

MIT Open Access Articles

*Adaptive Vision and Force Tracking Control
for Robots With Constraint Uncertainty*

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

Citation: Chien Chern Cheah et al. "Adaptive Vision and Force Tracking Control for Robots With Constraint Uncertainty." IEEE/ASME Transactions on Mechatronics 15.3 (2010): 389–399. Web. 4 Apr. 2012. © 2010 Institute of Electrical and Electronics Engineers

As Published: <http://dx.doi.org/10.1109/tmech.2009.2027115>

Publisher: Institute of Electrical and Electronics Engineers (IEEE)

Persistent URL: <http://hdl.handle.net/1721.1/69937>

Version: Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

Terms of Use: Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.



Adaptive Vision and Force Tracking Control for Robots With Constraint Uncertainty

Chien Chern Cheah, *Senior Member, IEEE*, Saing Paul Hou, *Student Member, IEEE*, Yu Zhao, and Jean-Jacques E. Slotine

Abstract—In force control applications of robots, it is difficult to obtain an exact model of a constraint surface. In presence of the constraint uncertainty, the robot needs to adapt to the uncertainty in external parameters due to the environment, in addition to the uncertainties in internal parameters of the robot kinematics and dynamics. In this paper, a visually servoed adaptive controller is proposed for motion and force tracking with uncertainties in the constraint surface, kinematics, dynamics, and camera model. We shall show that the robot can track the desired trajectories with the uncertain internal and external parameters updated online. This gives the robot a high degree of flexibility in dealing with changes and uncertainties in its model and the environment.

Index Terms—Adaptive control, force control, uncertain kinematics and dynamics, visual servoing.

I. INTRODUCTION

IN MANY applications of robots, such as spot welding, polishing, and deburring, the end-effector is required to make contact with an environment. In these applications, it is necessary to control not only the position but also the interaction force between the robot end-effector and the environment. There are two major approaches for robot force control, namely impedance control [1] and hybrid position/force control [2]. A review for the research in force control can be found in [3]. Several model-based approaches have been proposed for motion and force tracking control [4], [5] of robots, but these controllers require exact models of the robot systems that are difficult to obtain in practice. To relieve this problem, much effort has been devoted to understand how the robots can cope with dynamic uncertainty [6]–[9].

A major problem for most hybrid position/force controllers is that the exact knowledge of the constraint surface is required. However, in force control applications, it is difficult to obtain the exact model of the constraint. Several controllers [10], [11] have been proposed to deal with constraint uncertainty, but the robot kinematics is assumed to be known exactly. In addition, the control problems are either formulated in joint space or

Cartesian space. It is important to note that when the constraint surface is uncertain, the desired end-effector position on the constraint surface is unknown and, hence, cannot be obtained directly in Cartesian space or joint space.

A fundamental benefit of image-based control is that it allows noncontact measurement of the environment, so the position of the end-effector and its desired position on the uncertain constraint surface can be obtained in image space using cameras. Most research so far on vision control of robot manipulators has focused on free motion control. To deal with contact tasks, several hybrid vision–force controllers have been proposed [12]–[15]. However, these approaches have ignored the effects of nonlinearity and uncertainties of the robot dynamics in the design of the vision and force controllers. Therefore, it is not sure whether the stability of the overall control system can be guaranteed in the presence of dynamic, kinematic, and constraint uncertainties. A computed torque method is proposed in [16], but the dynamics and kinematics of the robot are assumed to be known exactly and the stability issue is not considered. This means that the robot is not able to adapt to any changes and uncertainties in its models. 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 change and are therefore difficult to derive exactly.

Arimoto [17] describes the importance of the research on robot control with uncertain kinematics. Several controllers [18]–[21] have been proposed to deal with uncertain kinematics and dynamics, but the results are focusing on free motion control. In force control problems [3], it is necessary to control not only the position but also the interaction force between the robot end-effector and the constraint. Recently, several position and force controllers [22]–[24] using approximate Jacobian have been proposed to overcome the uncertainties in both kinematics and dynamics. However, the results are limited to set point control, and the trajectory tracking control problem with uncertain constraint surface, kinematics, and dynamics has not been solved so far.

In this paper, we propose a visually servoed motion and force tracking controller, in which exact knowledge of the constraint surface, kinematics, dynamics, and camera model is not required. The use of a vision sensor introduces additional transformation and uncertainty from Cartesian space to image space, and hence, the motion and force errors are defined in two different coordinate frames. Adaptive Jacobian matrices are used to compensate the uncertainties due to internal and external parameters. A Lyapunov-like function is presented to prove the stability of the proposed vision–force controller. It is shown that

Manuscript received June 27, 2008; revised December 5, 2008 and June 19, 2009. First published August 18, 2009; current version published April 2, 2010. Recommended by Technical Editor G. Morel.

C. C. Cheah and S. P. Hou are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore S639798 (e-mail: ecccheah@ntu.edu.sg).

Y. Zhao was with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore S639798. He is now with OGS Industries, Singapore 189268.

J.-J. E. Slotine is with the Nonlinear Systems Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TMECH.2009.2027115

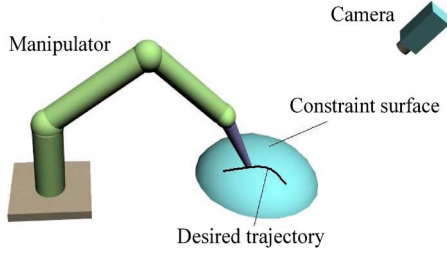


Fig. 1. Camera–manipulator system.

the proposed controller can track desired vision and force trajectories with the uncertainties in constraint surface, kinematics, dynamics, and camera model. Simulation results illustrate the effectiveness of the proposed controller.

II. ROBOT DYNAMICS AND KINEMATICS

We consider a force control system consisting of a robot manipulator and a camera (s) fixed in the work space, as shown in Fig. 1. In this system, the end-effector is in contact with a constraint surface.

First, let $r \in \mathfrak{R}^m$ denote a position of the end-effector in Cartesian space as [10], [19], [25], [26]

$$r = h(q) \quad (1)$$

where $h(\cdot) \in \mathfrak{R}^n \rightarrow \mathfrak{R}^m$ is generally a nonlinear transformation describing the relation between joint space and task space, and $q = [q_1, \dots, q_n]^T \in \mathfrak{R}^n$ is a vector of joint angles of the manipulator. The velocity of the end-effector \dot{r} is related to joint-space velocity \dot{q} as

$$\dot{r} = J_m(q)\dot{q} \quad (2)$$

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

For a visually servoed system, cameras are used to observe the position of the end-effector in image space. The mapping from Cartesian space to image space requires a camera-lens model in order to represent the projection of task objects onto the charge-coupled device (CCD) image plane. Let $x \in \mathfrak{R}^m$ denote a vector of image feature parameters and \dot{x} the corresponding vector of image feature parameter rates of change. The relationship between Cartesian space and image space is represented by [27]

$$\dot{x} = J_I(r)\dot{r} \quad (3)$$

where $J_I(r) \in \mathfrak{R}^{m \times m}$ is the image Jacobian matrix [28]–[30]. From (2) and (3), we have [31]

$$\dot{x} = J_I(r)J_m(q)\dot{q} = J(q)\dot{q} \quad (4)$$

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

The equations of motion of constrained robot with n degrees of freedom can be expressed in joint coordinates as [26], [32]

$$\begin{aligned} M(q)\ddot{q} + \left(\frac{1}{2}\dot{M}(q) + S(q, \dot{q}) + B \right) \dot{q} + g(q) + J_m^T(q)B_t\dot{r} \\ = \tau + J_m^T(q)f \end{aligned} \quad (5)$$

where $M(q) \in \mathfrak{R}^{n \times n}$ is the inertia matrix, $S(q, \dot{q})\dot{q} = (1/2)\dot{M}(q)\dot{q} - (1/2)\{(\partial/\partial q)\dot{q}^T M(q)\dot{q}\}^T$ is a skew-symmetric matrix, $B \in \mathfrak{R}^{n \times n}$ is the joint friction matrix, $g(q) \in \mathfrak{R}^n$ is the gravitational force, $f \in \mathfrak{R}^n$ is a contact force vector normal to the constraint surface, $B_t\dot{r}$ denotes the contact friction arising in the direction of manipulator end-point movement \dot{r} , $B_t \in \mathfrak{R}^{m \times m}$ is the contact friction matrix, and $\tau \in \mathfrak{R}^n$ is the joint torque applied to the robot.

We consider a constraint surface that can be defined in an algebraic term as

$$\Psi(r) = 0 \quad (6)$$

where $\Psi(r) : \mathfrak{R}^m \rightarrow \mathfrak{R}^1$ is a given scalar function. Differentiating (6) with respect to time yields the following velocity constraint:

$$\frac{\partial \Psi(r)}{\partial r} \dot{r} = 0. \quad (7)$$

The contact force normal to the constraint surface is then given by

$$f = d(r)\lambda \quad (8)$$

where $d(r) = (\partial \Psi(r)/\partial r) / \|\partial \Psi(r)/\partial r\| \in \mathfrak{R}^m$ is a unit vector denoting the normal direction to the constraint surface and $\lambda \in \mathfrak{R}$ is defined as a magnitude of the contact force. Hence, using (8), the robot dynamics (5) can be represented as

$$\begin{aligned} M(q)\ddot{q} + \left(\frac{1}{2}\dot{M}(q) + S(q, \dot{q}) + B \right) \dot{q} + g(q) + J_m^T(q)B_t\dot{r} \\ = \tau + D^T(q)\lambda \end{aligned} \quad (9)$$

where $D(q) = [(\partial \Psi(r)/\partial r)^T / \|\partial \Psi(r)/\partial r\|]J_m(q)$ is a Jacobian of the constraint function such that

$$D(q)\dot{q} = 0. \quad (10)$$

One of the early important results in dynamic force control was the model-based approach proposed by McClamroch and Wang [4]. It was shown using a nonlinear transformation that the dynamic equation can be decoupled into a differential equation that characterizes the motion and an equation that describes the contact force during the constrained motion. However, it is well known that the method cannot be extended into uncalibrated vision–force control problem with uncertain constraint because the nonlinear transformation requires the exact knowledge of the constraint surface.

Another important work in dynamic force control is the joint-space projection method proposed by Arimoto [26]. A sliding vector for motion subspace is proposed as

$$s = \dot{q} - Q(q)(\dot{q}_d - \alpha \Delta q) \quad (11)$$

where $\Delta q = q - q_d$, with q_d being the desired joint trajectory and α a positive constant gain, and $Q(q) = I - (D^T(q)D(q) / \|D(q)\|^2)$. The role of the projection matrix is to project the motion error onto a tangent plane perpendicular to the normal vector $D^T(q)$. Note that $s^T D^T(q)\lambda = \dot{q}^T D^T(q) - (\dot{q}_d - \alpha \Delta q)^T Q(q)D^T(q) = 0$, since $D(q)\dot{q} = 0$ and $Q(q)D^T(q) = 0$. Hence, the sliding motion error can be decoupled from the

force. Similarly, the aforementioned projection matrix requires the exact knowledge of the constraint surface. In addition, the dynamic force controller is formulated in the joint space. When the constraint is uncertain, the desired position on the constraint surface in Cartesian space is also uncertain, and hence, the desired joint position q_d cannot be computed using inverse kinematics even if the robot kinematics is known.

Using vision-based control, the desired position or trajectory on the uncertain constraint surface can then be obtained in image space directly. Let $x_d(t) \in \mathbb{R}^m$ denote the desired motion trajectory in image space and $\dot{x}_d(t) \in \mathbb{R}^m$ the desired speed trajectory; a sliding motion vector can be defined using image error as

$$s = \dot{q} - \hat{Q}(q)\hat{J}^+(q)(\dot{x}_d - \alpha\Delta x) \quad (12)$$

where $\Delta x = x - x_d$ is the image tracking error, $\hat{J}(q)$ is the estimation of $J(q)$, $\hat{Q}(q) = I - (\hat{D}^T(q)\hat{D}(q)/\|\hat{D}(q)\|^2)$ is the estimation of $Q(q)$, and $\hat{D}(q)$ is the estimation of $D(q)$. However, note that

$$s^T \hat{D}^T(q) = \dot{q}^T \hat{D}^T(q). \quad (13)$$

Since $\hat{D}(q)\dot{q} \neq 0$ in general, the image tracking error cannot be decoupled from the force. In addition, since $\hat{Q}(q)$ is not of full-rank and $\hat{J}(q)\dot{q} \neq \dot{x}$, it is clear that the convergence of the sliding vector s does not imply the convergence of the image errors. Therefore, the image and force cannot be resolved by using existing dynamic force control methods.

III. ADAPTIVE VISION AND FORCE TRACKING CONTROL

In this section, a visually servoed motion and force tracking controller is proposed. We consider a robot with unknown external parameters due to the constraint surface and unknown internal parameters due to dynamics and kinematics. Since the constraint surface is uncertain, the desired path on the surface cannot be obtained in Cartesian space. However, using cameras, the desired path can be obtained in image space by using features such as edges or corners. In the absence of obvious features, markers should be used on the constraint surface (see [14] and [16]).

The dynamic model, as described by (9), is linear in a set of physical parameters $\theta_d = (\theta_{d1}, \dots, \theta_{dp})^T$ as

$$M(q)\ddot{q} + \left(\frac{1}{2}\dot{M}(q) + S(q, \dot{q}) + B \right) \dot{q} + g(q) + J_m^T(q)B_t J_m(q)\dot{q} = Y_d(q, \dot{q}, \ddot{q})\theta_d \quad (14)$$

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

The right-hand side of (4) is linear in a set of kinematic parameters $\theta_{ki} = (\theta_{ki1}, \dots, \theta_{kiiq})^T$, such as link lengths, joint offsets, and camera parameters. Then, (4) can be expressed as

$$\dot{x} = J(q)\dot{q} = Y_{ki}(q, \dot{q})\theta_{ki} \quad (15)$$

where $Y_{ki}(q, \dot{q}) \in \mathbb{R}^{m \times h}$ is called the kinematic regressor matrix.

When the kinematics is uncertain, the parameters of the Jacobian matrix are uncertain, and we estimate the parameters in (15) as

$$\hat{x} = \hat{J}(q, \hat{\theta}_{ki})\dot{q} = Y_{ki}(q, \dot{q})\hat{\theta}_{ki} \quad (16)$$

where $\hat{J}(q, \hat{\theta}_{ki})$ is the estimations of $J(q)$. The parameters $\hat{\theta}_{ki}$ will be updated by parameter update law to be defined later.

When the constraint surface is uncertain, $J_m(q)\hat{d}(\hat{r})$ can be expressed as

$$J_m^T(q)\hat{d}(\hat{r}) = Y_f(q)\theta_f \quad (17)$$

where $\theta_f = (\theta_{f1}, \dots, \theta_{fi})^T$ is a vector of parameters of $J_m(q)$ and $Y_f(q) \in \mathbb{R}^{n \times i}$, and $\hat{d}(\hat{r})$ is a fixed estimation of $d(r)$ that is not updated. The use of $\hat{d}(\hat{r})$ results in an estimation error $J_m(q)(d(r) - \hat{d}(\hat{r}))$ that can be expressed as

$$J_m^T(q)(d(r) - \hat{d}(\hat{r})) = Y_{md}(q)\theta_{md} \quad (18)$$

where $r = h(q)$, $\hat{r} = \hat{h}(q)$, $\theta_{md} = (\theta_{md1}, \dots, \theta_{mdj})^T$ denotes the unknown lumped parameters of $J_m^T(q)$ and $d(r) - \hat{d}(\hat{r})$, and $Y_{md}(q) \in \mathbb{R}^{n \times j}$. The unknown lumped parameters will be updated by parameter update law to be defined later.

Let us define a vector $\dot{x}_r \in \mathbb{R}^m$ as

$$\dot{x}_r = (\dot{x}_d - \alpha\Delta x) + \beta(\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1}R(x)\hat{d}(\hat{r})\Delta F \quad (19)$$

where α and β are positive constants, $x_d(t) \in \mathbb{R}^m$ is the desired motion trajectory in image space, $\dot{x}_d(t) \in \mathbb{R}^m$ is the desired speed trajectory, $\Delta x = x - x_d$ is the image tracking error, $\hat{J}(q, \hat{\theta}_{ki})$ is an estimation of $J(q)$, $\hat{J}^+(q, \hat{\theta}_{ki}) = \hat{J}^T(q, \hat{\theta}_{ki})(\hat{J}(q, \hat{\theta}_{ki})\hat{J}^T(q, \hat{\theta}_{ki}))^{-1}$ is the pseudoinverse of the Jacobian matrix, $\hat{J}_m(q, \hat{\theta}_f)$ is an estimation of $J_m(q)$, $\Delta F = \int_0^t (\lambda(\sigma) - \lambda_d(\sigma))d\sigma$, $\lambda_d(t)$ is the desired force trajectory, and $R(x)$ is a rotation matrix that will be defined later.

Differentiating (19) with respect to time, we get

$$\begin{aligned} \ddot{x}_r = & (\ddot{x}_d - \alpha\Delta\dot{x}) + \beta(\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1}R(x)\hat{d}(\hat{r})\Delta\lambda \\ & + \beta(\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1}R(x)\dot{\hat{d}}(\hat{r}, \dot{\hat{r}})\Delta F \\ & + \beta(\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1}\dot{R}(x)\hat{d}(\hat{r})\Delta F \\ & + \beta\dot{J}_I R(x)\hat{d}(\hat{r})\Delta F \end{aligned} \quad (20)$$

where $\dot{J}_I = (d/dt)((\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1})$.

In order to prove the stability of the vision–force tracking system, we define an adaptive vision space sliding vector using (19) as

$$\hat{s}_x = \hat{x} - \dot{x}_r = \hat{J}(q, \hat{\theta}_{ki})\dot{q} - \dot{x}_r. \quad (21)$$

Differentiating the previous equation with respect to time, we have

$$\dot{\hat{s}}_x = \dot{\hat{x}} - \ddot{x}_r = \hat{J}(q, \hat{\theta}_{ki})\ddot{q} + \dot{\hat{J}}(q, \hat{\theta}_{ki})\dot{q} - \ddot{x}_r. \quad (22)$$

Next, let

$$\dot{q}_r = \hat{J}^+(q, \hat{\theta}_{ki})\dot{x}_r + (I_n - \hat{J}^+(q, \hat{\theta}_{ki})\hat{J}(q, \hat{\theta}_{ki}))\psi \quad (23)$$

where $\psi \in \mathfrak{R}^n$ is minus the gradient of the convex function to be optimized [33]. From (23), we have

$$\begin{aligned} \ddot{q}_r &= \hat{J}^+(q, \hat{\theta}_{ki})\ddot{x}_r + \dot{\hat{J}}^+(q, \hat{\theta}_{ki})\dot{x}_r \\ &+ (I_n - \hat{J}^+(q, \hat{\theta}_{ki})\hat{J}(q, \hat{\theta}_{ki}))\dot{\psi} \\ &- (\dot{\hat{J}}^+(q, \hat{\theta}_{ki})\hat{J}(q, \hat{\theta}_{ki}) + \hat{J}^+(q, \hat{\theta}_{ki})\dot{\hat{J}}(q, \hat{\theta}_{ki}))\psi. \end{aligned} \quad (24)$$

Hence, we have an adaptive sliding vector in joint space as

$$s = \dot{q} - \dot{q}_r \quad (25)$$

and

$$\dot{s} = \ddot{q} - \ddot{q}_r. \quad (26)$$

Multiplying both side of (25) by $\hat{J}(q, \hat{\theta}_{ki})$ and using (21), we have

$$\hat{J}(q, \hat{\theta}_{ki})s = \hat{J}(q, \hat{\theta}_{ki})\dot{q} - \dot{x}_r = \hat{s}_x. \quad (27)$$

Substituting (25) into (9), we have

$$\begin{aligned} M(q)\dot{s} + \left(\frac{1}{2}\dot{M}(q) + S(q, \dot{q}) + B \right) s + J_m^T(q)B_t J_m(q)s \\ + M(q)\ddot{q}_r + \left(\frac{1}{2}\dot{M}(q) + S(q, \dot{q}) + B \right) \dot{q}_r + g(q) \\ + J_m^T(q)B_t J_m(q)\dot{q}_r = \tau + D^T(q)\lambda. \end{aligned} \quad (28)$$

The last three terms on the left-hand side of (28) are linear in a set of dynamics parameters θ_d and can be expressed as

$$\begin{aligned} M(q)\ddot{q}_r + \left(\frac{1}{2}\dot{M}(q) + S(q, \dot{q}) + B \right) \dot{q}_r + g(q) \\ + J_m^T(q)B_t J_m(q)\dot{q}_r = Y_d(q, \dot{q}, \ddot{q}_r, \dot{q}_r)\theta_d. \end{aligned} \quad (29)$$

Then, the dynamics equation (28) can be expressed as

$$\begin{aligned} M(q)\dot{s} + \left(\frac{1}{2}\dot{M}(q) + S(q, \dot{q}) + B \right) s + J_m^T(q)B_t J_m(q)s \\ + Y_d(q, \dot{q}, \ddot{q}_r, \dot{q}_r)\theta_d = \tau + D^T(q)\lambda. \end{aligned} \quad (30)$$

The vision and force tracking controller is proposed as

$$\begin{aligned} u &= -\hat{J}^T(q, \hat{\theta}_{ki})K(\Delta\hat{x} + \alpha\Delta x) + Y_d(q, \dot{q}, \ddot{q}_r, \dot{q}_r)\hat{\theta}_d \\ &- Y_{md}(q)\hat{\theta}_{md}\lambda - \hat{J}_m^T(q, \hat{\theta}_f)\hat{d}(\hat{r})\lambda \\ &+ \hat{J}_m^T(q, \hat{\theta}_f)R(x)\hat{d}(\hat{r})(\Delta\lambda + \gamma\Delta F) \end{aligned} \quad (31)$$

where $\Delta\hat{x} = \hat{x} - \hat{x}_d$, $K \in \mathfrak{R}^{m \times m}$ is a positive-definite matrix, and γ is a positive constant. The estimated parameters $\hat{\theta}_d$, $\hat{\theta}_f$, $\hat{\theta}_{md}$, $\hat{\theta}_{ki}$ are updated by

$$\dot{\hat{\theta}}_d = -L_d Y_d^T(q, \dot{q}, \ddot{q}_r, \dot{q}_r)s \quad (32)$$

$$\dot{\hat{\theta}}_f = L_f Y_f^T(q)s\lambda \quad (33)$$

$$\dot{\hat{\theta}}_{md} = L_{md} Y_{md}^T(q)s\lambda \quad (34)$$

$$\dot{\hat{\theta}}_{ki} = 2L_{ki} Y_{ki}^T(q, \dot{q})K(\Delta\hat{x} + \alpha\Delta x) \quad (35)$$

where $L_d \in \mathfrak{R}^{p \times p}$, $L_f \in \mathfrak{R}^{n \times n}$, $L_{md} \in \mathfrak{R}^{j \times j}$, $L_{ki} \in R^{h \times h}$ are symmetric and positive-definite gain matrices of the update

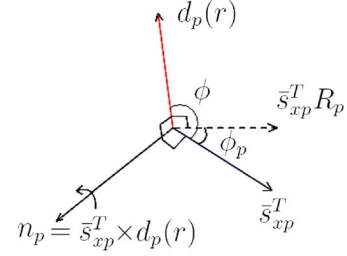


Fig. 2. Illustration of the rotation matrix.

laws. The rotation matrix $R(x)$ is designed so that

$$\bar{s}_x^T R(x)\hat{d}(\hat{r}) = 0 \quad (36)$$

where

$$\begin{aligned} \bar{s}_x &= \{\beta(\Delta\hat{x} + \alpha\Delta x)^T K(\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1}\Delta F \\ &+ s_m^T(\Delta\lambda + \gamma\Delta F)\}^T \\ s_m &= \hat{J}_m(q, \hat{\theta}_f)\{\dot{q} - \hat{J}^+(q, \hat{\theta}_{ki})(\dot{x}_d - \alpha\Delta x) \\ &- (I_n - \hat{J}^+(q, \hat{\theta}_{ki})\hat{J}(q, \hat{\theta}_{ki}))\psi\}. \end{aligned} \quad (37)$$

The role of the rotation matrix is to rotate $\hat{d}(\hat{r})$ so that $R(x)\hat{d}(\hat{r})$ is orthonormal to \bar{s}_x . In general, \bar{s}_x can be partitioned as $\bar{s}_x = (\bar{s}_{xp}, \bar{s}_{xo})^T$, where \bar{s}_{xp} is the position vector and \bar{s}_{xo} is the orientation vector. Therefore, $\hat{d}(\hat{r})$ can also be partitioned as $(\hat{d}_p(\hat{r}), \hat{d}_o(\hat{r}))^T$. Next, let n_p be a unit vector normal to both the vectors \bar{s}_{xp} and $\hat{d}_p(\hat{r})$, as illustrated in Fig. 2. The rotation matrix $R(x)$ is then introduced as

$$R(x) = \begin{bmatrix} R_p & 0 \\ 0 & R_o \end{bmatrix} \quad (38)$$

where the rotation matrix R_p is to rotate the vector \bar{s}_{xp} about the axis n_p , so that the vector $\bar{s}_{xp}^T R_p$ is perpendicular to the vector $\hat{d}_p(\hat{r})$, as shown in Fig. 2. ϕ_p is the angle between $\bar{s}_{xp}^T R_p$ and $\hat{d}_p(\hat{r})$, which can be determined from the angle ϕ between \bar{s}_{xp} and $\hat{d}_p(\hat{r})$ (see Fig. 2). Here, the symbol \times means cross product. The rotation matrix R_o can be similarly designed. However, since the constraint surface is usually independent of r_o , the rotation matrix R_o can be set as an identity matrix. Note that when \bar{s}_r reduces to zero, $R(x)$ can be set as any value because (36) is always satisfied when $\bar{s}_x = 0$.

Substituting (31) into (30), we have the closed-loop equation

$$\begin{aligned} M(q)\dot{s} + \left(\frac{1}{2}\dot{M}(q) + S(q, \dot{q}) + B \right) s + J_m^T(q)B_t J_m(q)s \\ + \hat{J}^T(q, \hat{\theta}_{ki})K(\Delta\hat{x} + \alpha\Delta x) + Y_d(q, \dot{q}, \ddot{q}_r, \dot{q}_r)\Delta\theta_d \\ = Y_f(q)\Delta\theta_f\lambda + Y_{md}(q)\Delta\theta_{md}\lambda \\ + \hat{J}_m^T(q, \hat{\theta}_f)R(x)\hat{d}(\hat{r})(\Delta\lambda + \gamma\Delta F) \end{aligned} \quad (39)$$

where I is the identity matrix, $\Delta\theta_d = \theta_d - \hat{\theta}_d$, and $\Delta\theta_f = \theta_f - \hat{\theta}_f$. To carry out the stability analysis, we define the

Lyapunov-like function candidate V as

$$\begin{aligned} V &= \frac{1}{2}s^T M(q)s + \alpha\Delta x^T K\Delta x + \frac{1}{2}\Delta\theta_d^T L_d^{-1}\Delta\theta_d \\ &+ \frac{1}{2}\Delta\theta_f^T L_f^{-1}\Delta\theta_f + \frac{1}{2}\Delta\theta_{ki}^T L_{ki}^{-1}\Delta\theta_{ki} \\ &+ \frac{1}{2}\Delta\theta_{md}^T L_{md}^{-1}\Delta\theta_{md} + \frac{1}{2}\beta\Delta F^2 \end{aligned} \quad (40)$$

where $\Delta\theta_{ki} = \theta_{ki} - \hat{\theta}_{ki}$. Differentiating V with respect to time, substituting (32)–(34) and (39) into it, and using (27), we have

$$\begin{aligned} \dot{V} &= -s^T (B + J_m^T(q)B_t J_m(q))s - \hat{s}_x^T K(\Delta\hat{x} + \alpha\Delta x) \\ &+ s^T \hat{J}_m^T(q, \hat{\theta}_f)R(x)\hat{d}(\hat{r})(\Delta\lambda + \gamma\Delta F) \\ &+ 2\alpha\Delta x^T K\Delta\dot{x} - \Delta\theta_{ki}^T L_{ki}^{-1}\hat{\theta}_{ki} + \beta\Delta F\Delta\lambda \end{aligned} \quad (41)$$

where we note that $S(q, \dot{q})$ is skew-symmetric.

From (19), (25), and (37), we note that

$$\begin{aligned} \hat{J}_m(q, \hat{\theta}_f)s &= \hat{J}_m(q, \hat{\theta}_f)\dot{q} - \hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki})(\dot{x}_d - \alpha\Delta x) \\ &- \hat{J}_m(q, \hat{\theta}_f)(I_n - \hat{J}^+(q, \hat{\theta}_{ki})\hat{J}(q, \hat{\theta}_{ki}))\psi \\ &- \beta R(x)\hat{d}(\hat{r})\Delta F = s_m - \beta R(x)\hat{d}(\hat{r})\Delta F. \end{aligned} \quad (42)$$

Next, substituting (19), (21), and (42) into (41) and using (37), we obtain

$$\begin{aligned} \dot{V} &= -s^T (B + J_m^T(q)B_t J_m(q))s \\ &- (\Delta\hat{x} + \alpha\Delta x)^T K(\Delta\hat{x} + \alpha\Delta x) \\ &+ \beta(\Delta\hat{x} + \alpha\Delta x)^T K(\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1}R(x)\hat{d}(\hat{r})\Delta F \\ &+ s_m^T R(x)\hat{d}(\hat{r})(\Delta\lambda + \gamma\Delta F) - \beta\Delta F(\Delta\lambda + \gamma\Delta F) \\ &+ 2\alpha\Delta x^T K\Delta\dot{x} - \Delta\theta_{ki}^T L_{ki}^{-1}\hat{\theta}_{ki} + \beta\Delta F\Delta\lambda \\ &= -s^T (B + J_m^T(q)B_t J_m(q))s \\ &- (\Delta\hat{x} + \alpha\Delta x)^T K(\Delta\hat{x} + \alpha\Delta x) \\ &+ \hat{s}_x^T R(x)\hat{d}(\hat{r}) - \beta\Delta F(\Delta\lambda + \gamma\Delta F) \\ &+ 2\alpha\Delta x^T K\Delta\dot{x} - \Delta\theta_{ki}^T L_{ki}^{-1}\hat{\theta}_{ki} + \beta\Delta F\Delta\lambda. \end{aligned} \quad (43)$$

Using (36) in the previous equation yields

$$\begin{aligned} \dot{V} &= -s^T (B + J_m^T(q)B_t J_m(q))s \\ &- (\Delta\hat{x} + \alpha\Delta x)^T K(\Delta\hat{x} + \alpha\Delta x) + 2\alpha\Delta x^T K\Delta\dot{x} \\ &- \Delta\theta_{ki}^T L_{ki}^{-1}\hat{\theta}_{ki} - \beta\gamma\Delta F^2 \end{aligned} \quad (44)$$

where we note that $R^T(x)R(x) = I$ and $\hat{d}^T(\hat{r})\hat{d}(\hat{r}) = 1$.

From (15) and (16), since $\hat{x} = \dot{x} - Y_{ki}(q, \dot{q})\Delta\theta_{ki}$, we have

$$\Delta\hat{x} = \Delta\dot{x} - Y_{ki}(q, \dot{q})\Delta\theta_{ki}. \quad (45)$$

Substituting (35) and (45) into (44), we have

$$\begin{aligned} \dot{V} &= -s^T (B + J_m^T(q)B_t J_m(q))s - \Delta\dot{x}^T K\Delta\dot{x} - \alpha^2\Delta x^T K\Delta x \\ &- \Delta\theta_{ki}^T Y_{ki}^T(q, \dot{q})KY_{ki}(q, \dot{q})\Delta\theta_{ki} - \beta\gamma\Delta F^2 \leq 0. \end{aligned} \quad (46)$$

We can now state the following theorem.

Theorem: The adaptive Jacobian control law (31) and the update laws (32)–(35) for the robot system (9) result in the convergence of vision and force tracking errors, i.e., $x(t) - x_d(t) \rightarrow 0$ and $\dot{x}(t) - \dot{x}_d(t) \rightarrow 0$ as $t \rightarrow \infty$. In addition, the contact force $\lambda(t)$ also converges to $\lambda_d(t)$ as $t \rightarrow \infty$.

Proof: Since $M(q)$ is positive-definite, V is positive-definite in s , Δx , $\Delta\theta_d$, $\Delta\theta_f$, $\Delta\theta_{ki}$, $\Delta\theta_{md}$, and ΔF . Since $\dot{V} \leq 0$, V is bounded. Therefore, s , Δx , $\Delta\theta_d$, $\Delta\theta_f$, $\Delta\theta_{ki}$, $\Delta\theta_{md}$, and ΔF are bounded vectors. This implies that $\hat{\theta}_d$, $\hat{\theta}_f$, $\hat{\theta}_{ki}$, $\hat{\theta}_{md}$, and x are bounded, and $\hat{s}_x = \hat{J}(q, \hat{\theta}_{ki})s$ is also bounded.

As seen from (46), \dot{V} is negative-definite in s , $\Delta\dot{x}$, Δx , $Y_{ki}(q, \dot{q})\Delta\theta_{ki}$, and ΔF . To show the convergence of the errors, Barbalat's lemma [34], [35] shall be used to check the uniform continuity of \dot{V} . In the following development, we proceed to show the uniform continuity by proving that \dot{V} is bounded.

First, note that \dot{x}_r , \dot{x} are bounded, as seen from (19) and (21). From (23), we can conclude that \dot{q}_r is bounded when $\hat{J}(q, \hat{\theta}_{ki})$ is nonsingular. Therefore, \dot{q} is bounded since s is bounded. The boundedness of \dot{q} means that \dot{x} , \dot{r} are bounded. Hence, $\Delta\dot{x}$ is bounded, and $\ddot{x}_r - \beta(\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1}R(x)d(r)\Delta\lambda$ from (20) is also bounded if \ddot{x}_d is bounded. $\hat{d}(r, \dot{r})$ is bounded because r , \dot{r} are bounded.

As seen from (35), $\hat{\theta}_{ki}$ is bounded because \dot{q} , $\Delta\dot{x}$, and Δx are all bounded. Hence, $\hat{J}(q, \hat{\theta}_{ki})$ is bounded. Using (24), we have

$$\begin{aligned} \ddot{q}_r - \beta\hat{J}^+(q, \hat{\theta}_{ki})(\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1}R(x)\hat{d}(\hat{r})\Delta\lambda \\ = \hat{J}^+(q, \hat{\theta}_{ki})(\ddot{x}_r - \beta(\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1}R(x)\hat{d}(\hat{r})\Delta\lambda) \\ + \hat{J}^+(q, \hat{\theta}_{ki})\dot{x}_r + (I_n - \hat{J}^+(q, \hat{\theta}_{ki})\hat{J}(q, \hat{\theta}_{ki}))\dot{\psi} \\ - (\hat{J}^+(q, \hat{\theta}_{ki})\hat{J}(q, \hat{\theta}_{ki}) + \hat{J}^+(q, \hat{\theta}_{ki})\hat{J}(q, \hat{\theta}_{ki}))\psi. \end{aligned} \quad (47)$$

Since all the terms on the right-hand side of the previous equation are bounded, $\ddot{q}_r - \beta\hat{J}^+(q, \hat{\theta}_{ki})(\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1}R(x)d(r)\Delta\lambda$ is therefore bounded. Next, let

$$\begin{aligned} Y_d(q, \dot{q}, \dot{q}_r, \ddot{q}_r)\Delta\theta_d &= (M(q) - \hat{M}(q))\ddot{q}_r \\ &+ \left[\frac{1}{2}(\dot{M}(q) - \hat{M}(q)) + S(q, \dot{q}) - \hat{S}(q, \dot{q}) + B - \hat{B} \right. \\ &+ J_m^T(q)B_t J_m(q) - \hat{J}_m^T(q)B_t \hat{J}_m(q) \left. \right] \dot{q}_r + g(q) - \hat{g}(q) \\ &+ \beta M(q)\hat{J}^+(q, \hat{\theta}_{ki})(\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1}R(x) \\ &\times (\dot{d}(h(q), J_m(q)\dot{q}) - \hat{d}(\hat{h}(q), \hat{J}_m(q, \hat{\theta}_f)\dot{q}))\Delta F \\ &= (M(q) - \hat{M}(q))\ddot{q}_r + \bar{Z}_d(q, \dot{q}, \dot{q}_r, x, \hat{\theta}_f, \hat{\theta}_{ki})\Delta\theta_d \end{aligned} \quad (48)$$

where $\bar{Z}_d(q, \dot{q}, \dot{q}_r, x, \hat{\theta}_f, \hat{\theta}_{ki})\Delta\theta_d = ((1/2)(\dot{M}(q) - \hat{M}(q)) + S(q, \dot{q}) - \hat{S}(q, \dot{q}) + B - \hat{B} + J_m^T(q)B_t J_m(q) - \hat{J}_m^T(q)B_t \hat{J}_m(q))\dot{q}_r + g(q) - \hat{g}(q) + \beta M(q)\hat{J}^+(q, \hat{\theta}_{ki})(\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1}R(x)(\dot{d}(h(q), J_m(q)\dot{q}) - \hat{d}(\hat{h}(q), \hat{J}_m(q, \hat{\theta}_f)\dot{q}))\Delta F$. Then, from (39) and (48), since $D(q)\ddot{q} = -\dot{D}(q)\dot{q}$, we have

$$\begin{aligned} -\dot{D}(q)\dot{q} + D(q)M^{-1}(q)\{-M(q)(\ddot{q}_r \\ - \beta\hat{J}^+(q, \hat{\theta}_{ki})(\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1}R(x)\hat{d}(\hat{r})\Delta\lambda \end{aligned}$$

$$\begin{aligned}
& + (M(q) - \hat{M}(q))(\ddot{q}_r - \beta \hat{J}^+(q, \hat{\theta}_{ki})) \\
& \times (\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^{-1} R(x) \hat{d}(\hat{r}) \Delta \lambda + r(t) \} \\
= & D(q) M^{-1}(q) \{ D_1 \Delta \lambda - \hat{J}_m^T(q, \hat{\theta}_f) R(x) \hat{d}(\hat{r}) \lambda_d \\
& + \beta \hat{M}(q) \hat{J}^+(q, \hat{\theta}_{ki}) (\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^{-1} R(x) \hat{d}(\hat{r}) \Delta \lambda \\
& + D_1 \lambda_d \} \quad (49)
\end{aligned}$$

where $r(t) = ((1/2)\dot{M}(q) + S(q, \dot{q}) + B)s + \bar{Z}_d(q, \dot{q}, \dot{q}_r, x, \hat{\theta}_f, \hat{\theta}_{ki}) \Delta \theta_d + \hat{J}^T(q, \hat{\theta}_{ki}) K(\Delta \hat{x} + \alpha \Delta x) - \gamma \hat{J}_m^T(q, \hat{\theta}_f) R(x) \hat{d}(\hat{r}) \Delta F$ and $D_1 = (Y_f(q) \Delta \theta_f + Y_{md}(q) \Delta \theta_{md} + \hat{J}_m^T(q, \hat{\theta}_f) R(x) \hat{d}(\hat{r}))$.

The previous equation can be written as

$$\bar{r}(t) = k(t) \Delta \lambda \quad (50)$$

where $\bar{r}(t) = -\dot{D}(q) \dot{q} + D(q) M^{-1}(q) \{ -M(q) (\ddot{q}_r - \beta \hat{J}^+(q, \hat{\theta}_{ki}) (\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^{-1} R(x) \hat{d}(\hat{r}) \Delta \lambda) + (M(q) - \hat{M}(q)) (\ddot{q}_r - \beta \hat{J}^+(q, \hat{\theta}_{ki}) (\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^{-1} R(x) \hat{d}(\hat{r}) \Delta \lambda) + r(t) - (Y_f(q) \Delta \theta_f + Y_{md}(q) \Delta \theta_{md}) \lambda_d \}$ and $k(t) = D(q) M^{-1}(q) \{ D_1 + \beta \hat{M}(q) \hat{J}^+(q, \hat{\theta}_{ki}) (\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^{-1} R(x) \hat{d}(\hat{r}) \}$ are bounded scalars. Hence, $\Delta \lambda$ is bounded, and the boundedness of $\Delta \lambda$ implies that \ddot{x}_r , \dot{q}_r , and \ddot{q}_r are bounded.

From the closed-loop equation (39), we can conclude that \dot{s} is bounded. The boundedness of \dot{s} implies the boundedness of \ddot{q} , as seen from (26). Since $\ddot{x} = J(q) \ddot{q} + \dot{J}(q) \dot{q}$, \ddot{x} is bounded, and hence, $\Delta \ddot{x}$ is also bounded. To apply Barbalat's lemma [34], [35], let us check the uniform continuity of \dot{V} . Differentiating (46) with respect to time gives

$$\begin{aligned}
\dot{V} = & -2s^T (B + J_m^T(q) B_t J_m(q)) \dot{s} + 2s^T J_m^T(q) B_t \dot{J}_m(q) s \\
& - 2\Delta \dot{x}^T K \Delta \ddot{x} - 2\alpha^2 \Delta x^T K \Delta \dot{x} \\
& + 2\Delta \theta_{ki}^T Y_{ki}^T(q, \dot{q}) K Y_{ki}(q, \dot{q}) \dot{\theta}_{ki} \\
& - 2\Delta \theta_{ki}^T Y_{ki}^T(q, \dot{q}) K \dot{Y}_{ki}(q, \dot{q}) \Delta \theta_{ki} - 2\beta \gamma \Delta F \Delta \lambda. \quad (51)
\end{aligned}$$

This shows that \dot{V} is bounded since Δx , $\Delta \dot{x}$, \dot{q} , $\Delta \ddot{x}$, \dot{s} , ΔF , $\Delta \lambda$ are all bounded. Hence, \dot{V} is uniformly continuous. Using Barbalat's lemma, we have $\Delta x \rightarrow 0$, $\Delta \dot{x} \rightarrow 0$, $Y_{ki}(q, \dot{q}) \Delta \theta_{ki} = \dot{x} - \hat{x} \rightarrow 0$, $\Delta F \rightarrow 0$ as $t \rightarrow \infty$.

Next, we proceed to show the convergence of the force tracking error. From (32), (33), and (35), we note that $\hat{\theta}_d$, $\hat{\theta}_f$, $\hat{\theta}_{ki}$ are bounded. Let $z_x = \ddot{x}_r - \beta \hat{J}_I R(x) \hat{d}(\hat{r}) \Delta \lambda$ and $\hat{J}_I = (\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^{-1}$. From (20), we have

$$\begin{aligned}
z_x = & \ddot{x}_d - \alpha \Delta \dot{x} + \beta \hat{J}_I R(x) \hat{d}(\hat{r}, \hat{r}) \Delta F \\
& + \beta \hat{J}_I \dot{R}(x) \hat{d}(\hat{r}) \Delta F + \beta \hat{J}_I R(x) \hat{d}(\hat{r}) \Delta F. \quad (52)
\end{aligned}$$

Since $\Delta \ddot{x}$, \dot{x} , $\Delta \lambda$, \dot{q} are bounded, \dot{z}_x is also bounded if the derivative of \ddot{x}_d is bounded. Hence, the derivative of $\ddot{q}_r - \beta \hat{J}^+(q, \hat{\theta}_{ki}) (\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^{-1} R(x) \hat{d}(\hat{r}) \Delta \lambda$ is also bounded, as seen from (47). From (49), the derivative λ is therefore bounded. Finally, using Barbalat's lemma and the convergence of ΔF to 0, we have $\Delta \lambda \rightarrow 0$ as $t \rightarrow \infty$. $\triangle \triangle \triangle$

Remark 1: In this paper, we have shown that the image and force tracking errors of the adaptive control system can be resolved as in (36). It is shown with consideration of full robot

dynamics that stability of the vision and force control system can be guaranteed even in the presence of uncertainties in kinematics, dynamics, and constraint surface.

Remark 2: 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 (31) and the parameter update laws (32)–(35) that $\hat{J}^+(q, \hat{\theta}_{ki})$ is used only in the definition of control variable \dot{q}_r in (23). Therefore, we should be able to control this by bounding the variable or using a singularity-robust inverse of the approximate Jacobian matrix [25].

Remark 3: This paper focuses on vision and force tracking control of a single robot manipulator, but the results can also be extended to multifingered robots [36] or cooperative manipulators [37], [38]. In such tightly coupled tasks, the robots may grasp the object at an uncertain position, and each robot may not know the kinematics and dynamics of other robots.

Remark 4: The convergence of the estimated parameters can only be ensured if the trajectory is persistently exciting [26], [34]. However, a key point in adaptive control is that the tracking error will converge regardless of whether the trajectory is persistently exciting or not, i.e., one does not need parameter convergence for task convergence.

A. Dimension of Feature Space

In the previous development, we consider the case where the dimension of feature space is equal to the dimension of task space. If $x \in \mathcal{R}^\kappa$ where the dimension of feature space is not equal to the dimension of task variable $r \in \mathcal{R}^m$, then the matrix $(\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki})) \in \mathcal{R}^{m \times \kappa}$ in (19) is not a square matrix and $(\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^{-1}$ does not exist. There are two cases to be considered, namely $\kappa \geq m$ and $\kappa \leq m$.

If $\kappa \geq m$, there are redundant features with respect to the task degrees of freedom. Therefore, \dot{x}_r in (19) should be defined as

$$\dot{x}_r = (\dot{x}_d - \alpha \Delta x) + \beta (\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^+ R(x) \hat{d}(\hat{r}) \Delta F \quad (53)$$

where

$$\begin{aligned}
& (\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^+ = (\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^T \\
& \times [(\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki})) (\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^T]^{-1} \quad (54)
\end{aligned}$$

is the pseudoinverse matrix. Hence, (42) is still valid as follows:

$$\begin{aligned}
\hat{J}_m(q, \hat{\theta}_f) s = & \hat{J}_m(q, \hat{\theta}_f) \dot{q} - \hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}) (\dot{x}_d - \alpha \Delta x) \\
& - \hat{J}_m(q, \hat{\theta}_f) (I_n - \hat{J}^+(q, \hat{\theta}_{ki}) \hat{J}(q, \hat{\theta}_{ki})) \psi \\
& - \beta (\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki})) \\
& \times (\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^+ R(x) \hat{d}(\hat{r}) \Delta F \\
= & s_m - \beta R(x) \hat{d}(\hat{r}) \Delta F \quad (55)
\end{aligned}$$

where we note that

$$(\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki})) (\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^+ = I.$$

If $\kappa \leq m$, the pseudoinverse matrix must be defined as

$$\begin{aligned} & (\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^+ \\ &= [(\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^T (\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))]^{-1} \\ & \quad \times (\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^T \end{aligned} \quad (56)$$

but in this case, (42) does not hold, and it is, therefore, difficult to demonstrate the stability of the system. This could be due to the fact that there are not enough feature points x to determine the end-effector motion r [27].

B. Linear Parameterization Problem of Depth and Constraint Function

Similar to standard robot adaptive controllers, we consider the case where the unknown parameters are constant and linearly parameterizable. If the unknown parameters are time varying or linear parameterization cannot be obtained, adaptive control using basis functions [39], [40] can be used. The basic idea is to approximate the models by using neural network where the unknown weights are adjusted online by the updated law. For example, the dynamic regressor can be replaced by

$$\begin{aligned} \hat{M}(q) \ddot{q}_r + \left(\frac{1}{2} \dot{\hat{M}}(q) + \hat{S}(q, \dot{q}) + B \right) \dot{q}_r + \hat{g}(q) \\ + \hat{J}_m^T(q) \hat{B}_t \hat{J}_m(q) \dot{q}_r = \hat{W}_d \theta_d(q, \dot{q}, \ddot{q}_r, \dot{q}_r) \end{aligned} \quad (57)$$

where $\theta_d(q, \dot{q}, \ddot{q}_r, \dot{q}_r)$ is a vector of activation functions and \hat{W}_d are estimated neural network weights updated by the updated law that can be designed similarly (see [39] and [40] for details).

When the depth information of the feature points is not constant [41], the relationship between velocities in camera image space and joint space is represented by [41], [42]

$$\dot{x} = Z^{-1}(q) L(q) J_m(q) \dot{q} = Z^{-1}(q) \bar{J}(q) \dot{q} \quad (58)$$

where $J_I = Z^{-1}(q) L(q)$ is the image Jacobian matrix, $Z(q)$ is a matrix containing the depths of the feature points with respect to the camera image frame, and the matrix $L(q)$ is the remaining image Jacobian matrix. Since the depths are not constant, linear parameterization cannot be obtained for the Jacobian matrix $J(q)$ if the depth is uncertain. However, the Jacobian matrix $J(q) = Z^{-1}(q) L(q) J_m(q)$ in (58) can be estimated by neural networks as

$$\hat{J}(q, \hat{W}_{ki}) = (\hat{W}_{li1} \theta_{ki}(q), \dots, \hat{W}_{lin} \theta_{ki}(q)) \quad (59)$$

where $\hat{J}(q, \hat{W}_{ki})$ is an estimation of $J(q)$, $\theta_{ki}(q)$ is a vector of activation functions, and the estimated neural network weights \hat{W}_{ki} are updated by the update law. In many force control applications, it is also difficult to determine the structure of the constraint function, and the neural network can similarly be used to approximate the constraint Jacobian. Another method of updating the estimated depth information can be found in [42].

C. Pose Control

The motion control task can be defined either as position control only or both position and orientation control. For position

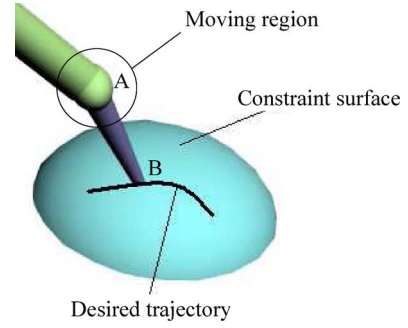


Fig. 3. Illustration of full pose control.

control, a feature point at the tool tip is specified, and the desired trajectory on the contact surface can be easily defined in image space. Some calibration-free path planning algorithms in image space can be found in [43] and [44]. However, for position and orientation control, additional feature points on the tool or end-effector are needed, but in the presence of kinematic uncertainty and camera calibration errors, it is difficult to specify the desired trajectories for the additional features with respect to the desired trajectory on the contact surface. In such cases, a desired moving region [45] should be specified to allow flexibility in the orientation, as illustrated in Fig. 3. This is reasonable since the accuracy of the position is more important as compared to the orientation. Let x_b be the features on the tool tip and x_a be the additional features for orientation control (see Fig. 3); we can partition the joint position vector q as $[q_a, q_b]$ [46], where $\dot{x}_a = J_a(q) \dot{q}_a$ and $\dot{x}_b = J_a(q) \dot{q}_a + J_b(q) \dot{q}_b$. A desired trajectory x_{bd} is specified for x_b while a desired region [45], [46] is defined for x_a as follows:

$$f(\Delta x_a) = [f_1(\Delta x_a), f_2(\Delta x_a), \dots, f_N(\Delta x_a)]^T \leq 0 \quad (60)$$

where $f_j(\Delta x_a) \in R$ are continuous scalar functions with continuous first partial derivatives, $j = 1, 2, \dots, N$, N is the total number of secondary objective functions, and $\Delta x_a = x_a - x_{a0}$, with x_{a0} being a reference point in the desired region. Some examples of desired moving regions can be found in [45]. The region tracking control formulation provides flexibility to the manipulator in the presence of uncertainties. Similarly, the estimated velocities can be defined based on the estimated Jacobian matrices as $\hat{\dot{x}}_b = \hat{J}_a(q, \hat{\theta}_{ki}) \dot{q}_a + \hat{J}_b(q, \hat{\theta}_{ki}) \dot{q}_b$ and $\hat{\dot{x}}_a = \hat{J}_a(q, \hat{\theta}_{ki}) \dot{q}_a$.

The control law is proposed with some slight modifications as follows:

$$\begin{aligned} \Delta \hat{x} &= \begin{bmatrix} \Delta \hat{x}_b \\ \Delta \hat{x}_a \end{bmatrix}, \quad \Delta x = \begin{bmatrix} \Delta x_b \\ \Delta \xi_a \end{bmatrix}, \quad x_d = \begin{bmatrix} x_{bd} \\ x_{ad} \end{bmatrix} \\ \hat{J}^T(q, \hat{\theta}_{ki}) &= \begin{bmatrix} \hat{J}_a^T(q, \hat{\theta}_{ki}) & \hat{J}_b^T(q, \hat{\theta}_{ki}) \\ \hat{J}_b^T(q, \hat{\theta}_{ki}) & 0 \end{bmatrix} \end{aligned}$$

and $\Delta \xi_a$ is the region error [45] given by

$$\Delta \xi_a = \sum_{i=1}^N k_i \max(0, f_i(\Delta x_a)) \left(\frac{\partial f_i(\Delta x_a)}{\partial \Delta x_a} \right)^T \quad (61)$$

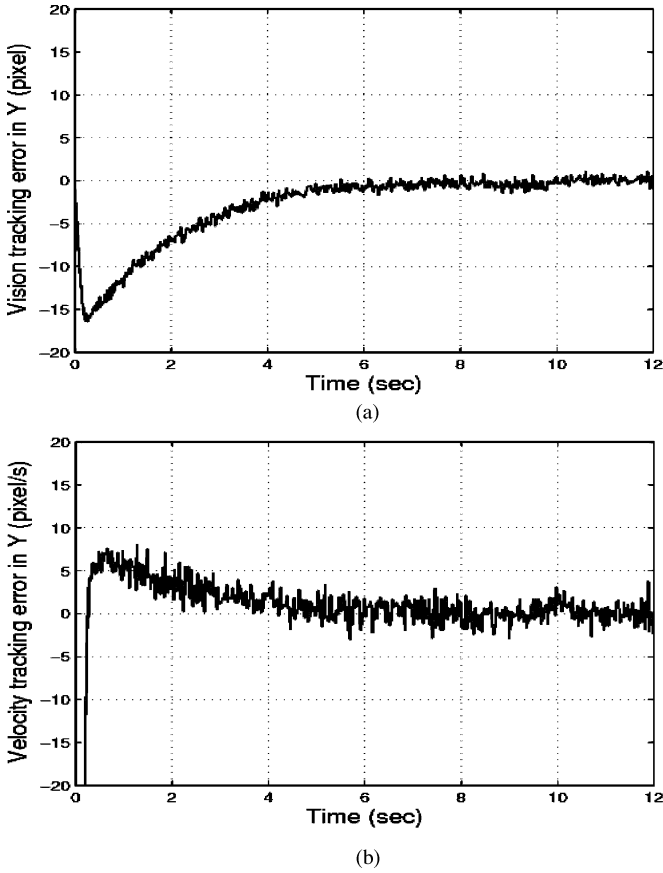


Fig. 4. Position and velocity tracking errors for the planar surface.

where k_i are positive constants. The proof of convergence can be carried out in a similar way.

IV. SIMULATION RESULTS

In this section, we present simulation results to illustrate the performance of the proposed controller. We consider a two-link manipulator whose end-effector is required to move on a constraint surface. A fixed camera is placed distance away from the robot. The Jacobian matrix $J(q)$ mapping from joint space to image space is given by

$$J(q) = \frac{f_1}{z_1 - f_1} \begin{bmatrix} \beta_1 & 0 \\ 0 & \beta_2 \end{bmatrix} \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix}$$

where $s_1 = \sin(q_1)$, $c_1 = \cos(q_1)$, $s_{12} = \sin(q_1 + q_2)$, $c_{12} = \cos(q_1 + q_2)$, and l_1, l_2 are the lengths of the first and second links, respectively. β_1, β_2 denote the scaling factors in pixels per meter, z_1 is the perpendicular distance between the robot and the camera, and f_c is the focal length of the camera.

In the simulations, the exact masses of the two links are set to 17.4 and 4.8 kg, the exact lengths l_1, l_2 of the links are set to 0.43 m, f_c is set as 50 mm, z_1 is set as 0.55 m, and $\beta_1 = \beta_2 = 10000$. White Gaussian noises with power of 1 dB are added to the position and force.

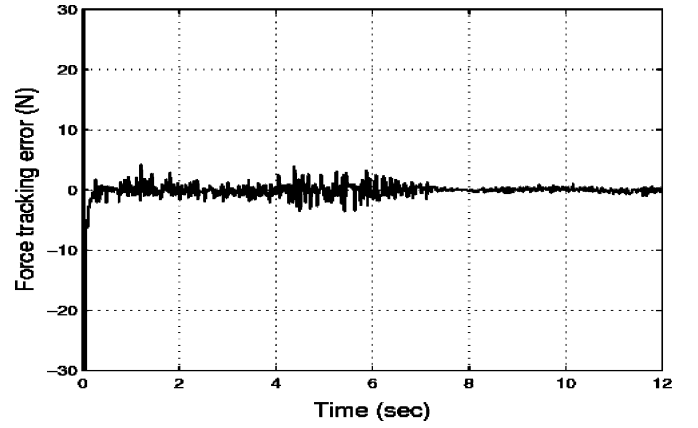


Fig. 5. Force tracking error for the planar surface.

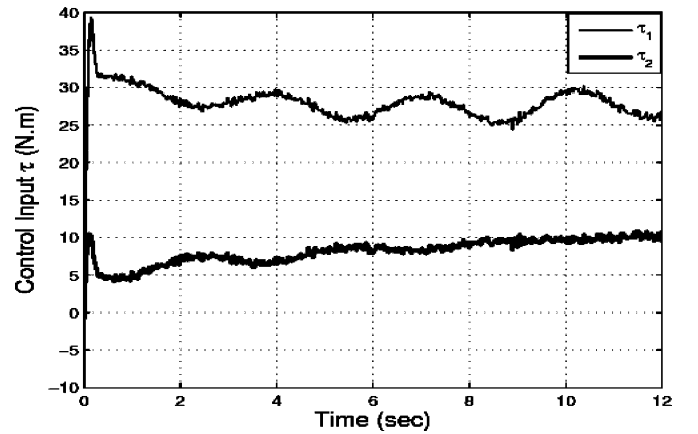


Fig. 6. Control inputs for the planar surface.

A. Planar Surface

In the first simulation, the end-effector is required to move on a constraint surface described by

$$\Psi(r(q)) = r_x + \gamma_s r_y + c = l_1 c_1 + l_2 c_{12} + \gamma_s (l_1 s_1 + l_2 s_{12}) + c = 0$$

where $r_x = l_1 c_1 + l_2 c_{12}$ and $r_y = l_1 s_1 + l_2 s_{12}$ are the positions of end-effector in Cartesian space, and γ_s and c are constant. Then, $d(r) = \partial\Psi(r)/\partial r = (1, \gamma_s)^T$. When the constraint surface is uncertain, $\hat{d}(\hat{r}) = (1, \hat{\gamma}_s)^T$ and $d(r) - \hat{d}(\hat{r}) = (0, \gamma_s - \hat{\gamma}_s)^T$. Hence, $J_m^T(q)\hat{d}(\hat{r})$ can be written as

$$J_m^T(q)\hat{d}(\hat{r}) = \begin{bmatrix} -s_1 + \hat{\gamma}_s c_1 & -s_{12} + \hat{\gamma}_s c_{12} \\ 0 & -s_{12} + \hat{\gamma}_s c_{12} \end{bmatrix} \begin{bmatrix} \theta_{f,1} \\ \theta_{f,2} \end{bmatrix} = Y_f(q)\theta_f$$

where $\theta_{f,1} = l_1$ and $\theta_{f,2} = l_2$. Note that $J_m^T(q)(d(r) - \hat{d}(\hat{r}))$ can also be written as

$$J_m^T(q)(d(r) - \hat{d}(\hat{r})) = \begin{bmatrix} c_1 & c_{12} \\ 0 & c_{12} \end{bmatrix} \begin{bmatrix} \theta_{m,d,1} \\ \theta_{m,d,2} \end{bmatrix} = Y_{m,d}(q)\theta_{m,d}$$

where $\theta_{m,d,1} = l_1(\gamma_s - \hat{\gamma}_s)$ and $\theta_{m,d,2} = l_2(\gamma_s - \hat{\gamma}_s)$.

The camera in this setup is placed parallel to the constraint surface, and hence, z_1 is fixed.

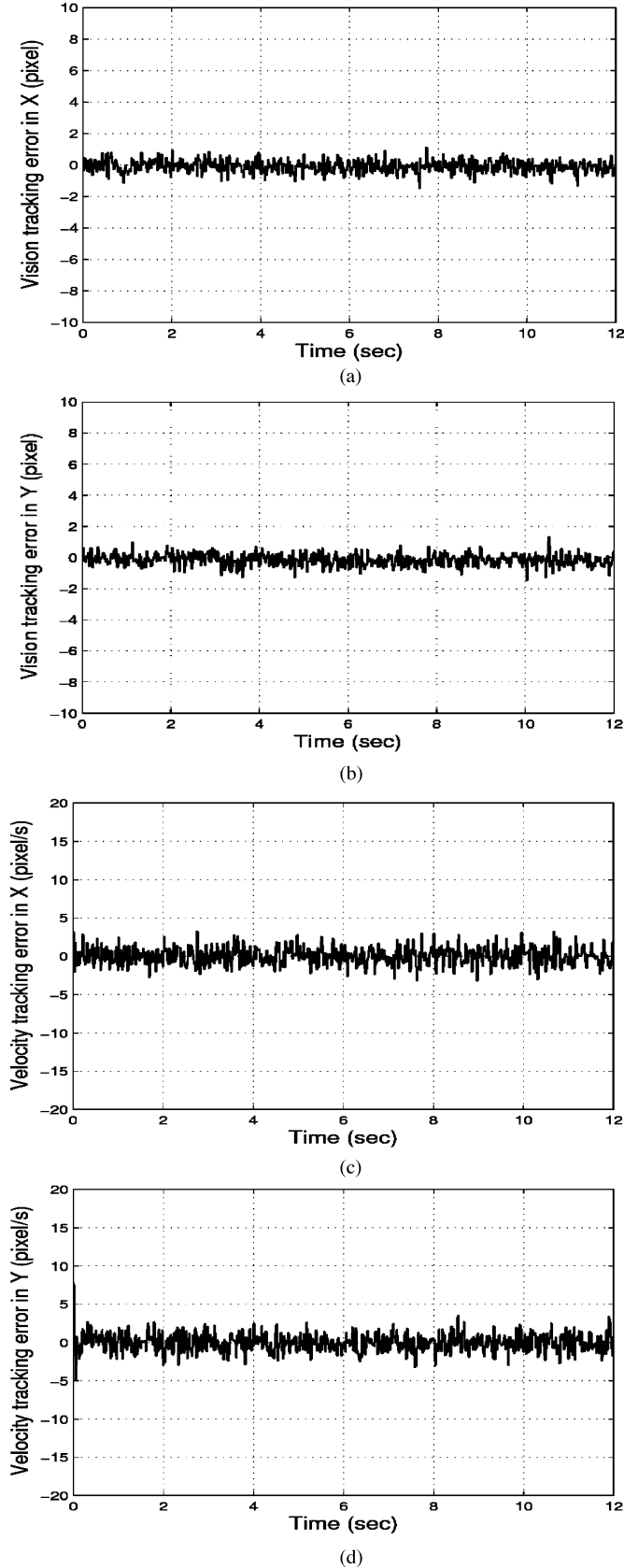


Fig. 7. Position and velocity tracking errors for the curve surface.

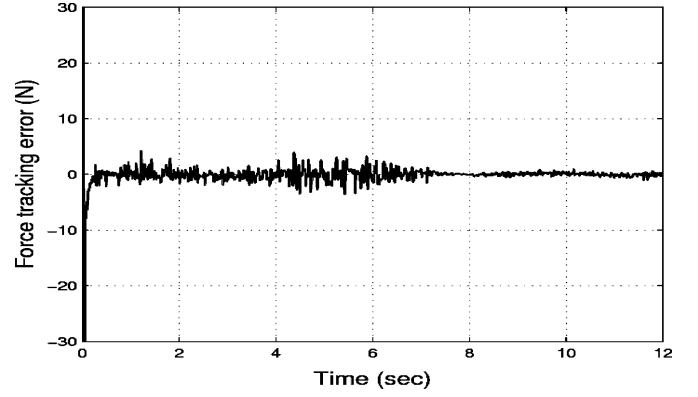


Fig. 8. Force tracking error for the curve surface.

Next, \dot{x} can be written as the product of a known regressor matrix $Y_{ki}(q, \dot{q})$ and an unknown vector θ_{ki} where

$$Y_{ki}(q, \dot{q}) = \begin{bmatrix} -s_1 \dot{q}_1 & -s_{12}(\dot{q}_1 + \dot{q}_2) & 0 & 0 \\ 0 & 0 & c_1 \dot{q}_1 & c_{12}(\dot{q}_1 + \dot{q}_2) \end{bmatrix}$$

and $\theta_{ki} = [(f_c/(z_1 - f_c))\beta_1 l_1, (f_c/(z_1 - f_c))\beta_1 l_2, (f_c/(z_1 - f_c))\beta_2 l_1, (f_c/(z_1 - f_c))\beta_2 l_2]^T$.

In this simulation, we set the parameters of the function of the constraint surface as $c = -0.52$ and $\gamma_s = 0$. The desired motion trajectory is set as $x_d(t) = 520$ pixels and $y_d(t) = 199 + 20t$ pixels. The desired contact force is set as $15 + 5 \sin(2t)$ Newton. The initial estimated parameters are set as $\hat{l}_1 = 0.4$ m, $\hat{l}_2 = 0.5$ m, $\hat{f}_c = 40$ mm, $\hat{z}_1 = 0.5$ m, and $\hat{\beta}_1 = \hat{\beta}_2 = 8000$, respectively. The control gains are set as $\alpha = 0.42$, $L_{ki} = 1/500I$, $L_d = 1/500I$, $L_f = 1/500I$, $L_{md} = 1/500I$, $K = 300/10^6 I$, $\beta = 0.0105$, $\gamma = 15$, and $\hat{\gamma}_s = 0.3$. The simulation results are shown in Figs. 4–6.

B. Curve Surface

In this simulation, we consider a constraint surface described by

$$\Psi_1(q(r)) = \sin(ar_x + b) - r_y = 0.$$

Note that in the previous constraint function, the parameters in $\Psi_1(r(q))$ cannot be linearly separated. The camera is also tilted by 15° so that the depth information z_1 is time varying.

The parameters of the constraint surface are set as $a = 3.1337$ and $b = -1.4292$. The control gains are set as $\alpha = 25$, $L_{ki} = 1/50I$, $L_d = 1/100I$, $L_f = 1/500I$, $L_{md} = 1/100I$, $K = 400/10^6 I$, $\beta = 0.25$, and $\gamma = 8$. An image path in image space is obtained from the camera. The initial position of the end-effector on the path is set at $(520, 199)$. $x_d(t)$ is set as $x_d(t) = 520 + 5t$ pixels and $y_d(t)$ is obtained from the image path. The desired contact force is set as $15 + 5 \sin(2t)$ Newton. In this simulation, Gaussian radial basis function (RBF) neural networks were used. The centers were chosen so that they were evenly distributed to span the input space of the network. The distance of neural networks was fixed at 1.8 and the number of neurons was set as 40. As can be seen from Figs. 7–9, the

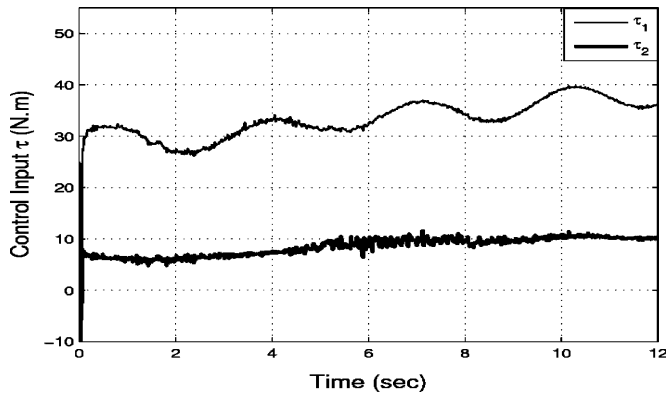


Fig. 9. Control inputs for the curve surface.

proposed controller is effective in dealing with uncertain structure of the constraint surface and Jacobian matrices.

V. CONCLUSION

In this paper, we have presented a visually servoed motion and force tracking controller with uncertain constraint surface, kinematics, dynamics, and camera model. A Lyapunov-like function has been presented for the stability analysis of the closed-loop control systems. We have shown that the robot end-effector can track the desired motion and force trajectories in the presence of the uncertainties. Simulation results have been presented to illustrate the performance of the proposed control law.

REFERENCES

- [1] N. Hogan, "Impedance control: An approach to manipulation: Part I-theory; part II-implementation; part III-applications," *Trans. ASME, J. Dyn. Syst., Meas. Control*, vol. 107, pp. 1–24, 1985.
- [2] M. H. Raibert and J. J. Craig, "Hybrid position/force control of manipulators," *ASME J. Dyn. Syst. Meas. Control*, vol. 102, pp. 126–133, 1981.
- [3] T. Yoshikawa, "Force control of robot manipulators," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2000, pp. 220–226.
- [4] N. H. McClamroch and D. Wang, "Feedback stabilization and tracking of constrained robots," *IEEE Trans. Autom. Control*, vol. 33, no. 5, pp. 419–426, May 1988.
- [5] T. Yoshikawa, T. Sugie, and M. Tanaka, "Dynamic hybrid position/force control of robot manipulators: Controller design and experiment," *IEEE J. Robot. Autom.*, vol. 4, no. 6, pp. 699–705, Dec. 1988.
- [6] S. Arimoto, T. Naniwa, and Y. H. Liu, "Model-based adaptive hybrid control for manipulators with geometric endpoint constraint," *Adv. Robot.*, vol. 9, no. 1, pp. 67–80, 1995.
- [7] R. Carelli and R. Kelly, "An adaptive impedance/force controller for robot manipulators," *IEEE Trans. Autom. Control*, vol. 36, no. 8, pp. 967–971, Aug. 1991.
- [8] V. Parra-Vega and S. Arimoto, "A passivity-based adaptive sliding mode position-force control for robot manipulators," *Int. J. Adaptive Control Signal Process.*, vol. 10, pp. 365–377, 1996.
- [9] L. L. Whitcomb, S. Arimoto, T. Naniwa, and F. Ozaki, "Adaptive model-based hybrid control of geometrically constrained robot arms," *IEEE Trans. Robot. Autom.*, vol. 13, no. 1, pp. 105–116, Feb. 1997.
- [10] T. Yoshikawa and A. Sudou, "Dynamic hybrid position/force control of robot manipulators: Online estimation of unknown constraint," *IEEE Trans. Robot. Autom.*, vol. 9, no. 2, pp. 220–226, Apr. 1993.
- [11] M. Namvar and F. Aghili, "Adaptive force-motion control of coordinated robots interacting with geometrically unknown environments," *IEEE Trans. Robot.*, vol. 21, no. 4, pp. 678–694, Aug. 2005.
- [12] B. J. Nelson and P. K. Khosla, "Force and vision resolvability for assimilating disparate sensory feedback," *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 714–731, Oct. 1996.
- [13] G. Morel, E. Malis, and S. Boudet, "Impedance based combination of visual and force control," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1998, pp. 1743–1748.
- [14] K. Hosoda, K. Igarashi, and M. Asada, "Adaptive hybrid control for visual and force servoing in an unknown environment," *IEEE Robot. Autom. Mag.*, vol. 5, no. 4, pp. 39–43, Dec. 1998.
- [15] J. Baeten, H. Bruyninckx, and J. D. Schutter, "Integrated vision/force robotic servoing in the task frame formalism," vol. 22, pp. 941–954, 2003.
- [16] D. Xiao, B. K. Ghosh, N. Xi, and T. J. Tarn, "Sensor-based hybrid position/force control of a robot manipulator in an uncalibrated environment," *IEEE Trans. Control Syst. Technol.*, vol. 8, no. 4, pp. 635–645, Jul. 2000.
- [17] S. Arimoto, "Robotics research toward explication of everyday physics," *Int. J. Robot. Res.*, vol. 18, no. 11, pp. 1056–1063, Nov. 1999.
- [18] C. C. Cheah, C. Liu, and J. J. E. Slotine, "Adaptive tracking control for robots with unknown kinematic and dynamic properties," *Int. J. Robot. Res.*, vol. 25, no. 3, pp. 283–296, 2006.
- [19] C. C. Cheah, S. Kawamura, and S. Arimoto, "Feedback control for robotic manipulator with an uncertain Jacobian matrix," *J. Robot. Syst.*, vol. 2, no. 12, pp. 119–134, 1999.
- [20] C. C. Cheah, M. Hirano, S. Kawamura, and S. Arimoto, "Approximate Jacobian control for robots with uncertain kinematics and dynamics," *IEEE Trans. Robot. Autom.*, vol. 19, no. 4, pp. 692–702, Aug. 2003.
- [21] W. E. Dixon, "Adaptive regulation of amplitude limited robot manipulators with uncertain kinematics and dynamics," *IEEE Trans. Autom. Control*, vol. 52, no. 3, pp. 488–493, Mar. 2007.
- [22] Y. Zhao and C. C. Cheah, "Hybrid vision-force control for robot with uncertainties," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2004, pp. 261–266.
- [23] Z. Douglari and S. Arimoto, "A position/force control for a robot finger with soft tip and uncertain kinematics," *J. Robot. Syst.*, vol. 19, no. 3, pp. 115–131, 2002.
- [24] C. C. Cheah, S. Kawamura, and S. Arimoto, "Stability of hybrid position and force control for robotic manipulator with kinematics and dynamics uncertainties," *Automatica*, vol. 39, pp. 847–855, 2003.
- [25] Y. Nakamura, *Advanced Robotics-Redundancy and Optimization*. Reading, MA: Addison-Wesley, 1991.
- [26] S. Arimoto, *Control Theory of Nonlinear Mechanical Systems: A Passivity-Based and Circuit-Theoretic Approach*. Oxford, U.K.: Clarendon, 1996.
- [27] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 651–670, Oct. 1996.
- [28] L. E. Weiss, A. C. Sanderson, and C. P. Neuman, "Dynamic sensor-based control of robots with visual feedback," *IEEE J. Robot. Autom.*, vol. 3, no. 5, pp. 404–417, Oct. 1987.
- [29] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robot. Autom.*, vol. 8, no. 3, pp. 313–326, Jun. 1992.
- [30] 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 vision and control," *IEEE Trans. Robot. Autom.*, vol. 9, no. 1, pp. 14–35, Feb. 1993.
- [31] M. Bonkovic, A. Haze, and K. Jezernik, "Population-based uncalibrated visual servoing," *IEEE/ASME Trans. Mechatronics*, vol. 13, no. 3, pp. 393–397, Jun. 2008.
- [32] B. Siciliano and L. Villani, *Robot Force Control*. Norwell, MA: Kluwer, 1999.
- [33] G. Niemeyer and J. J. E. Slotine, "Performance in adaptive manipulator control," *Int. J. Robot. Res.*, vol. 10, no. 2, pp. 149–161, 1991.
- [34] J. J. E. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice-Hall, 1991.
- [35] H. K. Khalil, *Nonlinear Systems*. Englewood Cliffs, NJ: Prentice-Hall, 2002.
- [36] C. C. Cheah, H. Y. Han, S. Kawamura, and S. Arimoto, "Grasping and position control for multi-fingered robot hands with uncertain Jacobian matrices," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1998, pp. 2403–2408.
- [37] W. Gueaieb, F. Karray, and S. Al Sharhan, "A robust hybrid intelligent position/force control scheme for cooperative manipulators," *IEEE/ASME Trans. Mechatronics*, vol. 12, no. 2, pp. 109–125, Apr. 2007.
- [38] F. Caccavale, P. Chiacchio, A. Marino, and L. Villani, "Six-dof impedance control of dual-arm cooperative manipulators," *IEEE/ASME Trans. Mechatronics*, vol. 13, no. 5, pp. 576–586, Oct. 2008.
- [39] R. M. Sanner and J. E. Slotine, "Gaussian networks for direct adaptive control," *IEEE Trans. Neural Netw.*, vol. 3, no. 6, pp. 837–863, Nov. 1992.

- [40] F. L. Lewis, "Neural network control of robot manipulators," *IEEE Expert*, vol. 11, no. 3, pp. 64–75, Jun. 1996.
- [41] E. Malis and P. Rives, "Robustness of image-based visual servoing with respect to depth distribution errors," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2003, pp. 1056–1061.
- [42] C. C. Cheah, C. Liu, and J. Slotine, "Adaptive vision based tracking control of robots with uncertainty in depth information," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 2817–2822.
- [43] F. Schramm and G. Morel, "Ensuring visibility in calibration-free path planning for image-based visual servoing," *IEEE Trans. Robot.*, vol. 22, no. 4, pp. 848–854, 2006.
- [44] E. Malis, "Visual servoing invariant to changes in camera intrinsic parameters," *IEEE Trans. Robot. Autom.*, vol. 20, no. 1, pp. 71–82, Feb. 2004.
- [45] S. P. Hou and C. C. Cheah, "Region tracking control for robot manipulators," in *Proc. IEEE Int. Conf. Control Appl.*, 2007, pp. 1438–1443.
- [46] C. C. Cheah, D. Q. Wang, and Y. C. Sun, "Region reaching control of robots," *IEEE Trans. Robot.*, vol. 23, no. 6, pp. 1260–1264, Dec. 2007.



Chien Chern Cheah (S'92–M'93–SM'03) was born in Singapore. He received the B.Eng. degree in electrical engineering from the National University of Singapore, Singapore, in 1990, and the M.Eng. and Ph.D. degrees in electrical engineering from Nanyang Technological University, Singapore, in 1993 and 1996, respectively.

From 1990 to 1991, he was a Design Engineer with Chartered Electronics Industries, Singapore. From 1996 to 1998, he was a Research Fellow in the Department of Robotics, Ritsumeikan University, Japan.

In 1998, he joined, as an Assistant Professor, the School of Electrical and Electronic Engineering, Nanyang Technological University, where he has been an Associate Professor since 2003. In November 2002, he received the Oversea Attachment Fellowship from the Agency for Science, Technology and Research (A*STAR), Singapore, to visit the Nonlinear Systems Laboratory, Massachusetts Institute of Technology, Cambridge.

Dr. Cheah was the Program Chair of the International Conference on Control, Automation, Robotics and Vision, 2006. Since 2007, he has been an Associate Editor of the IEEE Robotics and Automation Society Conference Editorial Board.



Saing Paul Hou (S'09) was born in Kandal, Cambodia. He received the B.Eng. degree (with first-class honors) in electrical and electronic engineering in 2006 from Nanyang Technological University, Singapore, where he is currently working toward the Ph.D. degree.

His current research interests include formation control of multirobot systems and adaptive control.

Mr. Hou was the recipient of the Control Chapter Book Prize and the Motorola Book Prize in 2006.



Yu Zhao was born in China. He received the B.Eng. degree in automation from Beijing University of Aeronautics and Astronautics, Beijing, China, in 2002, and the Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore, in 2007.

He is currently an I&C System Engineer at OGS Industries, Singapore. His current research interests include adaptive vision–force control of robot manipulators and multifingered robot hands using neural networks.



Jean-Jacques E. Slotine was born in Paris, France, in 1959. He received the Ph.D. degree from Massachusetts Institute of Technology (MIT), Cambridge, in 1983.

He was with the Computer Research Department, Bell Laboratories. In 1984, he joined the faculty at MIT, where he is currently a Professor of mechanical engineering and information sciences, a Professor of brain and cognitive sciences, and the Director of the Nonlinear Systems Laboratory. He is the coauthor of the textbooks *Robot Analysis and Control* (Wiley, 1986) and *Applied Nonlinear Control* (Prentice-Hall, 1991).

Prof. Slotine was a member of the French National Science Council from 1997 to 2002 and the Agency for Science, Technology and Research (A*STAR) Sign Advisory Board, Singapore.