(\dot{q}_1+\dot{q}_2) \end{bmatrix} \begin{bmatrix} v_1 l_1 \\ v_1 l_2 \\ v_2 l_1 \\ v_2 l_2 \end{bmatrix} = Y_k(q,\dot{q})\theta_k \tag{8}$$

Similar to most robot adaptive controllers, we consider the case where the unknown parameters are linearly parameterizable as in property 3 and property 4. If linear parameterization cannot be obtained due to presence of time varying parameters or unknown robot structure, adaptive control using basis functions [54], [55] is normally used. The basic idea is to approximate the models with unknown structure or time varying parameters, by a neural network where the unknown weights are adjusted online by the updated law (see [54], [55] for details).

III. Adaptive Jacobian Tracking Control

We now present our adaptive Jacobian tracking controller for robots with uncertain kinematics and dynamics. Tracking convergence is guaranteed by the combination of an adaptive control law of straightforward structure, an adaptation law for the dynamic parameters, and an adaptation law for the kinematic parameters. The main idea of the derivation is to introduce an adaptive sliding vector based on estimated task-space velocity, so that kinematic and dynamic adaptation can be performed concurrently.

In the presence of kinematic uncertainty, the parameters of the Jacobian matrix is uncertain and hence equation (4) can be expressed as

$$\hat{\dot{x}} = \hat{J}(q, \hat{\theta}_k)\dot{q} = Y_k(q, \dot{q})\hat{\theta}_k \tag{9}$$

where $\hat{\dot{x}} \in R^n$ denotes an estimated task-space velocity, $\hat{J}(q, \hat{\theta}_k) \in R^{n\times n}$ is an approximate Jacobian matrix and $\hat{\theta}_k \in R^q$ denotes a set of estimated kinematic parameters.

To illustrate the idea of adaptive Jacobian control, let us first consider the simpler setpoint control problem, and the controller

$$\tau = -\hat{J}^T(q, \hat{\theta}_k)K_p\Delta x - K_v\dot{q} + g(q)$$

where $\Delta x = x - x_d$, $x_d \in R^n$ is a desired position in task space, K_p and K_v are symmetric positive definite gain matrices, and $g(q)$ is known. The estimated kinematic parameter vector $\hat{\theta}_k$ of the approximate Jacobian matrix is updated by

$$\dot{\hat{\theta}}_k = L_k Y_k^T(q,\dot{q}) K_p \Delta x$$

where L_k is a symmetric positive definite gain matrix. Let us define a Lyapunov-like function candidate as

$$V = \tfrac{1}{2}\dot{q}^T M(q)\dot{q} + \tfrac{1}{2}\Delta\theta_k^T L_k^{-1}\Delta\theta_k + \tfrac{1}{2}\Delta x^T K_p \Delta x$$

where $\Delta\theta_k = \theta_k - \hat{\theta}_k$. Using the above controller and equation (1), the time derivative of V is

$$\dot{V} = -\dot{q}^T K_v \dot{q} + \dot{q}^T (J^T(q) - \hat{J}^T(q,\hat{\theta}_k)) K_p \Delta x - \Delta\theta_k^T Y_k^T(q,\dot{q}) K_p \Delta x = -\dot{q}^T K_v \dot{q} \leq 0$$

Since $\dot{V} = 0$ implies that $\dot{q} = 0$, points on the largest invariant set satisfy $\hat{J}^T(q,\hat{\theta}_k) K_p \Delta x = 0$. Hence, both $\dot{q}$ and $\hat{J}^T(q,\hat{\theta}_k) K_p \Delta x = 0$ tend to zero. In turn this implies that Δx converges to zero as long as $\hat{J}^T(q,\hat{\theta}_k)$ is of full rank.

The above controller is only effective for point to point control. In the following development, we present an adaptive Jacobian tracking controller with uncertain kinematics and dynamics. To avoid the need for measuring task-space velocity in adaptive Jacobian tracking control, we introduce a known signal y based on filtered differentiation of the measured position x,

$$\dot{y} + \lambda y = \lambda \dot{x} \tag{10}$$

The signal y can be computed by measuring x alone. With p the Laplace variable, y can be written from equations (4) and (10) as

$$y = \frac{\lambda p}{p+\lambda}\, x = W_k(t)\theta_k \tag{11}$$

where

$$W_k(t) = \frac{\lambda}{p+\lambda}\, Y_k(q,\dot{q})$$

with $y(0) = 0$ and $W_k(0) = 0$ since the robot usually starts from a rest position. Other linear filters may also be used based on noise or vibration models.

Let $x_d(t) \in R^n$ be the desired trajectory in task space. The algorithm we shall now derive is composed of (i) a control law

$$\tau = -\hat{J}^T(q,\hat{\theta}_k)(K_v \Delta\hat{\dot{x}} + K_p \Delta x) + \bar{Y}_d(q,\dot{q},\dot{q}_r,\hat{\ddot{q}}_r,\hat{\theta}_k)\hat{\theta}_d \tag{12}$$

where $\Delta x = x - x_d$, $\Delta\hat{\dot{x}} = \hat{\dot{x}} - \dot{x}_d$, $\bar{Y}_d(q, \dot{q}, \dot{q}_r, \hat{\ddot{q}}_r, \hat{\theta}_k)$ is a dynamic regressor matrix as detailed later and $\dot{q}_r$ and $\hat{\ddot{q}}_r$ are defined based on the adaptive sliding vector as detailed later, (ii) a dynamic adaptation law

$$\dot{\hat{\theta}}_d = -L_d\bar{Y}_d(q, \dot{q}, \dot{q}_r, \hat{\ddot{q}}_r, \hat{\theta}_k)s \tag{13}$$

and (iii) a kinematic adaptation law.

$$\dot{\hat{\theta}}_k = -L_k W_k^T(t)K_v(W_k(t)\hat{\theta}_k - y) + L_k Y_k^T(q, \dot{q})(K_p + \alpha K_v)\Delta x \tag{14}$$

All gain matrices are symmetric positive definite. Thus, while the expression of the controller and dynamic adaptation laws are straightforward extensions of standard results, the key novelties are that the algorithm is now augmented by a composite *kinematic adaptation law* (14), and that a specific choice of $\dot{q}_r$ is exploited throughout. In the proposed controller, x is measured from a position sensor. Many commercial sensors are available for measurement of x, such as vision systems, electromagnetic measurement systems, position sensitive detectors, or laser trackers.

Let us now detail the proof. First, define a vector $\dot{x}_r \in R^n$ as

$$\dot{x}_r = \dot{x}_d - \alpha\Delta x \tag{15}$$

Differentiating equation (15) with respect to time, we have

$$\ddot{x}_r = \ddot{x}_d - \alpha\Delta\dot{x} \tag{16}$$

where $\ddot{x}_d \in R^n$ is the desired acceleration in task space.

Next, define an **adaptive task-space sliding vector** using equation (9) as,

$$\hat{s}_x \;\; = \;\; \hat{\dot{x}} - \dot{x}_r = \hat{J}(q, \hat{\theta}_k)\dot{q} - \dot{x}_r \tag{17}$$

where $\hat{J}(q, \hat{\theta}_k)\dot{q} = Y_k(q, \dot{q})\hat{\theta}_k$ as indicated in equation (9). The above vector is adaptive in the sense that the parameters of the approximate Jacobian matrix is updated by the kinematic update law (14). Differentiating equation (17) with respect to time, we have,

$$\dot{\hat{s}}_x = \hat{\ddot{x}} - \ddot{x}_r = \hat{J}(q, \hat{\theta}_k)\ddot{q} + \dot{\hat{J}}(q, \hat{\theta}_k)\dot{q} - \ddot{x}_r \tag{18}$$

where $\hat{\ddot{x}}$ denotes the derivative of $\hat{\dot{x}}$. Next, let

$$\dot{q}_r = \hat{J}^{-1}(q, \hat{\theta}_k)\dot{x}_r \tag{19}$$

where $\hat{J}^{-1}(q,\hat{\theta}_k)$ is the inverse of the approximate Jacobian matrix $\hat{J}(q,\hat{\theta}_k)$. Since $\hat{J}^{-1}(q,\hat{\theta}_k)$ is a function of the estimated kinematic parameters $\hat{\theta}_k$, a standard parameter projection algorithm [58] can be adopted to keep the estimated kinematic parameters $\hat{\theta}_k$ remain in an appropriate region. We also assume that the robot is operating in a finite task space such that the approximate Jacobian matrix is of full rank. From equation (19), we have

$$\ddot{q}_r = \hat{J}^{-1}(q,\hat{\theta}_k)\ddot{x}_r + \dot{\hat{J}}^{-1}(q,\hat{\theta}_k)\dot{x}_r \tag{20}$$

where $\dot{\hat{J}}^{-1}(q,\hat{\theta}_k) = -\hat{J}^{-1}(q,\hat{\theta}_k)\dot{\hat{J}}(q,\hat{\theta}_k)\hat{J}^{-1}(q,\hat{\theta}_k)$. To eliminate the need of task-space velocity in $\ddot{q}_r$, we define

$$\hat{\ddot{q}}_r = \hat{J}^{-1}(q,\hat{\theta}_k)\hat{\ddot{x}}_r + \dot{\hat{J}}^{-1}(q,\hat{\theta}_k)\dot{x}_r \tag{21}$$

where

$$\hat{\ddot{x}}_r = \ddot{x}_d - \alpha\Delta\hat{\dot{x}} \tag{22}$$

From equations (22) and (16), we have

$$\hat{\ddot{x}}_r = \ddot{x}_d - \alpha\Delta\dot{x} + \alpha(\dot{x} - \hat{\dot{x}}) = \ddot{x}_r + \alpha(\dot{x} - \hat{\dot{x}}) \tag{23}$$

Substituting equation (23) into equation (21) and using equation (20) yields

$$\hat{\ddot{q}}_r = \ddot{q}_r + \alpha\hat{J}^{-1}(q,\hat{\theta}_k)(\dot{x} - \hat{\dot{x}}) = \ddot{q}_r - \alpha\dot{q} + \alpha\hat{J}^{-1}(q,\hat{\theta}_k)J(q)\dot{q} \tag{24}$$

Next, we define an adaptive sliding vector in joint space as,

$$\begin{aligned} s &= \dot{q} - \dot{q}_r = \hat{J}^{-1}(q,\hat{\theta}_k)((\hat{\dot{x}} - \dot{x}_d) + \alpha(x - x_d)) \\ &= \hat{J}^{-1}(q,\hat{\theta}_k)\hat{s}_x \end{aligned} \tag{25}$$

and

$$\dot{s} = \ddot{q} - \ddot{q}_r \tag{26}$$

Substituting $\ddot{q}_r$ from equation (24) into equation (26) yields

$$\dot{s} = \ddot{q} - (\hat{\ddot{q}}_r + \alpha\dot{q}) + \alpha\hat{J}^{-1}(q,\hat{\theta}_k)J(q)\dot{q} \tag{27}$$

Substituting equations (25) and (27) into equation (1), the equations of motion can be expressed as,

$$\begin{aligned} M(q)\dot{s} + (\tfrac{1}{2}\dot{M}(q) + S(q,\dot{q}))s + M(q)\hat{\ddot{q}}_r + (\tfrac{1}{2}\dot{M}(q) + S(q,\dot{q}))\dot{q}_r + g(q) \\ +\alpha M(q)\dot{q} - \alpha M(q)\hat{J}^{-1}(q,\hat{\theta}_k)J(q)\dot{q} = \tau \end{aligned} \tag{28}$$

The last six terms of equation (28) are linear in a set of **dynamics parameters** $\bar{\theta}_d$ and hence can be expressed as,

$$M(q)\hat{\ddot{q}}_r + (\tfrac{1}{2}\dot{M}(q) + S(q,\dot{q}))\dot{q}_r + g(q) + \alpha M(q)\dot{q} - \alpha M(q)\hat{J}^{-1}(q,\hat{\theta}_k)J(q)\dot{q}$$
$$= \bar{Y}_d(q,\dot{q},\dot{q}_r,\hat{\ddot{q}}_r,\hat{\theta}_k)\bar{\theta}_d \qquad (29)$$

so dynamics (28) can be written

$$M(q)\dot{s} + (\tfrac{1}{2}\dot{M}(q) + S(q,\dot{q}))s + \bar{Y}_d(q,\dot{q},\dot{q}_r,\hat{\ddot{q}}_r,\hat{\theta}_k)\bar{\theta}_d = \tau \qquad (30)$$

Consider now the adaptive control law (12), where $K_v \in R^{n\times n}$ and $K_p \in R^{n\times n}$ are symmetric positive definite matrices. The first term is an approximate Jacobian transpose feedback law of the task-space velocity and position errors, and the last term is an estimated dynamic compensation term based on equation (29). Update the estimated dynamic parameters $\hat{\theta}_d$ using (13), and the estimated kinematic parameters using (14), where L_k and L_d are symmetric positive definite matrices. The closed-loop dynamics is obtained by substituting (12) into (30),

$$M(q)\dot{s} + (\tfrac{1}{2}\dot{M}(q) + S(q,\dot{q}))s + \bar{Y}_d(q,\dot{q},\dot{q}_r,\hat{\ddot{q}}_r,\hat{\theta}_k)\Delta\theta_d$$
$$+\hat{J}^T(q,\hat{\theta}_k)(K_v\Delta\hat{\dot{x}} + K_p\Delta x) = 0 \qquad (31)$$

where $\Delta\theta_d = \bar{\theta}_d - \hat{\theta}_d$. The estimated kinematic parameters $\hat{\theta}_k$ of the approximate Jacobian matrix $\hat{J}(q,\hat{\theta}_k)$ is updated by the parameter update equation (14). Note that some kinematic parameters appear in the dynamics and are updated separately as the lumped dynamic parameters $\hat{\theta}_d$ using (13).

The linear parameterization of the kinematic parameters is obtained from equation (4). The estimated parameters $\hat{\theta}_k$ is then used in the inverse approximate Jacobian matrix $\hat{J}^{-1}(q,\hat{\theta}_k)$ and hence $\dot{q}_r$ and $\hat{\ddot{q}}_r$ in the dynamic regressor matrix. Note that $\hat{\theta}_k$ (like q and $\dot{q}$) is just part of the states of the adaptive control system and hence can be used in the control variables even if it is nonlinear in the variables (provided that a linear parameterization can be found else where in the system model i.e. equation (4)). Since $\hat{J}(q,\hat{\theta}_k)$ and its inverse $\hat{J}^{-1}(q,\hat{\theta}_k)$, are updated by q and $\hat{\theta}_k$, $\dot{\hat{J}}(q,\hat{\theta}_k)$ and $\dot{\hat{J}}^{-1}(q,\hat{\theta}_k) = -\hat{J}^{-1}(q,\hat{\theta}_k)\dot{\hat{J}}(q,\hat{\theta}_k)\hat{J}^{-1}(q,\hat{\theta}_k)$ are functions of q, $\dot{q}$, $\Delta\theta_k$ and Δx because $\dot{\hat{\theta}}_k$ is described by equation (14).

Let us define a Lyapunov-like function candidate as

$$V = \tfrac{1}{2}s^T M(q)s + \tfrac{1}{2}\Delta\theta_d^T L_d^{-1}\Delta\theta_d + \tfrac{1}{2}\Delta\theta_k^T L_k^{-1}\Delta\theta_k + \tfrac{1}{2}\Delta x^T(K_p + \alpha K_v)\Delta x \qquad (32)$$

where $\Delta\theta_k = \theta_k - \hat{\theta}_k$. Differentiating with respect to time and using *Property 1*, we have

$$\dot{V} = s^T M(q)\dot{s} + \frac{1}{2}s^T \dot{M}(q)s - \Delta\theta_d^T L_d^{-1}\dot{\hat{\theta}}_d - \Delta\theta_k^T L_k^{-1}\dot{\hat{\theta}}_k + \Delta x^T(K_p + \alpha K_v)\Delta\dot{x} \qquad (33)$$

Substituting $M(q)\dot{s}$ from equation (31), $\dot{\hat{\theta}}_k$ from equation (14) and $\dot{\hat{\theta}}_d$ from equation (13) into the above equation, using *Property 2*, equation (25) and equation (11), we have,

$$\begin{aligned}\dot{V} &= -\hat{s}_x^T K_v \Delta\dot{\hat{x}} - \hat{s}_x^T K_p \Delta x + \Delta x^T(K_p + \alpha K_v)\Delta\dot{x} \\ &\quad -\Delta\theta_k^T W_k^T(t) K_v W_k(t)\Delta\theta_k - \Delta\theta_k^T Y_k^T(q,\dot{q})(K_p + \alpha K_v)\Delta x \qquad (34)\end{aligned}$$

From equations (17), (4) and (15), we have

$$\hat{s}_x = \Delta\dot{\hat{x}} + \alpha\Delta x = \Delta\dot{x} + \alpha\Delta x - Y_k(q,\dot{q})\Delta\theta_k \qquad (35)$$

where

$$Y_k(q,\dot{q})\Delta\theta_k = J(q)\dot{q} - \hat{J}(q,\hat{\theta}_k)\dot{q} = \dot{x} - \dot{\hat{x}} \qquad (36)$$

Substituting $\Delta\dot{\hat{x}} = \Delta\dot{x} - Y_k(q,\dot{q})\Delta\theta_k$ and $\hat{s}_x = \Delta\dot{x} + \alpha\Delta x - Y_k(q,\dot{q})\Delta\theta_k$ into equation (34), we have

$$\begin{aligned}\dot{V} &= -\Delta\dot{x}^T K_v \Delta\dot{x} + 2\Delta\dot{x}^T K_v Y_k(q,\dot{q})\Delta\theta_k - \Delta\theta^T Y_k^T(q,\dot{q}) K_v Y_k(q,\dot{q})\Delta\theta_k \\ &\quad -\alpha\Delta x^T K_p \Delta x - \Delta\theta_k^T W_k^T(t) K_v W_k(t)\Delta\theta_k\end{aligned}$$

Since $\Delta\dot{\hat{x}} = \Delta\dot{x} - Y_k(q,\dot{q})\Delta\theta_k$, the above equation can be simplified to

$$\dot{V} = -\Delta\dot{\hat{x}}^T K_v \Delta\dot{\hat{x}} - \alpha\Delta x^T K_p \Delta x - \Delta\theta_k^T W_k^T(t) K_v W_k(t)\Delta\theta_k \qquad (37)$$

We are now in a position to state the following Theorem:

Theorem For a finite task space such that the approximate Jacobian matrix is non-singular, the adaptive Jacobian control law (12) and the parameter update laws (14) and (13) for the robot system (1) result in the convergence of position and velocity tracking errors. That is, $x - x_d \to 0$ and $\dot{x} - \dot{x}_d \to 0$, as $t \to \infty$. In addition, $W_k(t)\Delta\theta_k \to 0$ as $t \to \infty$.

Proof: Since $M(q)$ is uniformly positive definite, V in equation (32) is positive definite in s, Δx, $\Delta\theta_k$ and $\Delta\theta_d$. Since $\dot{V} \leq 0$, V is also bounded, and therefore s, Δx, $\Delta\theta_k$ and $\Delta\theta_d$ are bounded vectors. This implies that $\hat{\theta}_k$, $\hat{\theta}_d$ are bounded, x is bounded if x_d is bounded, and $\hat{s}_x = \hat{J}(q,\hat{\theta}_k)s$ is bounded as seen from equation (25). Using equation (35), we can conclude that $\Delta\hat{\dot{x}}$ is also bounded. Since Δx is bounded, $\dot{x}_r$ in equation (15) is also bounded if $\dot{x}_d$ is bounded. Therefore, $\dot{q}_r$ in equation (19) is also bounded if the inverse approximate Jacobian matrix is bounded. From equations (25), $\dot{q}$ is bounded and the boundedness of $\dot{q}$ means that $\dot{x}$ is bounded since the Jacobian matrix is bounded. Hence, $\Delta\dot{x}$ is bounded and $\ddot{x}_r$ in equation (16) is also bounded if $\ddot{x}_d$ is bounded. In addition, $\hat{\ddot{x}}_r$ in equation (22) is bounded since $\Delta\hat{\dot{x}}$ is bounded. From equation (14), $\dot{\hat{\theta}}_k$ is bounded since Δx, $\Delta\theta_k$, $\dot{q}$ are bounded and $Y_k(\cdot)$ is a trigonometric function of q. Therefore, $\hat{\ddot{q}}_r$ in equation (21) is bounded. From the closed-loop equation (31), we can conclude that $\dot{s}$ is bounded. The boundedness of $\dot{s}$ imply the boundedness of $\ddot{q}$ as seen from equation (27). From equation (18), $\dot{\hat{s}}_x$ is therefore bounded. Differentiating equation (35) with respect to time and re-arranging yields,

$$\Delta\hat{\ddot{x}} + \alpha\Delta\dot{x} = \dot{\hat{s}}_x$$

which means that $\Delta\hat{\ddot{x}} = \hat{\ddot{x}} - \ddot{x}_d$ is also bounded.

To apply Barbalat's lemma, let us check the uniform continuity of $\dot{V}$. Differentiating equation (37) with respect to time gives,

$$\ddot{V} = -2\Delta\hat{\dot{x}}^T K_v \Delta\hat{\ddot{x}} - 2\alpha\Delta x^T K_p \Delta\dot{x} - 2\Delta\theta_k^T W_k^T(t) K_v(\dot{W}_k(t)\Delta\theta_k - W_k(t)\dot{\hat{\theta}}_k)$$

where $W(t)$ and $\dot{W}(t)$ are bounded since $\dot{q}$, $\ddot{q}$ are bounded. This shows that $\ddot{V}$ is bounded since Δx, $\Delta\dot{x}$, $\Delta\hat{\dot{x}}$, $\Delta\hat{\ddot{x}}$, $\Delta\theta_k$, $\dot{\hat{\theta}}_k$ are all bounded. Hence, $\dot{V}$ is uniformly continuous. Using Barbalat's lemma, we have $\Delta x = x - x_d \to 0$, $\Delta\hat{\dot{x}} = \hat{\dot{x}} - \dot{x}_d \to 0$ and $W_k(q)\Delta\theta_k \to 0$ as $t \to \infty$. Finally, differentiating equation (35) with respect to time and re-arranging yields,

$$\Delta\ddot{x} + \alpha\Delta\dot{x} = \dot{\hat{s}}_x + \dot{Y}_k(q,\dot{q},\ddot{q})\Delta\theta_k - Y_k(q,\dot{q})\dot{\hat{\theta}}_k$$

which means that $\Delta\ddot{x} = \ddot{x} - \ddot{x}_d$ is also bounded. Since Δx and $\Delta\ddot{x}$ are bounded, we have $\Delta\dot{x} \to 0$ as $t \to \infty$. $\triangle\triangle\triangle$

Remark 1. If some of the kinematic parameters are known, they are not adapted upon but all the proofs still apply. For example, if the link parameters of the manipulator are known with sufficient accuracy, we can focus on the object parameters to save computation (unlike object parameters, link parameters are usually fixed). In this case, note that equation (4) is replaced by

$$\dot{x} = J(q)\dot{q} = Y_k(q,\dot{q})\theta_k + v(q,\dot{q})$$

where $v(q,\dot{q}) \in R^n$ is a known vector containing the known kinematic parameters. In some cases, we can simply put the known parameters into the known kinematic regressor $Y_k(q,\dot{q})$. Similarly, equation (9) can be expressed as

$$\hat{\dot{x}} = \hat{J}(q,\hat{\theta}_k)\dot{q} = Y_k(q,\dot{q})\hat{\theta}_k + v(q,\dot{q})$$

and hence

$$\hat{s}_x \;\; = \;\; Y_k(q,\dot{q})\hat{\theta}_k + v(q,\dot{q}) - \dot{x}_r = \Delta\dot{x} + \alpha\Delta x - Y_k(q,\dot{q})\Delta\theta_k.$$

In the case of the filtered differentiation of the measured position x, one can define

$$\dot{y} + \lambda y = \lambda(\dot{x} - v(q,\dot{q}))$$

and hence

$$y = W_k(t)\theta_k$$

where $W_k(t) = \frac{\lambda}{p+\lambda}\, Y_k(q,\dot{q})$.

For example, consider a 2-link robot holding an object with uncertain length l_o and grasping angle q_o, the velocity of the tool tip is given in Cartesian coordinates as [57]:

$$\begin{aligned}
\dot{x} \;&=\; \begin{bmatrix} -l_1 s_1 - l_2 s_{12} - l_o c_o s_{12} - l_o s_o c_{12} & -l_2 s_{12} - l_o c_o s_{12} - l_o s_o c_{12} \\ l_1 c_1 + l_2 c_{12} + l_o c_o c_{12} - l_o s_o s_{12} & l_2 c_{12} + l_o c_o c_{12} - l_o s_o s_{12} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \\
&=\; \begin{bmatrix} -(\dot{q}_1 + \dot{q}_2)s_{12} & -(\dot{q}_1 + \dot{q}_2)c_{12} \\ (\dot{q}_1 + \dot{q}_2)c_{12} & -(\dot{q}_1 + \dot{q}_2)s_{12} \end{bmatrix} \begin{bmatrix} l_o c_o \\ l_o s_o \end{bmatrix} + \begin{bmatrix} -l_1 s_1 \dot{q}_1 - l_2 s_{12}(\dot{q}_1 + \dot{q}_2) \\ l_1 c_1 \dot{q}_1 + l_2 c_{12}(\dot{q}_1 + \dot{q}_2) \end{bmatrix} \\
&=\; Y_k(q,\dot{q})\theta_k + v(q,\dot{q}) \qquad (38)
\end{aligned}$$

where $c_o = \cos(q_0),\ s_o = \sin(q_o)$.

Remark 2. A standard projection algorithm [58], [35] can be used to ensure that the estimated kinematic parameters $\hat{\theta}_k$ remain in an appropriate region so that the control signal $\dot{q}_r$ in equation (19) is defined for all $\hat{\theta}_k$ during adaptation. In additions, singularities often depend only on q, not $\hat{\theta}_k$. We assume that the robot is operating in a region such that the approximate Jacobian matrix is of full rank. Note from the adaptive Jacobian control law (12) and the dynamic parameter update law (14) that $\hat{J}^{-1}(q,\hat{\theta}_k)$ is used only in the definition of control variable $\dot{q}_r$ in equation (19). Therefore, we should be able to control this by bounding the variable or using a singularity-robust inverse of the approximate Jacobian matrix [38].

Remark 3. In the proposed controller, $W_k(t)\Delta\theta_k$ converges to zero. This implies parameter convergence in the case that the associated ”persistent excitation” (P.E.) conditions are satisfied.

Remark 4. In the redundant case, the null space of the approximate Jacobian matrix can be used to minimize a performance index [38], [24]. Following [24], equation (19) can be written as,

$$\dot{q}_r = \hat{J}^{+}(q,\hat{\theta}_k)\dot{x}_r + (I_n - \hat{J}^{+}(q,\hat{\theta}_k)\hat{J}(q,\hat{\theta}_k))\psi$$

where $\hat{J}^{+}(q,\hat{\theta}_k) = \hat{J}^{T}(q,\hat{\theta}_k)(\hat{J}(q,\hat{\theta}_k)\hat{J}^{T}(q,\hat{\theta}_k))^{-1}$ is the generalized inverse of the approximate Jacobian matrix, and $\psi \in R^n$ is minus the gradient of the convex function to be optimized. The above formulation is especially useful in application when x represents the position in the image space. This is because the image features is, in general, lesser than the number of degree of freedoms of robot. Hence, using the generalized inverse Jacobian matrix allows our results to be immediately applied to robots beyond two degrees of freedom.

Remark 5. From equation (35), the adaptive sliding vector can be expressed as:

$$\hat{s}_x = \Delta\dot{x} + \alpha\Delta x + Y_k(q,\dot{q})\hat{\theta}_k - Y_k(q,\dot{q})\theta_k \tag{39}$$

Hence, the sign of the parameter update laws in equations (14) and (13) are different because the last term in equation (12) is positive while the last term in equation (39) is negative.

Remark 6. As in [24], a computationally simpler implementation can be obtained by replacing definitions (19) and (20) by filtered signals as,

$$\ddot{q}_r + \lambda\dot{q}_r = \hat{J}^{-1}(q,\hat{\theta}_k)(\ddot{x}_r + \lambda\dot{x}_r - \dot{\hat{J}}(q,\hat{\theta}_k)\dot{q}_r)$$

with $\lambda > 0$. This implies that,

$$\frac{d}{dt}(\hat{J}(q,\hat{\theta}_k)\dot{q}_r) + \lambda\hat{J}(q,\hat{\theta}_k)\dot{q}_r = \ddot{x}_r + \lambda\dot{x}_r$$

so that $\hat{J}(q,\hat{\theta}_k)\dot{q}_r$ and its derivative tend to $\dot{x}_r$ and its derivative. In this case, $\ddot{q}_r$ may be used directly in the dynamic regressor.

Remark 7. The kinematic update law (14) can be modified as,

$$\begin{aligned}\hat{\theta}_k &= \bar{a}_k - PW_k^T(t)K_v(W_k(t)\hat{\theta}_k - y) + P\ Y_k^T(q,\dot{q})(K_p + \alpha K_v)\Delta x,\\ \dot{\bar{a}}_k &= -L_kW_k^T(t)K_v(W_k(t)\hat{\theta}_k - y) + L_kY_k^T(q,\dot{q})(K_p + \alpha K_v)\Delta x\end{aligned}$$

where P is a symmetric positive definite matrix. The adding of the ”proportional adaptation term” to the usual integral adaptation term typically makes the transients faster. In this case, the potential energy $\frac{1}{2}\Delta\theta_k^T L_k^{-1}\Delta\theta_k$ in the Lyapunov-like function candidate (32) should be replaced by an energy term $\frac{1}{2}(\theta_k - \bar{a}_k)^T L_k^{-1}(\theta_k - \bar{a}_k)$. That is,

$$V = \tfrac{1}{2}s^T M(q)s + \tfrac{1}{2}\Delta\theta_d^T L_d^{-1}\Delta\theta_d + \tfrac{1}{2}(\theta_k - \bar{a}_k)^T L_k^{-1}(\theta_k - \bar{a}_k) + \tfrac{1}{2}\Delta x^T(K_p + \alpha K_v)\Delta x$$

This add to $\dot{V}$ minus the square P-norm of $W_k^T(t)K_v(W_k(t)\hat{\theta}_k - y) + L_k Y_k^T(q,\dot{q})(K_p + \alpha K_v)\Delta x$. A similar argument can be applied to the dynamic parameters update law described by equation (13).

Remark 8. In general, the interaction or image Jacobian matrix in equation (3) can be linearly parameterized, except for depth information parameters in 3D visual servoing or position information parameters in fish-eye lenses. In practice, adaptive control is still effective in cases where the depth information is slowly time-varying. It is assumed that the desired endpoint position is defined in visual space when adapting to the interaction or image Jacobian matrix. If linear parameterization cannot be obtained, basis functions [54], [55] can be used to adaptively approximate the estimated image velocity. One interesting point to note is that image velocity or optical flow is not required in the visual tracking control algorithm.

Remark 9. In the approximate Jacobian setpoint controllers proposed in [32]-[34], it is shown that adaptation to kinematic parameters is not required for point-to-point control. Hence, the proposed controllers in [32]-[34] can deal with time varying uncertainties as far as setpoint control is concerned. In most visual servoing techniques, adaptation to camera parameters is also not required but the effects of the uncertainties of nonlinear robot kinematics and dynamics are not taken into consideration in the stability analysis. Hence it is not sure whether the stability can still be guaranteed in the presence of these uncertainties.

$\diamondsuit$

If a DC motor driven by an amplifier is used as actuator at each joint of the robot, the dynamics of the robot can be expressed as [11], [30]:

$$M(q)\ddot{q} + (\frac{1}{2}\dot{M}(q) + S(q,\dot{q}))\dot{q} + g(q) = Ku, \tag{40}$$

where $u \in R^n$ is either a voltage or current inputs to the amplifiers and $K \in R^{n\times n}$ is a diagonal transmission matrix that relates the actuator input u to the control torque τ. In actual implementations of the robot controllers, it is necessary to identify the exact parameters of matrix

K in equation (40). However, no model can be obtained exactly. In addition, K is temperature sensitive and hence may change as temperature varies due to overheating of motor or changes in ambient temperature. In the presence of uncertainty in K, position error may result and stability may not be guaranteed.

We propose an adaptive controller based on the approximate Jacobian matrix and an approximate transmission matrix $\hat{K}$ as,

$$u = \hat{K}^{-1}(-\hat{J}^T(q,\hat{\theta}_k)(K_v\Delta\dot{\hat{x}} + K_p\Delta x) + \bar{Y}_d(q,\dot{q},\dot{q}_r,\ddot{\hat{q}}_r,\hat{\theta}_k)\hat{\theta}_d + Y_a(\tau_o)\hat{\theta}_a) \tag{41}$$

$$\dot{\hat{\theta}}_a = -L_a Y_a(\tau_o)s, \tag{42}$$

where the kinematic parameter are updated by (14), the dynamic parameter are updated by (13), $L_a \in R^{n\times n}$ is a positive definite diagonal matrix, $\hat{\theta}_a \in R^n$ is an estimated parameter updated by the parameter update law (42), $Y_a(\tau_o) = diag\{-\tau_{o1}, -\tau_{o2}, \ldots, -\tau_{on}\}$ and τ_{oi} denotes the i^{th} element of the vector τ_o which is defined as

$$\tau_o = \hat{J}^T(q,\hat{\theta}_k)(K_v\Delta\dot{\hat{x}} + K_p\Delta x) - \bar{Y}_d(q,\dot{q},\dot{q}_r,\ddot{\hat{q}}_r,\hat{\theta}_k)\hat{\theta}_d \tag{43}$$

In the above controller, a constant $\hat{K}^{-1}$ is used to transform the control torque to an approximate actuator input and an additional adaptive input $Y_a(\tau_o)\hat{\theta}_a$ is added to compensate for the uncertainty introduced by the estimated transmission matrix $\hat{K}$.

Applying a similar argument as in the previous section on equation (40), and using equation (41), we have

$$\begin{aligned} &M(q)\dot{s} + (\tfrac{1}{2}\dot{M}(q) + S(q,\dot{q}))s + \bar{Y}_d(q,\dot{q},\dot{q}_r,\ddot{\hat{q}}_r,\hat{\theta}_k)\Delta\theta_d \\ &+\hat{J}^T(q,\hat{\theta}_k)(K_v\Delta\dot{\hat{x}} + K_p\Delta x) + (K\hat{K}^{-1} - I)\tau_o - K\hat{K}^{-1}Y_a(\tau_o)\hat{\theta}_a = 0, \end{aligned} \tag{44}$$

where τ_o is defined in equation (43). Since K, $\hat{K}$ and $Y_a(\tau_o)$ are diagonal matrices, the last two terms of equation (44) can be expressed as

$$(K\hat{K}^{-1} - I)\tau_o - K\hat{K}^{-1}Y_a(\tau_o)\hat{\theta}_a = Y_a(\tau_o)\Delta\bar{\theta}_a \tag{45}$$

where $\bar{\theta}_{ai} = 1 - \frac{k_i}{\hat{k}_i}$ and $k_i, \hat{k}_i$ are the i^{th} diagonal elements of $K, \hat{K}$ respectively, $\Delta\bar{\theta}_a = \bar{\theta}_a - K\hat{K}^{-1}\hat{\theta}_a$ and hence $\Delta\dot{\bar{\theta}}_a = -K\hat{K}^{-1}\dot{\hat{\theta}}_a$.

The proof follows a similar argument as in the proof of the Theorem by using a Lyapunov-like function candidate as

$$V_1 = V + \frac{1}{2}\Delta\bar{\theta}_a^T L_a^{-1}\hat{K}K^{-1}\Delta\bar{\theta}_a \tag{46}$$

where V is defined in equation (32). Hence, we have

$$\dot{V}_1 = -\Delta\hat{\dot{x}}^T K_v \Delta\hat{\dot{x}} - \alpha\Delta x^T K_p \Delta x - \Delta\theta_k^T W_k^T(t) K_v W_k(t) \Delta\theta_k \leq 0 \tag{47}$$

where we note that $\Delta\bar{\theta}_a$ is also bounded.

IV. Experiments

A series of experiments were conducted to illustrate the performance of the new adaptive Jacobian tracking controller.

A. *Experiment 1: using Shadow Feedback*

Recent psychophysical evidence by Pavani and Castiello [3] suggests that our brains respond to our shadows as if they were another part of the body. This imply that body shadows may form part of the approximate sensory-to-motor transformation of the Human motor control system. In this section, we implement the proposed adaptive Jacobian controller on the first two joints of an industrial robot and show that robot's shadow can be used to control the robot. The experimental setup consists of a camera, a light source and a SONY SCARA robot as shown in figure 1. An object is attached to second joint of the robot and is parallel to the second link. A robot's shadow is created using the light source and projected onto a white screen. The camera is located under the screen and the tip of the object's shadow is monitored by the camera (see figure 1).

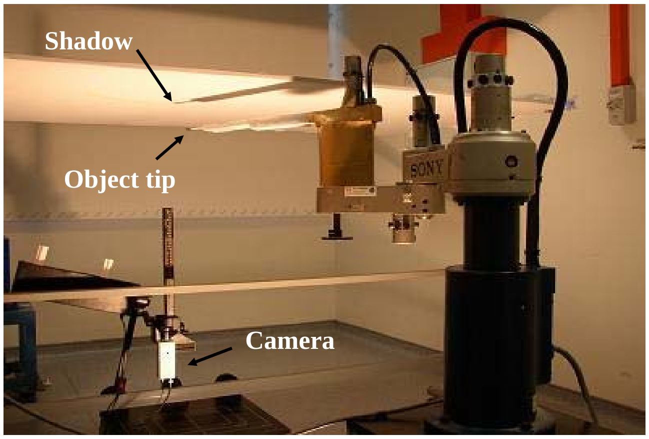

Fig. 1. A SONY Robot with its Shadow

The robot's shadow is required follow a straight line starting from the initial position $(X_0, Y_0) = (153, 73)$ to the final position $(X_f, Y_f) = (75, 185)$ specified in image space (pixels). The desired trajectory (X_d, Y_d) for the robot's shadow is hence specified as

$$Y_d = mX_d + c$$

where

$$X_d = \begin{cases} X_0 - 6d(\frac{t^2}{2T^2} - \frac{t^3}{3T^3}) & \text{for } 0 \leq t \leq T \\ X_f & \text{for } T < t \leq T_f \end{cases}$$

and $m = -1.448$, $c = 294.65$, $d = 78$ pixels , $T = 5$ sec and $T_f = 6$ sec.

To illustrate the idea we discussed in remark 1, we first assume that the lengths of the robot links were sufficiently accurate in this experiment. Experiments with uncertain link parameters will be presented in the next subsection. Hence only the object parameters were updated. The object was placed very closed to the white screen in order to cast a sharp shadow onto the screen. Therefore, the unknown mapping from the shadow to object is just a scalar in this experiment. The length of the object was initially estimated as 0.5 m. The experiment was performed with $L_k = 0.03I$, $L_d = 0.0005I$, $K_v = diag\{0.03, 0.029\}$, $K_p = diag\{0.175, 0.13\}$, $\alpha = 2$, $\lambda = 200\pi$. A sequence of the images capturing the motion of the robot's shadow are presented in figure 2 and a video of the results is shown in Extension 1. The shadow started from an initial position as shown in figure 2(a), followed the specified straight line and stopped in an end point as shown in figure 2(f). The maximum tracking error of the experiments was about $4.2\ mm$. As seen from the results, the robot's shadow is able to follow the straight line closely. Note that the shadow experiment is also similar to using a finger with an overhead projector to point at a specific equation for instance.

B. Experiment 2: using Position Feedback

Next, we implemented the proposed controllers on a 2-link direct-drive robot as shown in figure 3. The masses of the first and second links are approximately equal to $1.6kg$ and $1kg$ respectively, and the masses of the first and second motors are approximately equal to $9.5kg$ and $3kg$ respectively. The lengths of the first and second links are approximately equal to $l_1 = 0.31m$ and $l_2 = 0.3m$ respectively. The robot is holding an object with an length of $0.10m$ and a grasping angle of 60^o. A PSD camera (position sensitive detector) manufactured by Hamamatsu is used to measure the position of the robot end effector.

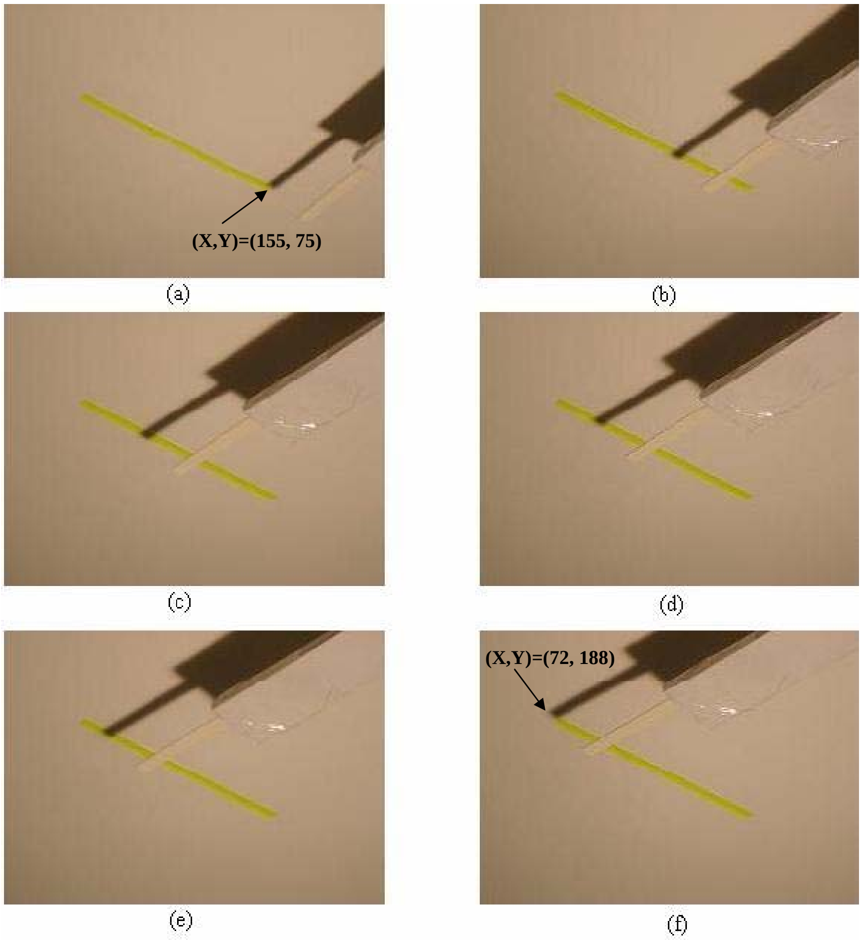

Fig. 2. Experimental results showing robot's shadow following a line

The robot is required to hold an object with uncertain length and grasping angle and follow a circular trajectory specified in Cartesian space as:

$$
\begin{aligned}
X_d &= 0.33 + 0.1\sin(0.54 + 3t) \\
Y_d &= 0.41 + 0.1\cos(0.54 + 3t)
\end{aligned}
$$

In this experiment, uncertainties in both robot parameters and object parameters were also considered. The link lengths were estimated as $\hat{l}_1(0) = 0.25m$, $\hat{l}_2(0) = 0.27m$ and the object length and grasping angle were estimated as 0.12 m and 20^o respectively. The initial position of the robot end effector was specified as $(X(0), Y(0)) = (0.28, 0.52)$. Experimental results with $L_k = diag\{0.04, 0.045, 0.015\}$, $L_d = diag\{0.01, 0.002, 0.002, 0.002, 0.015, 0.01, 0.01\}$, $K_v = diag\{2, 2\}$, $K_p = diag\{450, 450\}$, $\alpha = 1.2$, $\lambda = 200\pi$ are presented in figures 4 and 5. The

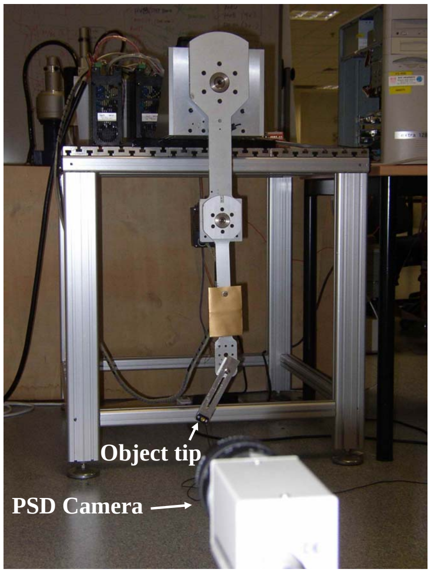

Fig. 3. A 2-link Direct-Drive Robot

transient response is shown in figure 6. As seen from the results, the tracking errors converge with updating of the estimated kinematic and dynamic parameters.

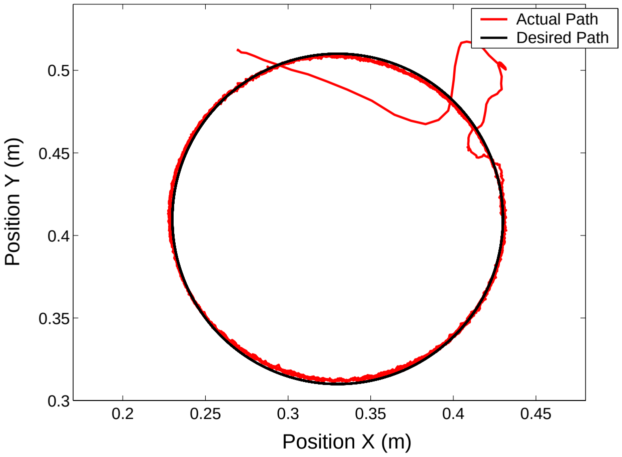

Fig. 4. Path of the end effector

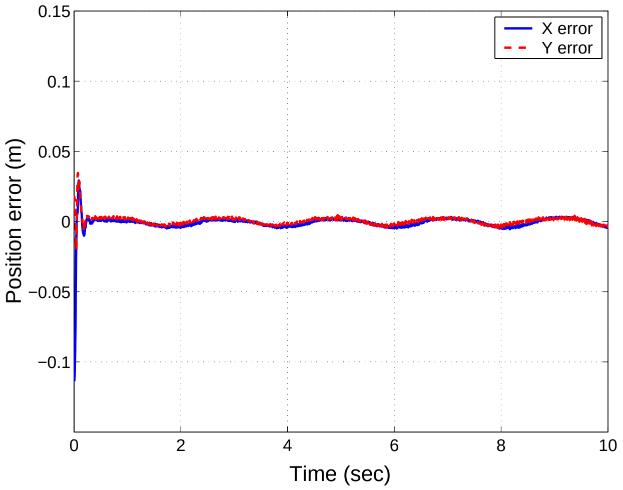

Fig. 5. Position Errors

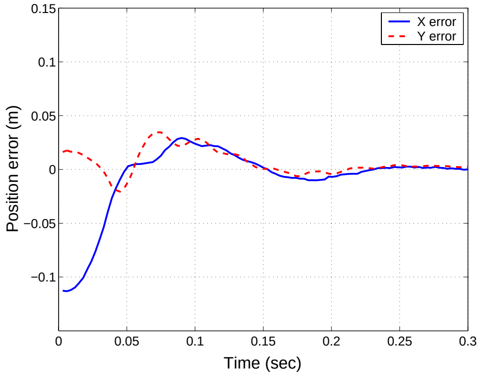

Fig. 6. Position Errors (Transient)

In the next experiments, a proportional term is added to the kinematic update law (see Remark 7). The experimental results in figure 7 and figure 8 show that the tracking errors converge and the transient response is shown in figure 9. We used $L_k = diag\{0.075, 0.105, 0.025\}$, $P = diag\{0.00018, 0.0002, 0.0001\}$, with the rest of the control gains remaining the same.

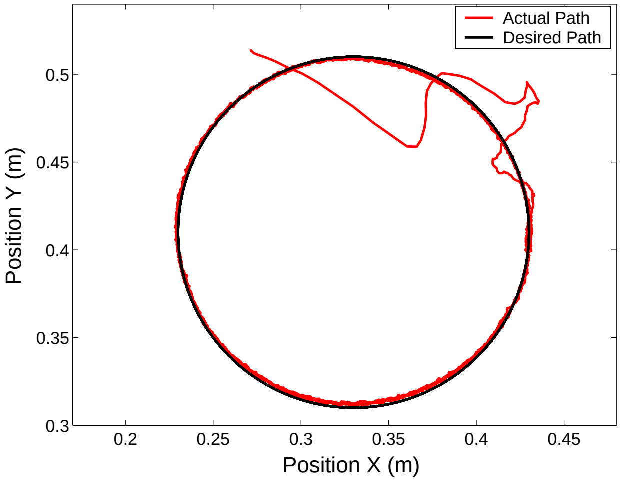

Fig. 7. Path of the end effector

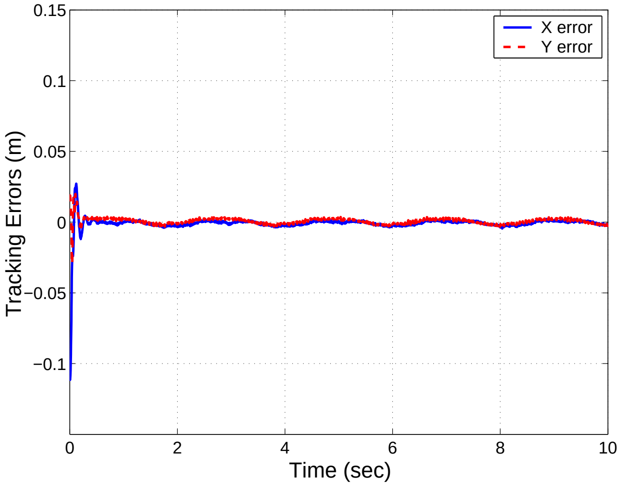

Fig. 8. Position Errors

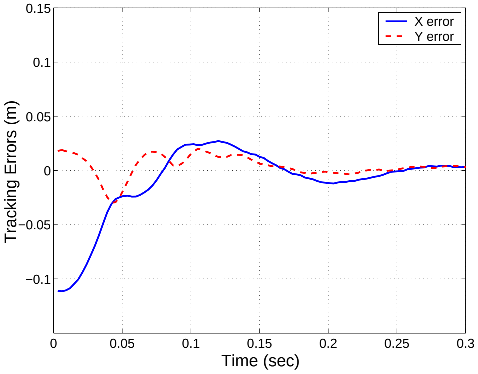

Fig. 9. Position Errors (Transient)

V. Concluding Remarks

We have proposed an adaptive Jacobian controller for robot tracking control with uncertain kinematics and dynamics. A novel update law is introduced to update uncertain kinematics parameters, using sensory feedback of the robot end effector position. The robot end effector is able to track a desired trajectory with the uncertain kinematics and dynamics parameters being updated online. Experimental results illustrate the performance of the proposed controllers. The experiments also show that a robot can be controlled using its shadow. As pointed in [36], the research on robot control with uncertain kinematics and dynamics is just at the beginning stage [36]. Future works would be devoted to extending the adaptive Jacobian controller to force tracking control and object manipulation by robot hand with soft tips. In these control problems, the Jacobian matrices are uncertain. For example, the constraint Jacobian is uncertain in presence of uncertainty in the constraint surface; the contact points of the robot fingers with soft tips are also difficult to estimate exactly since they are changing during manipulation. Due to the depressions at the soft contact points, the kinematics of the fingers also become uncertain. It is also interesting to investigate the applicability of the proposed adaptive Jacobian control theory to the study of internal model in sensorimotor integration [59], [60].

References

[1] M.A. Arbib, N. Schweighofer, and W.T. Thach, ”Modeling the Cerebellum: From Adaptation to Coordination”, *Motor Control and Sensory-Motor Integration: Issues and Directions*, (D.J. Glencross and J.P. Piek, Eds.), pp 11-36, Elsevier Science, 1995.

[2] K. Sekiyama, S. Miyauchi, T. Imaruoka, H. Egusa, T. Tashiro, ”Body image as a visuomotor transformation device revealed in adaptation to reversed vision”, *Nature*, vol. 407, pp 374-377, 2000.

[3] F. Pavani and U. Castiello, ”Binding personal and extrapersonal space through body shadows”, *Nature Neuroscience*, vol. 7, pp. 14 - 16, 2004.

[4] A. Pouget, and L.H. Snyder, ”Computational approaches to sensorimotor transformations”, *Nature Neuroscience*, vol. 3, pp 1192-1198, 2000.

[5] J. M. Hollerbach, “A recursive lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity,” *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 10, 1980.

[6] J. Y. S. Luh, M. H. Walker, and R. P. Paul, “On-line computational scheme for mechanical manipulator,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 102, pp. 69–76, 1980.

[7] J. J. Craig, *Introduction to Robotics*. New York: Addison-Wesley, 1986.

[8] C.H. An, C. G. Atkeson, J.M. Hollerbach, ”Model-based control of a robot manipulator”, Cambridge, Mass.: MIT Press, 1988.

[9] J.M. Renders, E. Rossignol, M. Becquet, and R. Hanus, ”Kinematic calibration and geometrical parameter identification for robots”, *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 6, pp 721 - 732, 1991.

[10] M. Takegaki and S. Arimoto, “A new feedback method for dynamic control of manipulators,” *ASME J. of Dynamic Systems, Measurement and Control*, vol. 102, pp. 119–125, 1981.

[11] S. Arimoto, *Control Theory of Nonlinear Mechanical Systems - A Passivity-Based and Circuit-Theoretic Approach*. Oxford: Clarendon Press, 1996.

[12] J. J. Craig, P. Hsu, and S. S. Sastry, “Adaptive control of mechanical manipulators,” *Int. J. Robotics Research*, vol. 6, no. 2, pp. 16–28, 1987.

[13] J. J. Craig, *Adaptive Control of Mechanical Manipulators*. New York: Addison-Wesley, 1988.

[14] J. J. E. Slotine and W. Li, “On the adaptive control of robot manipulators,” *Int. J. Robotics Research*, no. 6, pp. 49–59, 1987.

[15] J. J. E. Slotine and W. Li, “Adaptive manipulator control: A case study,” *IEEE Trans. on Automatic Control*, vol. 33, no. 11, pp. 995–1003, 1988.

[16] J. J. E. Slotine and W. Li, ”Adaptive strategies in constrained manipulation”, in *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 595 – 601, 1987.

[17] R. H. Middleton and G. C. Goodwin, “Adaptive computed torque control for rigid link manipulators,” *Systems and Control Letter*, vol. 10, pp. 9–16, 1988.

[18] D. E. Koditschek, “Adaptive techniques for mechanical systems,” in *Fifth Yale Workshop on Applications of Adaptive Systems Theory*, (New Haven, CT), pp. 259–265, 1987.

[19] J. T. Wen and D. Bayard, “New class of control laws for robotic manipulators -part 2. adaptive case,” *International Journal of Control*, vol. 47, no. 5, pp. 1387–1406, 1988.

[20] B. Paden and R. Panja, “A globally asymptotically stable ’PD+’ controller for robot manipulator,” *International Journal of Control*, vol. 47, no. 6, pp. 1697–1712, 1988.

[21] R. Kelly and R. Carelli, “Unified approach to adaptive control of robotic manipulators,” in *Proc. 27th IEEE Conf. on Decision and Control*, 1988.

[22] R. Ortega and M. W. Spong, “Adaptive motion control of rigid robots: a tutorial,” *Automatica*, vol. 25, no. 6, pp. 877–888, 1989.

[23] N. Sadegh and R. Horowitz, “Stability and robustness analysis of a class of adaptive controllers for robotic manipulators,” *International Journal of Robotics Research*, vol. 9, no. 3, pp. 74–92, 1990.

[24] G. Niemeyer and J.J.E. Slotine, ”Performance in Adaptive Manipulator Control,” *Int. J. Robotics Research*, vol. 10, no. 2, 1991.

[25] H. Berghuis, R. Ortega and H. Nijmeijer, ”A robust adaptive robot controller”, *IEEE Transactions on Robotics and Automation*, vol. 9, no. 6, pp 825 - 830, 1993.

[26] L. L. Whitcomb, A. Rizzi, and D.E. Koditschek, ”Comparative Experiments with a New Adaptive Controller for Robot Arms”, *IEEE Transactions on Robotics and Automation*, vol. 9, no. 1, pp 59-70, 1993.

[27] L. L. Whitcomb, S. Arimoto, T. Naniwa, and F. Ozaki, ”Experiments in Adaptive Model-Based Robot Force Control”, *IEEE Control Systems Magazine*, vol. 16, no. 1, pp 49-57, 1996.

[28] K.W. Lee and H. Khalil, ”Adaptive output feedback control of robot manipulators using high gain observer”, *Int. J. Control*, vol. 67, no. 6, 1997

[29] P. Tomei, “Robust adaptive friction compensation for tracking control of robot manipulators,” *IEEE Transactions on Automatic Control*, vol. 45, no. 11, pp. 2164 – 2169, 2000.

[30] F. L. Lewis, C. T. Abdallah, and D. M. Dawson, *Control of Robot Manipulators*. New York: Macmillan Publishing Company, 1993.

[31] L. Sciavicco and B. Siciliano, *Modelling and control of robot manipulators*. New York: Springer-Verlag, 2000.

[32] C. C. Cheah, S. Kawamura, and S. Arimoto, “Feedback control for robotic manipulators with an uncertain jacobian matrix,” *Journal of Robotic System*, vol. 12, no. 2, pp. 119–134, 1999.

[33] H. Yazarel and C. C. Cheah, “Task-space adaptive control of robotic manipulators with uncertainties in gravity regressor matrix and kinematics,” *IEEE Trans. on Automatic Control*, vol. 47, no. 9, pp. 1580 – 1585, 2002.

[34] C. C. Cheah, M. Hirano, S. Kawamura, and S. Arimoto, “Approximate jacobian control for robots with uncertain kinematics and dynamics,” *IEEE Trans. on Robotics and Automation*, vol. 19, no. 4, pp. 692–702, 2003.

[35] W. E. Dixon, “Adaptive regulation of amplitude limited robot manipulators with uncertain kinematics and dynamics,” in *Proc. of American Control Conference*, (Boston, USA), pp. 3939–3844, 2004.

[36] S. Arimoto, “Robotics research toward explication of everyday physics,” *International Journal of Robotic Research*, vol. 18, no. 11, pp. 1056–1063, 1999.

[37] J. J. E. Slotine and W. Li, *Applied nonlinear control*. Englewood Cliffs, New Jersey: Prentice Hall, 1991.

[38] Y. Nakamura, *Advanced Robotics*. Reading MA: Addison-Wesleyn, 1985.

[39] G. H. S. Hutchinson and P. Corke, “A tutorial on visual servo control,” *IEEE Trans. on Robotics and Automation*, vol. 12, no. 5, pp. 651 – 670, 1996.

[40] L. E. Weiss, A. C. Sanderson and C. P. Neuman, “Dynamic sensor-based control of robots with visual feedback,” *IEEE Trans. on Robotics and Automation*, vol. RA-3, no. 5, pp. 404 – 417, 1987.

[41] B. Espiau, F. Chaumette and P. Rives, ”A new approach to visual servoing in robotics”, *IEEE Trans. on Robotics and Automation*, Vol. 8 , No. 3, pp 313 - 326, 1992.

[42] N.P. Papanikolopoulos, P.K. Khosla, and T. Kanade, ”Visual tracking of a moving target by a camera mounted on a robot: a combination of control and vision”, *IEEE Trans. on Robotics and Automation*, Vol. 9 , No. 1, pp 14 - 35, 1993.

[43] N.P. Papanikolopoulos and P.K. Khosla, ”Adaptive robotic visual tracking: theory and experiments”, *IEEE Trans. on Automatic Control*, Vol. 38, No. 3, 429 - 445, 1993.

[44] M. Jgersand, O. Fuentes, R. Nelson, ”Acquiring Visual-Motor Models for Precision Manipulation with Robot Hands”, in *Proc of 4th European Conference on Computer Vision*,pp 603-612, 1996.

[45] E. Malis, F. Chaumette and S. Boudet, ”2D 1/2 visual servoing”, *IEEE Transaction on Robotics and Automation*, Vol. 15, No. 2, pp 234-246, April 1999.

[46] E. Malis and F. Chaumette, ”Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods”, *IEEE Transaction on Robotics and Automation*, Vol. 18, No. 2, pp 176-186, April 2002.

[47] K. Miura, K. Hashimoto, J. Gangloff, and M. de Mathelin, ”Visual servoing without Jacobian using modified simplex optimization”, in *Proc. of IEEE International Conference on Robotics and Automation*, pp 3515-3529, (Barcelona, Spain), 2005.

[48] N.R. Gans, G. H. S. Hutchinson, P. Corke, ”Performance tests for visual servo control systems, with application to partitioned approaches to visual servo control”, *Int. J. Robotics Research*, vol.10, no. 11,pp 955-981, 2003.

[49] B. Espiau, Effect of camera calibration errors on visual servoing in robotics, in *Proc. of International Symposium on Experimental Robotics*, (Kyoto, Japan),r 1993.

[50] L. Deng, F. Janabi-Sharifi, and W. J. Wilson, Stability and robustness of visual servoing methods, in *Proc. of IEEE International Conference on Robotics and Automation*, pp. 16051609, (Washington D.C.), 2002.

[51] E. Malis and P. Rives, ”Robustness of Image-Based Visual Servoing with Respect to Depth Distribution Errors”, in *Proc. of IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, September 2003.

[52] E. Malis, ”Visual servoing invariant to changes in camera intrinsic parameters”, *IEEE Transaction on Robotics and Automation*, Vol. 20, No. 1, pp 72-81, 2004.

[53] W. Li, ”Adaptive control of robot manipulators”, *Doctoral Thesis, MIT Dept. of Mechanical Engineering*, June 1990.

[54] R. Sanner and J.J.E. Slotine, ”Gaussian Networks for Direct Adaptive Control,” IEEE. Trans. on Neural Networks, 3(6), 1992.

[55] F.L. Lewis, ”Neural network control of robot manipulators”, *IEEE Expert: Intelligent Systems and Their Applications*, vol. 11, no. 3, pp. 64-75, June 1996.

[56] C. C. Cheah, C. Liu and J.J.E. Slotine, ”Approximate Jacobian Adaptive Control for Robot Manipulators”, in *Proc. of IEEE Int. Conference on Robotics and Automation*, (New Orleans, USA), pp 3075-3080, 2004.

[57] C. C. Cheah, C. Liu and J.J.E. Slotine, ”Experiments on Adaptive Control of Robots with Uncertain Kinematics and Dynamics”, in *Proc. of Int. Symposium on Experimental Robotics*, (Singapore).

[58] P. Ioannou and J. Sun, ’Robust Adaptive Control’ Prentice Hall, 1996.

[59] C. Tin and C.S. Poon, ”Internal models in sensorimotor integration: Perspectives from adaptive control theory”, *Journal of Neural Engineering*, Vol 2, no. 3, pp. S147-S163, 2005.

[60] H. Imamizu, Y. Uno and M. Kawato, ”Adaptive Internal Model of Intrinsic Kinematics Involved in Learning an Aiming Task”, *Journal of Experimental Psychology: Human Perception and Performance*, Vol. 24, No. 3, pp. 812-829, 1998.

APPENDIX A: INDEX TO MULTIMEDIA EXTENSION

The multimedia extensions to this article are at: http://www.ijrr.org.

Extension	Type	Description
1	Video	Experimental results of robot tracking control using shadow's feedback.