2) The determined calibration plane can in turn be used to detect errors associated with the range images of a 3-D LIDAR.
3) Both a polynomial model and a polynomial-ARV model have been proposed in this new method to compensate for the range image errors.

The experimental results indicate that the compensation based on the polynomial model can reduce the range image errors from 163 counts to 18 counts. The polynomial-ARV model can further reduce the range image error by more than seven counts by considering the dynamic characteristics imbedded in the range image errors.

REFERENCES

[1] P. J. Besl, "Active, optical range imaging sensors," *Machine Vision and Application,* no. 1, pp. 127–152, 1988.
[2] F. Blais, M. Rioux, and J. A. Beraldin, "Practical consideration for a design of a high precision 3-D laser scanner system," *Optomechanical and Electro-Optical Design of Industrial Systems,* SPIE, vol. 958, pp. 225–246, 1988.
[3] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM,* vol. 24, no. 6, pp. 381–395, June 1981.
[4] M. A. Fischler, *Readings in Computer Vision : Issues, Problems, Principles and Paradigms.* San Mateo, CA: Morgan Kaufmann, 1987.
[5] R. C. Gonzalez and R. Safabakhsh, "Computer vision techniques for industrial application and robot control," *Computer,* pp. 17–32, 1982.
[6] K. D. Gremban and L. A. Fedkamp, "Camera calibration methodology based on a linear perspective transformation error model," in *Proc. 1988 IEEE Int. Conf. Robotics Automat.*.
[7] R. Krishnapuram and D. Casasent "Determination of 3-D object location and orientation from range image," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 11, pp. 1158–1167, Nov. 1989.
[8] R. K. Lenz and R. Y. Tsai, "Techniques for calibration of scale factor and image center for high accuracy 3-D machine vision metrology," *IEEE Trans. Trans. Pattern Anal. Machine Intell.,* vol. 10, pp. 713–720, Sept. 1988.
[9] Y. C. Liu and T. S. Huang, "Estimation of rigid body motion using straight line correspondences," in *Proc. IEEE Workshop Motion,* Kaiwah Island, SC, 1986, pp. 47–53.
[10] W. D. Mcfarland, "Three dimensional images for robot vision," in *Robotics and Robotic Sensing System: Proc. Int. Soc. Opt. Eng.,* vol. 442, 1983, pp. 108–116.
[11] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3-D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE J. Robotics Automat.,* vol. RA-3, Aug. 1987.

Regressor Formulation of Robot Dynamics: Computation and Applications

W.-S. Lu and Q.-H. Meng

***Abstract*—Two approaches to the evaluation of the manipulator regressor of a general n-degree-of-freedom (DOF) robot are presented. The first method is an "energy-based" approach using Lagrangian formulation of robot dynamics as a starting point. A key fact used in deriving the solution is that the manipulator Lagrangian is linkwise additive. The second approach generates an iterative algorithm for efficient numerical evaluation of the regressor. It is obtained by reformulating the Newton–Euler recursion using vector analysis type techniques. In addition, a modified Slotine–Li algorithm for adaptive motion control is presented and is then applied in a simulation study to a 4-DOF PUMA-type robot, where the manipulator regressor is evaluated using the iterative algorithm proposed.**

I. INTRODUCTION

The manipulator regressor, often denoted by $Y(q, \dot{q}, \dot{q})$,is a key quantity in derivation as well as implementation of the many established adaptive motion and force control algorithms [1], [2]. This is because its availability enables one to express the dynamics of a robot arm as $Y\theta = \tau$ with $\theta \in R^r$ representing the manipulator parameters, thus a Lyapunov approach may lead to a *linear* law for updating the parameters. Studies on this linear parameter-dependence issue from an identification point of view can be found in [3]–[5] among others.

In principle, the regressor can be obtained by using a two-step approach. The first step is to formulate the manipulator dynamics as

$$H(q)\dot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau. \quad (1)$$

This can be accomplished, for example, by using the Newton–Euler or Lagrange formulation, see [6, ch. 6] for the details of these formulations and their computation complexity. Having done this, the second step of the approach defines a parameter vector θ and then works on every entry on the left-hand side of (1) to extract vector θ, leading (1) to the regressor formulation $Y\theta = \tau$. So we see computationally that this is an indirect approach that requires formulating (1) plus a parameter extraction procedure. As the entries of θ are, in general, spread over all the entries of $H(q), C(q, \dot{q})$, and $G(q)$, the second step is also computationally complicated.

In this paper, we propose two methods that compute the regressor of a general n-degree-of-freedom (DOF) robot without using (1). Our first method provides a closed-form solution, which is obtained by extracting parameter θ from link Lagrangians during the Lagrangian formulation; our second method gives a recursive-type solution, which is obtained by extracting parameter θ from joint velocities, accelerations, forces, and torques during the Newton–Euler formulation. As opposed to the conventional two-step approach, in which one derives (1) with the entries of θ spread widely over the terms and then extracts these parameters term by term, the proposed

Manuscript received May 15, 1990; revised May 4, 1992. A portion of this work was presented at the Canadian Conference on Electrical and Computer Engineering, Ottawa, Canada, September 1990, and at the IEEE Pacific Rim Conference, Victoria, BC, Canada, May 1991. This work was supported in part by the Networks of Excellence Program of the Government of Canada and in part by PRECARN Associates Inc.

W.-S. Lu is with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, V8W 3P6 Canada.

Q.-H. Meng is with the Department of Electrical Engineering, Lakehead University, ON, Canada.

IEEE Log Number 9207364.

methods perform the parameter extraction and dynamics formulation simultaneously and, therefore, more efficiently. Furthermore, unlike the two-step approach, which defines parameter vector θ by trial-and-error until θ can be extracted from every term on the left-hand side of (1) (see the example in Section III-C for details), the formation of θ in the proposed methods has an explicit rule to follow. It is found that, if for a specified link the mass, mass center, and inertia tensor are the parameters to be extracted, then this portion of vector θ has dimension 16 although the number of physical parameters related to the link is 10. In addition, the so-called filtered regressor adopted in several adaptive motion control algorithms [7], [8] can readily be obtained from the proposed formula.

The paper is organized as follows. Preliminaries on the manipulatorregressor and its relation to link Lagrangians are discussed in Section II. In Section III, we present a closed-form solution to the regressor evaluation problem. Special cases for robots having point-mass links (or load) and links with regular geometry are addressed. In Section IV, we propose an iterative algorithm for evaluating the manipulator regressor. The proposed methods are applied to obtain the regressor of a 2-DOF robot grasping a non-point-mass object. In Section V, a modified version of the Slotine–Li algorithm for adaptive motion control [5] is described, and its stability is shown via a Lyapunov approach. The proposed algorithm is then applied in a simulation study to a simplified 4-DOF PUMA-type robot, where the major steps of the control algorithm are expressed in terms of the regressor, which is evaluated using the proposed recursive algorithm.

II. Preliminaries

A. $Y_u(q, \dot{q}, \dot{q})$—*Regressor Associated with Unknown Parameters*

It is known [1], [5] that (1) can be written in the form of

$$Y(q, \dot{q}, \dot{q})\theta = \tau, \tag{2}$$

where $Y(q, \dot{q}, \dot{q}) \in R^{n\times r}$ is the manipulator regressor and $\theta \in R^{r\times 1}$ is the vector formed by the dynamic parameters of the manipulator in a certain manner. Denoting

$$\theta = \begin{bmatrix} \theta_k \\ \theta_u \end{bmatrix} \tag{3}$$

with $\theta_k \in R^{r_1}$ and $\theta_u \in R^{r_2}$ representing the known and unknown parameters, respectively, and

$$Y = [Y_k \; Y_u] \tag{4}$$

with $Y_k \in R^{n\times r_1}$ and $Y_u \in R^{n\times r_2}$, (1) and (2) imply that

$$H\dot{q} + C\dot{q} + G = Y_k\theta_k + Y_u\theta_u. \tag{5}$$

If $\hat{\theta}_u$ is an estimate of θ_u, then

$$\hat{H}\dot{q} + \hat{C}\dot{q} + \hat{G} = Y_k\theta_k + Y_u\hat{\theta}_u, \tag{6}$$

where $\hat{H}, \hat{C}$, and $\hat{G}$ assume the same forms as H, C, and g, respectively, with θ_u replaced by $\hat{\theta}_u$. It follows that

$$\tilde{H}\dot{q} + \tilde{C}\dot{q} + \tilde{G} = Y_u\tilde{\theta}_u, \tag{7}$$

where $(\tilde{*}) = (\hat{*}) - (*)$. It is (7) that plays a role in the establishment of the many stable algorithms for adaptive control of robots.

B. *Relation of* $Y(q, \dot{q}, \dot{q})$ *to the Lagrangians of Manipulator Links and Load*

Consider an n-DOF manipulator grasping firmly a non-point-mass load. Denote by $k^{(i)}$ and $u^{(i)} (1 \le i \le n)$ the kinetic and potential energy of link i, and by $k^{(n+1)}$ and $u^{(n+1)}$ the kinetic and potential energy of the load, respectively. If the load is treated as link $n+1$ and the link Lagrangian of link i is defined as

$$\mathcal{L}^{(i)} = k^{(i)} - u^{(i)}, \qquad 1 \le i \le n+1, \tag{8}$$

then the manipulator Lagrangian is

$$\mathcal{L} = \sum_{i=1}^{n+1} \mathcal{L}^{(i)}. \tag{9}$$

If follows that the manipulator Lagrangian is linkwise additive; that is, if a new link is added to the robot, its manipulator Lagrangian is then equal to the original $\mathcal{L}$ plus the Lagrangian of the new link. This property turns out to be a key fact in the subsequent derivation of the regressor as it allows one to separate the parameters of a specific link from the parameters of other links. From (2), (9), and the Lagrange's equation of motion

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}}{\partial q} = \tau, \tag{10}$$

we obtain

$$\sum_{i=1}^{n+1}\left[\frac{d}{dt}\left(\frac{\partial \mathcal{L}^{(i)}}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}^{(i)}}{\partial q}\right] = Y(q, \dot{q}, \dot{q})\theta. \tag{11}$$

Now if θ is partitioned as

$$\theta = \begin{bmatrix} \theta^{(1)} \\ \theta^{(2)} \\ \vdots \\ \theta^{(n+1)} \end{bmatrix} \tag{12}$$

with $\theta^{(i)}$ representing the dynamic parameters of link i, that is, its mass, mass center, and inertia tensor, and if Y is partitioned into $n+1$ blocks

$$Y = [Y^{(1)} \quad Y^{(2)} \quad \cdots \quad Y^{(n+1)}] \tag{13}$$

with dimensions consistent to (12), then

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}^{(i)}}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}^{(i)}}{\partial q} = Y^{(i)}\theta^{(i)}, \qquad 1 \le i \le n+1. \tag{14}$$

In the next section, (14) will be used to obtain a formula of $Y^{(i)}$.

Once Y is computed, Y_k and Y_u in (4) can readily be foundas follows. Vector θ in (12) can be regrouped as in (3), where θ_u is formed by collecting the entries, each of which involves at least one of the unknown parameters, and θ_k is simply the complement of θ_u in θ. Obviously, this regrouping can be done by premultiplying θ by an elementary transformation matrix T, that is,

$$T\theta = \begin{bmatrix} \theta_k \\ \theta_u \end{bmatrix}_{(r_1+r_2)\times 1}. \tag{15}$$

From

$$Y\theta = YT^TT\theta = YT^T\begin{bmatrix} \theta_k \\ \theta_u \end{bmatrix} \tag{16}$$

it follows that

$$Y_k = \text{first } r_1 \text{ columns of} YT^T \tag{17}$$

and

$$Y_u = \text{last } r_2 \text{ columns of} YT^T. \tag{18}$$

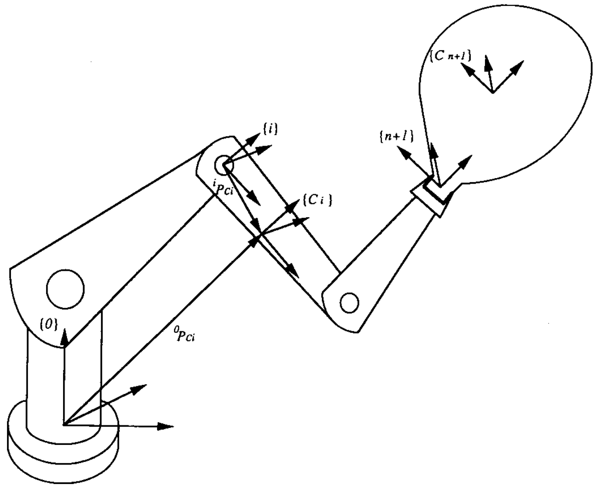


Fig. 1. Frame setup for link i.

C. Two Identities from Vector Analysis

The identities

$$(a \times b)^T c = (b \times c)^T a = (c \times a)^T b \tag{19}$$

and

$$(a \times b) \times c = (c^T a) b - (b^T c) a \tag{20}$$

with $a, b, c \in R^{3\times 1}$ are well known in vector analysis. In Appendix I, we consider several expressions that will be used in the subsequent sections and show how identities (19) and (20) can be applied to extract the unknown parameters from the expressions.

III. Formula of $Y^{(i)}(q, \dot{q}, \dot{q})$

As is shown in Fig. 1, denote by $\{i\}$ the frame attached to link i according to the Denavit–Hartenberg convention, and by $\{c_i\}$ the frame attached to the mass center of the link, which differs from $\{i\}$ only by a translation. Denote by m, ${}^ip_{c_i} = [p_x \ p_y \ p_z]^T$, and ${}^cI_i$ the mass, mass center, and inertia tensor of the link, respectively, where ${}^ip_{c_i}$ is expressed in $\{i\}$, and ${}^cI_i$ is expressed with respect to $\{c_i\}$ and is explicitly given by (105). Further denote by ${}^0_iR$ the rotation matrix describing frame $\{i\}$ in the world frame $\{0\}$ and by $J^{(i)} \in R^{6\times n}$ the link Jacobian expressed in frame $\{0\}$, which will often be partitioned into $J_v^{(i)} \in R^{3\times n}$—the Jacobian associated with the translational velocity of $\{i\}$ and $J_\omega^{(i)} \in R^{3\times n}$—the Jacobian associated with the rotational velocity of $\{i\}$, that is,

$$J^{(i)} = \begin{bmatrix} J_v^{(i)} \\ J_\omega^{(i)} \end{bmatrix}.$$

For the sake of simplicity in what follows the superindex i in the Jacobian will be omitted. In the sequel, J_v and J_ω will sometimes be used in the form

$$\begin{aligned} J_v &= [J_{v1} \quad J_{v2} \quad \cdots \quad J_{vn}], \\ J_\omega &= [J_{\omega 1} \quad J_{\omega 2} \quad \cdots \quad J_{\omega n}], \end{aligned} \tag{21}$$

where J_{vj} and $J_{\omega j}$ are the jth columns of J_v and J_ω, respectively.

A. Formula

By (8), the left-hand side of (14) is equal to

$$\frac{d}{dt}\left(\frac{\partial k^{(i)}}{\partial \dot{q}}\right) - \frac{\partial k^{(i)}}{\partial q} + \frac{\partial u^{(i)}}{\partial q}, \tag{22}$$

where the kinetic energy of link i is given by

$$k^{(i)} = \frac{1}{2} m \|{}^0v_{c_i}\|^2 + \frac{1}{2}\, {}^i\omega_i^T \tag{23}$$

with

$${}^0v_{c_i} = J_v \dot{q} + J_\omega \dot{q} \times {}^0_iR p_{c_i} \tag{24}$$

and

$${}^i\omega_i = {}^i_0R J_\omega \dot{q}, \tag{25}$$

and the potential energy of the link is given by

$$u^{(i)} = -m\, {}^0p_{c_i}^T\, {}^0g \tag{26}$$

with ${}^0g$ the gravity vector and ${}^0p_{c_i}$ the position vector from the origin of frame $\{0\}$ to the link's mass center, as is shown in Fig. 1.

From (23)–(25) it follows that

$$\frac{\partial k^{(i)}}{\partial \dot{q}} = m \frac{\partial\, {}^0v_{c_i}^T}{\partial \dot{q}}\, {}^0v_{c_i} + J_\omega^T\, {}^0_iR\, {}^cI_i\, {}^i_0R J_\omega \dot{q}. \tag{27}$$

With ${}^0v_{c_i}$ given by (24), (126) can be used to express the first term in (27) as $W_1(q, \dot{q})\theta_5$ with $W_1(q, \dot{q})$ defined by (127) and θ_5 defined by (125). Furthermore, let

$$d = {}^i_0R J_\omega \dot{q} \equiv [d_1 \quad d_2 \quad d_3]^T. \tag{28}$$

Formula (107) gives

$$J_\omega^T\, {}^0_iR\, {}^cI_{c_i}\, {}^i_0R J_\omega^T \dot{q} = J_\omega^T\, {}^0_iR B(d)\theta_3 \equiv W_2(q, \dot{q})\theta_3, \tag{29}$$

where θ_3 is defined by (106) and $B(d)$ is given by (108). Equation (27) can now be written as

$$\frac{\partial k^{(i)}}{\partial \dot{q}} = [W_1(q, \dot{q}) \quad W_2(q, \dot{q})]\theta^{(i)} \tag{30}$$

with

$$\theta^{(i)} = \begin{bmatrix} \theta_5 \\ \theta_3 \end{bmatrix}. \tag{31}$$

By (23)–(25) the second term in (22) becomes

$$-\frac{\partial k^{(i)}}{\partial q} = -m \frac{\partial\, {}^0v_{c_i}^T}{\partial q}\, {}^0v_{c_i} - \frac{\partial (\dot{q}^T J_\omega^T\, {}^0_iR)}{\partial q}\, {}^cI_i\, {}^i_0R J_\omega \dot{q}. \tag{32}$$

By repeatedly using (19) and (20), it is found that

$$\begin{aligned} -m \frac{\partial\, {}^0v_{c_i}^T}{\partial q}\, {}^0v_{c_i} &= [Y_{11}(q, \dot{q}) \quad Y_{12}(q, \dot{q}) \quad Y_{13}(q, \dot{q}) \quad Y_{14}(q, \dot{q})]\theta_5 \\ &\equiv Y_1(q, \dot{q})\theta_5, \end{aligned} \tag{33}$$

where

$$Y_{11}(q, \dot{q}) = -D_v^T J_v \dot{q} \tag{34}$$

$$Y_{12}(q, \dot{q}) = -[D_v \times J_\omega \dot{q} + J_v \dot{q} \times D_\omega]^T\, {}^0_iR - S \tag{35}$$

$$Y_{13}(q, \dot{q}) = -\left[\frac{\partial J_\omega^T}{\partial q_1} J_\omega \dot{q} \cdots \frac{\partial J_\omega^T}{\partial q_n} J_\omega \dot{q}\right]^T \dot{q} \tag{36}$$

$$D_v = \left[\frac{\partial J_v}{\partial q_1}\dot{q} \cdots \frac{\partial J_v}{\partial q_n}\dot{q}\right], \qquad D_\omega = \left[\frac{\partial J_\omega}{\partial q_1}\dot{q} \cdots \frac{\partial J_\omega}{\partial q_n}\dot{q}\right] \tag{37}$$

$$S = \left[\frac{\partial\, {}^i_0R}{\partial q_1}(J_v \dot{q} \times J_\omega \dot{q}) \cdots \frac{\partial\, {}^i_0R}{\partial q_n}(J_v \dot{q} \times J_\omega \dot{q})\right] \tag{38}$$

and $Y_{14}(q, \dot{q})$ is determined by

$$\begin{bmatrix} {}^ip_{c_i}^T E_1\, {}^ip_{c_i} \\ \vdots \\ {}^ip_{c_i}^T E_n\, {}^ip_{c_i} \end{bmatrix} = Y_{14}(q, \dot{q})\theta_4, \tag{39}$$

where θ_4 is defined by (112), and

$$E_j = {}^i_0RJ_\omega\dot{q}\dot{q}^T\frac{\partial J_\omega^T}{\partial q_j}{}^0_iR - \frac{\partial {}^i_0R}{\partial q_j}(\|J_\omega\dot{q}\|^2I_3 - J_\omega\dot{q}\dot{q}^TJ_\omega^T){}^0_iR. \quad (40)$$

With d defined by (28) and (107), the second term in (32) can be written as

$$-\frac{\partial(\dot{q}^TJ_\omega^T{}^0_iR)}{\partial q}{}^cI_i{}^i_0RJ_\omega\dot{q} = Y_2(q,\dot{q})\theta_3 \quad (41)$$

where

$$Y_2(q,\dot{q}) = -\left[\frac{\partial({}^i_0RJ_\omega)}{\partial q_1}\dot{q}\cdots\frac{\partial({}^i_0RJ_\omega)}{\partial q_n}\dot{q}\right]B(d). \quad (42)$$

To compute the last term in (22), note that

$${}^0v_{c_i} = \frac{d}{dt}({}^0p_{c_i}) = \left(\frac{\partial{}^0p_{c_i}^T}{\partial q}\right)^T\dot{q}$$

which, in conjunction with (24) and (99), leads to

$$-m\dot{q}^T\frac{\partial{}^0p_{c_i}^T}{\partial q}{}^0g = \dot{q}^TY_3(q)\theta_2, \quad (43)$$

where θ_2 is defined by (101) and

$$Y_3(q) = [Y_{31}(q) \quad Y_{32}(q)] \quad (44)$$
$$Y_{31}(q) = -J_v^T{}^0g \quad (45)$$
$$Y_{32}(q) = (J_\omega \times {}^0g)^T{}^0_iR. \quad (46)$$

Using (14), (22), (30), (32), (33), (41), and (43), we obtain

$$Y^{(i)} = [\dot{W}_1(q,\dot{q}) + \hat{Y}_1(q,\dot{q}) \quad \dot{W}_2(q,\dot{q}) + Y_2(q,\dot{q})] \quad (47)$$

where

$$\hat{Y}_1(q,\dot{q}) = [Y_{11}+Y_{31} \quad Y_{12}+Y_{32} \quad Y_{13} \quad Y_{14}]. \quad (48)$$

Remarks

1) From (47) and (31) it is observed that, although there are only ten physical parameters involved in a link or load, in general, the dimension of $\theta^{(i)}$ is 16. This parameter redundancy, presented in vector θ_5 which is defined by(125), appears to be necessary to reform the dynamics so that a linear appearance of vector$\theta^{(i)}$ in the dynamics is achieved. Conversely, if the link(load) has a regular geometry, the dimension of $\theta^{(i)}$ will very likely be reduced, leading to a simplified solution. Additional discussion on this issue will be given in the next subsection.
2) Another feature of (47) is that the formula as it stands is suitable to serve as a starting point to derive a closed-form solution for the so-called filtered regressor that has been used in several globally stable adaptive control algorithms [7], [8].This is due to the fact that in (47), $Y^{(i)}$ depends implicitly on $\ddot{q}$ through the time derivative of W_1 and W_2.

B. Special Cases

1) Point-Mass Link (or Load): If the link (or load) can be treated as a point mass, then $\theta^{(i)} = \theta_5$, which is defined by (125). Consequently, $Y^{(i)}$ is given by

$$Y^{(i)} = \dot{W}_1(q,\dot{q}) + \hat{Y}_1(q,\dot{q}). \quad (49)$$

2) When Axes of c_i are the Principal Axes of the Link: If link i has a regular geometry, such that the axes of c_i coincide with the principal axes of the link, that is,

$${}^cI_i = \mathrm{diag}\{I_{xx}, I_{yy}, I_{zz}\} \quad (50)$$

and if the mass center of the link lies on the x or z axis of frame c_i, then $\theta^{(i)}$ is reduced to a six-dimensional vector of the form shown in (51), at the bottom of the page, and expressions for matrices $W_1, W_2, \hat{Y}_1$, and Y_2 can be simplified considerably.

For ${}^ip_{c_i} = [p_x\,0\,0]^T$, matrix $W_1(q,\dot{q}) \in R^{n\times 3}$ becomes

$$W_1(q,\dot{q}) = [J_v^TJ_v\dot{q} \quad (J_v \times J_\omega\dot{q} + J_v\dot{q}\times J_\omega)^T{}^0_ir_1$$
$$J_\omega^TJ_\omega\dot{q} - b] \quad (52)$$

where ${}^0_ir_1$ denotes the first column of ${}^0_iR$, and b is defined by $b = [b_1\cdots b_n]^T$ with

$$b_j = {}^0_ir_1^TJ_{\omega_j}\dot{q}^TJ_\omega^T{}^0_ir_1, \qquad 1 \le j \le n. \quad (53)$$

Matrix $W_2(q,\dot{q})$ defined by (29) becomes a 3×3 matrix with $B(d) \in R^{3\times 3}$ given by

$$B(d) = \mathrm{diag}({}^i_0RJ_\omega\dot{q}) \quad (54)$$

that is, the 3×3 diagonal matrix with ${}^i_0RJ_\omega\dot{q}$ as entries along its main diagonal. Matrix $\hat{Y}_1$ becomes

$$\hat{Y}_1 = [Y_{11}+Y_{31} \quad y_{121}+y_{321} \quad Y_{13}+y_{141}] \quad (55)$$

where

$$y_{121} = -\left\{\left[\frac{\partial J_v}{\partial q_1}\dot{q}\cdots\frac{\partial J_v}{\partial q_n}\dot{q}\right]\times J_\omega\dot{q} + J_v\dot{q}\left[\frac{\partial J_\omega}{\partial q_1}\dot{q}\cdots\frac{\partial J_\omega}{\partial q_n}\dot{q}\right]\right\}^T$$
$$\cdot{}^0_ir_1 - \left[\frac{\partial{}^0_ir_1^T}{\partial q_1}(J_v\dot{q}\times J_\omega\dot{q})\cdots\frac{\partial{}^0_ir_1^T}{\partial q_n}(J_v\dot{q}\times J_\omega\dot{q})\right]^T \quad (56)$$
$$y_{321} = -(J_\omega \times {}^0g)^T{}^0_ir_1 \quad (57)$$
$$y_{141} = [e_{11}\cdots e_{1n}]^T \quad (58)$$
$$e_{1j} = {}^0_ir_1^TJ_\omega\dot{q}\dot{q}^T\frac{\partial J_\omega^T}{\partial q_j}{}^0_ir_1 - \frac{\partial{}^0_ir_1^T}{\partial q_j}$$
$$\cdot(\|J_\omega\dot{q}\|^2I_3 - J_\omega\dot{q}\dot{q}^TJ_\omega^T){}^0_ir_1, \qquad 1\le j\le n \quad (59)$$

and $Y_2 \in R^{n\times 3}$ is given by (42) with $B(d)$ given by (54).

For ${}^ip_{c_i} = [0\,0\,p_z]^T$, a simplified formula can also be established that is almost identical to (52)–(59) except that ${}^0_ir_1$ there should be replaced by ${}^0_ir_3$—the third column of ${}^0_iR$.

C. Example

Consider the 2-DOF planar arm shown in Fig. 2, where the length of link i is denoted by l_i. It is assumed that the links are of point-mass type, with each mass center at the origin of its link frame. The robot handles a rectangular bar with uniform material density. The parameters of the load include mass m_3, mass center ${}^3p_{c3}$, and inertia tensor ${}^cI_{c_3}$, where

$${}^3p_{c_3} = [p_x \quad 0 \quad 0] \quad \text{and} \quad {}^cI_3 = \mathrm{diag}(I_{xx}, I_{yy}, I_{zz}).$$

For comparison, the manipulator regressor will be evaluated using the conventional approach described in Section I and the approach proposed in Section III.

$$\theta^{(i)} = \begin{cases} [m \quad mp_x \quad mp_x^2 \quad I_{xx} \quad I_{yy} \quad I_{zz}]^T, & \text{if the mass center on } x \text{ axis} \\ [m \quad mp_z \quad mp_z^2 \quad I_{xx} \quad I_{yy} \quad I_{zz}]^T, & \text{if the mass center on } z \text{ axis} \end{cases} \quad (51)$$

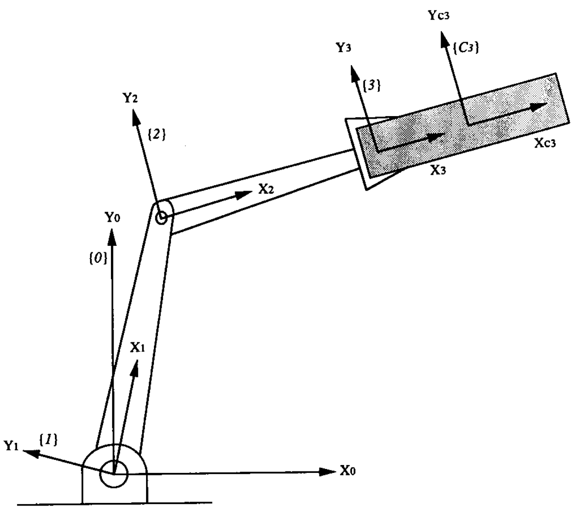


Fig. 2. Two-DOF planar robot handling a nonpoint-massload.

Conventional Approach: As mentioned earlier, there are two steps that need to be carried out in this indirect approach.

Step 1—Establishing the Robot Dynamics: Following [14], we compute

$$H(q) = \sum_{i=1}^{3} [m_i (\hat{J}_v^{(i)})^T \hat{J}_v^{(i)} + (J_\omega^{(i)})^T\, {}^cI_i J_\omega^{(i)}],$$

where $\hat{J}_v^{(i)}$ and $\hat{J}_\omega^{(i)}$ form the link Jacobian for link i that relates the joint velocity to the velocity of the frame $\{c_i\}$ which is obtained by translating$\{i\}$ to the mass center of link i. In our case, $\{c_1\} = \{1\}, \{c_2\} = \{2\}$, and ${}^cI_1 = {}^cI_2 = 0$, so $\hat{J}_v^{(1)} = J_v^{(1)}, \hat{J}_v^{(2)} = J_v^{(2)}$, where

$$J_v^{(1)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad J_v^{(2)} = \begin{bmatrix} -l_1 s_1 & 0 \\ l_1 c_1 & 0 \\ 0 & 0 \end{bmatrix}. \tag{60}$$

Since the linear velocity of $\{c_3\}$ is

$$v_{c_3} = [-l_1 s_1 \dot{\theta}_1 - (l_2 + p_x) s_{12}(\dot{\theta}_1 + \dot{\theta}_2) \cdot l_1 c_1 \dot{\theta}_1 + (l_2 + p_x) c_{12}(\dot{\theta}_1 + \dot{\theta}_2)],$$

we obtain

$$\hat{J}_v^{(3)} = \begin{bmatrix} -l_1 s_1 - (l_2 + p_x) s_{12} & -(l_2 + p_x) s_{12} \\ l_1 c_1 + (l_2 + p_x) c_{12} & (l_2 + p_x) c_{12} \end{bmatrix}.$$

Furthermore, notice that

$$J_\omega^{(3)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}$$

hence the equation at the bottom of the page results. Denoting $H(q) = (H_{ij})$, it follows from [14] that the term $C\dot{q}$ in (1) is

$$C(q, \dot{q})\dot{q} = \begin{bmatrix} \sum_{j=1}^{2} \left(\frac{\partial H_{1j}}{\partial q_1} - \frac{1}{2} \frac{\partial H_{j1}}{\partial q_1} \right) \dot{q}_j & \sum_{j=1}^{2} \left(\frac{\partial H_{1j}}{\partial q_2} - \frac{1}{2} \frac{\partial H_{j2}}{\partial q_1} \right) \dot{q}_j \\ \sum_{j=1}^{2} \left(\frac{\partial H_{2j}}{\partial q_1} - \frac{1}{2} \frac{\partial H_{j1}}{\partial q_2} \right) \dot{q}_j & \sum_{j=1}^{n} \left(\frac{\partial H_{2j}}{\partial q_2} - \frac{1}{2} \frac{\partial H_{j2}}{\partial q_2} \right) \dot{q}_j \end{bmatrix} \dot{q}$$
$$= \begin{bmatrix} -l_1(l_2 + p_x) s_2 m_3 \dot{q}_2 (2\dot{q}_1 + \dot{q}_2) \\ l_1 (l_2 + p_x) s_2 m_3 \dot{q}_1^2 \end{bmatrix}.$$

Finally, we compute the total potential energy of the system as

$$U = m_2 g l_1 s_1 + m_3 g [l_1 s_1 + (l_2 + p_x) s_{12}]$$

hence

$$G = \begin{bmatrix} \frac{\partial U}{\partial q_1} \\ \frac{\partial U}{\partial q_2} \end{bmatrix} = \begin{bmatrix} (m_2 + m_3) l_1 g c_1 + m_3 g (l_2 + p_x) c_{12} \\ m_3 g (l_2 + p_x) c_{12} \end{bmatrix}$$

Step 2—Parameter Extraction: As mentioned in Section I, this step begins with defining a parameter vector θ. Suppose that the parameters of interest are link mass m_2 (note that the link mass m_1 plays no role in the dynamics), and the parameters associated with the load, that is, m_3, p_x, and I_{zz}. From what we have done in Step 1, it is observed that every entry of H, C, and G involves some of these parameters. In addition, in a number of entries, parameters present themselves in a nonlinear manner such as $m_3 p_x$. It takes a while to figure out that

$$\theta = [m_2 \quad m_3 \quad m_3 p_x \quad m_3 p_x^2 \quad I_{zz}]^T$$

defines a parameter vector with minimum dimension such that for every entry of H, C, and G, the parameters of interest can all be extracted. Note that one does not need to include p_x as a single component in θ, as parameter p_x in all the terms involved always presents itself together with m_3. Obviously, such a parameter vector cannot be defined adequately before a careful inspection of all entries is completed. For a robot with more non-point-mass joints, this step will become quite involved and time-consuming.

With θ as defined earlier, we can now extract it from the entries of H, C, and G as

$$H_{11} = [l_1^2 \quad l_1^2 + 2 l_1 l_2 c_2 + l_2^2 \quad 2(l_1 c_2 + l_2) \quad 1 \quad 1]\theta \equiv h_{11}\theta$$
$$H_{12} = [0 \quad l_2^2 + l_1 l_2 c_2 \quad l_1 c_2 + 2 l_2 \quad 1 \quad 1]\theta \equiv h_{12}\theta$$
$$H_{22} = [0 \quad l_2^2 \quad 2 l_2 \quad 1 \quad 1]\theta \equiv h_{22}\theta$$
$$C(q, \dot{q})\dot{q} = \begin{bmatrix} 0 & -l_1 l_2 s_2 \dot{q}_2 (2\dot{q}_1 + \dot{q}_2) & -l_1 s_2 \dot{q}_2 (2\dot{q}_1 + \dot{q}_2) & 0 & 0 \\ 0 & l_1 l_2 s_2 \dot{q}_1^2 & l_1 s_2 \dot{q}_1^2 & 0 & 0 \end{bmatrix} \cdot \theta \equiv Y_c \theta$$
$$G = \begin{bmatrix} l_1 c_1 g & (l_1 c_1 + l_2 c_{12}) g & c_{12} g & 0 & 0 \\ 0 & l_2 c_{12} g & c_{12} g & 0 & 0 \end{bmatrix} \theta \equiv Y_G \theta.$$

Therefore, the manipulator regressor associated with θ is given by (61), at the bottom of the next page.

$$H(q) = m_2 (J_v^{(2)})^T J_v^{(2)} + m_3 (\hat{J}_v^{(3)})^T \hat{J}_v^{(3)} + (J_\omega^{(3)})^T\, {}^cI_3 J_\omega^{(3)}$$
$$= \begin{bmatrix} l_1^2 m_2 + [l_1^2 + (l_2 + p_x)^2 + 2 l_1 (l_2 + p_x) c_2] m_3 + I_{zz} & [(l_2 + p_x)^2 + l_1 (l_2 + p_x) c_2] m_3 + I_{zz} \\ [(l_2 + p_x)^2 + l_1 (l_2 + p_x) c_2] m_3 + I_{zz} & (l_2 + p_x)^2 m_3 + I_{zz} \end{bmatrix}$$

Proposed Approach: The problem of computing $Y^{(3)}(q,\dot q,\ddot q)$ falls obviously within the special case addressed in Section III-B-2. Using

$$J_v(q)=\begin{bmatrix}-l_1s_1-l_2s_{12} & -l_2s_{12}\\ l_1c_1+l_2c_{12} & l_2c_{12}\\ 0 & 0\end{bmatrix}\quad\text{and}$$

$$J_\omega(q)=\begin{bmatrix}0&0\\0&0\\1&1\end{bmatrix},$$

(52) becomes the equation shown at the bottom of next the page. Since $B(d)=\mathrm{diag}(0,0,\dot q_1+\dot q_2)$, (29) gives

$$W_2(q,\dot q)=\begin{bmatrix}0&0&\dot q_1+\dot q_2\\0&0&\dot q_1+\dot q_2\end{bmatrix}.$$

By (34), (45), and (56)–(59), we obtain $\hat Y_1$ with

$$Y_{11}+Y_{31}=\begin{bmatrix}(l_1c_1+l_2c_{12})g\\ l_1l_2s_2\dot q_1(\dot q_1+\dot q_2)+l_2c_{12}g\end{bmatrix}$$

$$y_{121}+y_{321}=\begin{bmatrix}c_{12}g\\ l_1s_2\dot q_1(\dot q_1+\dot q_2)+c_{12}g\end{bmatrix}$$

$$y_{141}=\begin{bmatrix}0\\0\end{bmatrix}.$$

Furthermore, by (42) it is found that $Y_2(q,\dot q)$ is a 2×3 zero matrix.

Since Y_2 and the first two columns of W_2 are all zero, $\theta^{(3)}$ can be redefined as

$$\theta^{(3)}=[m_3\quad m_3p_x\quad m_3p_x^2\quad I_{zz}]^T \tag{62}$$

and $Y^{(3)}$ is now obtained as (63), shown at the bottom of the page. To compute $Y^{(1)}$ and $Y^{(2)}$, note that each link is of point-mass involving only one parameter—its link mass. The link Jacobians for link 1 and 2 are given by (60), and (49) gives

$$Y^{(1)}=\begin{bmatrix}0\\0\end{bmatrix},\qquad Y^{(2)}=\begin{bmatrix}l_1^2\ddot q_1+l_1c_1g\\0\end{bmatrix}. \tag{64}$$

Since the link mass m_1 does not play a role in the dynamics, Y_1 (the zero column) can be dropped in Y, that is

$$Y=[Y^{(2)}\quad Y^{(3)}] \tag{65}$$

with $\theta=[m_2\,m_3\,m_3p_x\,m_3p_x^2\,I_{zz}]^T$.

On comparing these two approaches and their results, namely (63)–(65) and (61), it is observed that the proposed approach finds the regressor by directly and, therefore, more efficiently manipulating the link Jacobians. As our method is evolved from Lagrange's equation of motion, it may be viewed as a counterpart of the Lagrangian formulation for the evaluation of regressor dynamics.

IV. Iterative Computation of $Y(q,\dot q,\ddot q)$

We begin our analysis by reformulating the Newton–Euler recursion [9] using the techniques developed in Appendix I. For the sake of simplicity, all joints are assumed to be revolute. Notation adopted below will be consistent with those used in [6, chap. 6].

A. Outward Iterations (i: $0\to n$)

By (104), the outward iterations [6, p. 200] imply that

$${}^{i+1}F_{i+1}=A_{i+1}\theta_2^{(i+1)} \tag{66}$$

where $\theta_2^{(i+1)}$ is defined by θ_2 in (101) with the understanding that m_{i+1} is the mass of link $i+1$, ${}^{i+1}p_{c_{i+1}}=[p_x\,p_y\,p_z]^T$, and

$$A_{i+1}=[{}^{i+1}\dot v_{i+1}\quad H_{i+1}] \tag{67}$$

$$H_{i+1}=\Omega({}^{i+1}\dot\omega_{i+1})+{}^{i+1}U_{i+1}-\|{}^{i+1}\omega_{i+1}\|^2I \tag{68}$$

$${}^{i+1}U_{i+1}={}^{i+1}\omega_{i+1}\,{}^{i+1}\omega_{i+1}^T.$$

By (107), we obtain

$${}^{i+1}N_{i+1}=E_{i+1}\theta_3^{(i+1)} \tag{69}$$

$$\begin{aligned}Y&=\begin{bmatrix}h_{11}\ddot q_1+h_{12}\ddot q_2\\h_{12}\ddot q_1+h_{11}\ddot q_2\end{bmatrix}+Y_c+Y_G\\&=\begin{bmatrix}l_1(\ddot q_1+c_1g) & \begin{gathered}(l_1^2+2l_1l_2+l_2^2)\ddot q_1\\+l_2(l_1c_2+l_2)\ddot q_2\\-l_1l_2s_2\dot q_2(2\dot q_1+q_2)\\+(l_1c_1+l_2c_{12})g\end{gathered} & \begin{gathered}2(l_1c_2+l_2)\ddot q_1\\+(l_1c_2+2l_2)\ddot q_2\\-l_1s_2\dot q_2(2\dot q_1+\dot q_2)\\+c_{12}g\end{gathered} & \ddot q_1+\ddot q_2 & \ddot q_1+\ddot q_2\\[2ex] 0 & \begin{gathered}l_2(l_1c_2+l_2)\ddot q_1+l_2^2\ddot q_2\\+l_1l_2s_2\dot q_1^2+l_2c_{12}g\end{gathered} & \begin{gathered}(l_1c_2+2l_2)\ddot q_1+2l_2^2\ddot q_2\\+l_1s_2\dot q_1^2+c_{12}g\end{gathered} & \ddot q_1+\ddot q_2 & \ddot q_1+\ddot q_2\end{bmatrix}\end{aligned} \tag{61}$$

$$W_1(q,\dot q)=\begin{bmatrix}(l_1^2+l_2^2+2l_1l_2c_2)\dot q_1+(l_2^2+l_1l_2c_2)\dot q_2 & 2(l_1c_2+l_2)\dot q_1+(l_1c_2+2l_2)\dot q_2 & \dot q_1+\dot q_2\\ (l_2^2+l_1l_2c_2)\dot q_1+l_2^2\dot q_2 & (l_1c_2+2l_2)\dot q_1+2l_2\dot q_2 & \dot q_1+\dot q_2\end{bmatrix}$$

$$Y^{(3)}=\begin{bmatrix}\begin{gathered}(l_1^2+2l_1l_2+l_2^2)\ddot q_1\\+l_2(l_1c_2+l_2)\ddot q_2\\-l_1l_2s_2\dot q_2(2\dot q_1+q_2)\\+(l_1c_1+l_2c_{12})g\end{gathered} & \begin{gathered}2(l_1c_2+l_2)\ddot q_1\\+(l_1c_2+2l_2)\ddot q_2\\-l_1s_2\dot q_2(2\dot q_1+\dot q_2)\\+c_{12}g\end{gathered} & \ddot q_1+\ddot q_2 & \ddot q_1+\ddot q_2\\ \begin{gathered}l_2(l_1c_2+l_2)\ddot q_1+l_2^2\ddot q_2\\+l_1l_2s_2\dot q_1^2+l_2c_{12}g\end{gathered} & \begin{gathered}(l_1c_2+2l_2)\ddot q_1+2l_2^2\ddot q_2\\+l_1s_2\dot q_1^2+c_{12}g\end{gathered} & \ddot q_1+\ddot q_2 & \ddot q_1+\ddot q_2\end{bmatrix} \tag{63}$$

where $\theta_3^{(i+1)}$ is defined by θ_3 in (106) with $I_{xx}, \cdots, I_{yz}$ from ${}^cI_{i+1}$, and

$$E_{i+1} = B({}^{i+1}\dot{\omega}_{i+1}) + \Omega({}^{i+1}\omega_{i+1})B({}^{i+1}\omega_{i+1}). \quad (70)$$

B. Inward Iterations (i: $n+1 \rightarrow 1$)

If no contact occurs between the load and its environment, we canbegin the inward iteration with zero boundary conditions. By (66) and (69), the inward iterations [6, p. 200] imply that

$$\begin{aligned} {}^{n+1}f_{n+1} &= {}^{n+1}F_{n+1} = A_{n+1}\theta_2^{(n+1)} \\ {}^{n+1}n_{n+1} &= E_{n+1}\theta_3^{(n+1)} + {}^{n+1}p_{c_{n+1}} \times A_{n+1}\theta_2^{(n+1)} \end{aligned} \quad (71)$$

where

$$\begin{aligned} {}^{n+1}P_{c_{n+1}} \times A_{n+1}\theta_2^{(n+1)} &= m_{n+1}\,{}^{n+1}p_{c_{n+1}} \\ &\quad \times ({}^{n+1}\dot{v}_{n+1} + H_{n+1}\,{}^{n+1}p_{c_{n+1}}) \end{aligned}$$

with m_{n+1} the mass of link $n+1$ (i.e., the load) and H_{n+1} is defined in (68). From (109)–(116) it follows that

$${}^{n+1}p_{c_{n+1}} \times A_{n+1}\theta_2^{(n+1)} = \Phi_{n+1}\begin{bmatrix}\theta_1^{(n+1)} \\ \theta_4^{(n+1)}\end{bmatrix} \quad (72)$$

where $\theta_1^{(n+1)}$ and $\theta_4^{(n+1)}$ are defined by (98) and (112), respectively, with

$$m = m_{n+1}, \qquad [p_x p_y p_z]^T = {}^{n+1}p_{c_{n+1}},$$

and

$$\Phi_{n+1} = [-\Omega({}^{n+1}\dot{v}_{n+1}) \quad \Psi_{n+1}] \quad (73)$$

with Ψ_{n+1} defined by (114). Hence,

$${}^{n+1}n_{n+1} = \Pi_{n+1}\theta^{(n+1)} \quad (74)$$

where

$$\Pi_{n+1} = [\phi \quad \Phi_{n+1} \quad E_{n+1}], \qquad \phi = [0 \quad 0 \quad 0]^T. \quad (75)$$

To obtain the iterative relation of Π_{i+1} to Π_i for $i = n, n-1, \cdots, 1$, first we write the vector θ in (12) as

$$\begin{aligned}\theta = [&\theta_2^{(1)T} \quad \theta_4^{(1)T} \quad \theta_3^{(1)T} \quad \theta_2^{(2)T} \quad \theta_4^{(2)T} \quad \theta_3^{(2)T} \cdots \theta_2^{(n+1)T} \\ &\theta_4^{(n+1)T} \quad \theta_3^{(n+1)T}]^T\end{aligned} \quad (76)$$

and write (71) and (74) as

$${}^{n+1}f_{n+1} = \tilde{A}_{n+1}\theta \quad (77)$$

with

$$\tilde{A}_{n+1} = [0 \quad 0 \quad \cdots \quad 0 \quad A_{n+1} \quad 0] \quad (78)$$

and

$${}^{n+1}n_{n+1} = \tilde{\Pi}_{n+1}\theta \quad (79)$$

with

$$\tilde{\Pi}_{n+1} = [0 \quad 0 \quad \cdots \quad 0 \quad \Pi_{n+1}] \quad (80)$$

where A_{n+1} is placed in a position consistent with that of $\theta_2^{(n+1)}$ in (76), and Π_{n+1} is placed in a position consistent with that of $\theta^{(n+1)}$. By (77) and (66), the inward iterations [6, p. 200] for $1 \leq i \leq n$ give

$$\begin{aligned}{}^if_i &= {}^i_{i+1}R\tilde{A}_{i+1}\theta + A_i\theta_2^{(i)} \\ &\tilde{A}_i\theta\end{aligned} \quad (81)$$

where

$$\tilde{A}_i = {}^i_{i+1}R\tilde{A}_{i+1} + [0 \quad \cdots \quad 0 \quad A_i \quad 0 \quad \cdots \quad 0] \quad (82)$$

with A_i placed in a position consistent with that of $\theta_2^{(i)}$ in (76). Furthermore, by (69), (79), (72), (102), and (81), the inward iterations give

$${}^in_i = \tilde{\Pi}_i\theta \quad (83)$$

where

$$\begin{aligned}\tilde{\Pi}_i = &{}^i_{i+1}R\tilde{\Pi}_{i+1} + \Omega({}^ip_{i+1}){}^i_{i+1}R\tilde{A}_{i+1} \\ &+ [0 \quad \cdots \quad 0 \quad \Phi_i \quad E_i \quad 0 \quad \cdots \quad 0]\end{aligned} \quad (84)$$

with Φ_i placed in a position consistent with that of $\theta_1^{(i)}$ (which is the last three components of $\theta_2^{(i)}$) and $\theta_4^{(i)}$, and E_i placed in a position consistent with that of $\theta_3^{(i)}$, respectively.

We now obtain the manipulator dynamics as $\tau = Y(q, \dot{q}, \dot{q})\theta$, where

$$Y(q, \dot{q}, \dot{q}) = \begin{bmatrix}\hat{z}\tilde{\Pi}_1 \\ \vdots \\ \hat{z}\tilde{\Pi}_n\end{bmatrix} \quad (85)$$

with $\hat{z} = [0\,0\,1]$ and $\tilde{\Pi}_i$ given by (84) for $1 \leq i \leq n$. Obviously, if joint i is not revolute but prismatic, then the ith row of Y in (85) should be $\hat{z}\tilde{A}_i$. In summary, the computation of $Y(q, \dot{q}, \dot{q})$ in (85) can be accomplished by following the steps listed below.

Algorithm for Computation of $Y(q, \dot{q}, \dot{q})$

Step 1: Compute A_{n+1} using (67) and (68).
Step 2: Compute E_{n+1} using (69) and (70).
Step 3: Compute Φ_{n+1} using (73).
Step 4: Form $\tilde{A}_{n+1}$ and $\tilde{\Pi}_{n+1}$ using (78) and (75), (80), respectively.
Step 5: For $i = n, n-1, \cdots, 1$.

- Compute A_i using (67) and (68).
- Compute E_i using (69) and (70).
- Compute Φ_i using (73).
- Compute $\tilde{A}_i$ using (82).
- Compute $\tilde{\Pi}_i$ using (84).

Step 6: Form $Y(q, \dot{q}, \dot{q})$ where

$$\text{The } i\text{th row of } Y(q, \dot{q}, \dot{q}) = \begin{cases}\hat{z}\tilde{\Pi}_i, & \text{if joint } i \text{ is revolute} \\ \hat{z}\tilde{A}_i, & \text{if joint } i \text{ is prismatic.}\end{cases}$$

C. Example

We consider the robot used in Section III and perform the iterative algorithm to compute its regressor as follows.

Following the steps given in Section IV-B, we compute

$$\begin{aligned}A_3 &= [{}^3\dot{v}_3 \quad \Omega({}^3\dot{\omega}_3) + {}^3U_3 - \|{}^3\omega_3\|^2 I] \\ &= \begin{bmatrix}\beta_1 & -(\dot{q}_1 + \dot{q}_2)^2 & -(\dot{q}_1 + \dot{q}_2) & 0 \\ \beta_2 & \dot{q}_1 + \dot{q}_2 & -(\dot{q}_1 + \dot{q}_2)^2 & 0 \\ 0 & 0 & 0 & 0\end{bmatrix}\end{aligned}$$

where

$$\begin{aligned}\beta_1 &= l_1 s_2\dot{q}_1 - l_1 c_2\dot{q}_1^2 + s_{12}g - l_2(\dot{q}_1 + \dot{q}_2)^2 \\ \beta_2 &= l_1 c_2\dot{q}_1 + l_1 s_2\dot{q}_1^2 + c_{12}g + l_2(\dot{q}_1 + \dot{q}_2).\end{aligned}$$

Because of the regular geometry of the load, we have $p_y = p_z = 0$, hence $\theta_2^{(3)}$ can be regarded as a two-dimensional vector

$$\theta_2^{(3)} = [m_3 \quad m_3 p_x]^T.$$

Consequently, the last two columns of matrix A_3 are not needed in the computation and A_3 can be redefined as

$$A_3 = \begin{bmatrix}\beta_1 & -(\dot{q}_1 + \dot{q}_2)^2 \\ \beta_2 & (\dot{q}_1 + \dot{q}_2) \\ 0 & 0\end{bmatrix}.$$

Similarly, we compute

$$E_3 = \begin{bmatrix} 0 \\ 0 \\ \dot{q}_1 + \dot{q}_2 \end{bmatrix}$$

with

$$\theta_3^{(3)} = I_{zz}.$$

Next note that

$$\begin{bmatrix} \theta_1^{(3)} \\ \theta_4^{(3)} \end{bmatrix} = \begin{bmatrix} m_3 p_x \\ m_3 p_x^2 \end{bmatrix}$$

hence, we only need the first and fourth columns of Φ_3. By (73) it is found that

$$\Phi_3 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \beta_2 & \dot{q}_1 + \dot{q}_2 \end{bmatrix}.$$

Using (78), (75), and (80), we have

$$\tilde{A}_3 = \begin{bmatrix} 0 & 0 & \beta_1 & -(\dot{q}_1+\dot{q}_2)^2 & 0 & 0 \\ 0 & 0 & \beta_2 & \dot{q}_1+\dot{q}_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

and

$$\tilde{\Pi}_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta_2 & \dot{q}_1+\dot{q}_2 & \dot{q}_1+\dot{q}_2 \end{bmatrix}.$$

Furthermore, using (84), it is found that

$$\tilde{\Pi}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -(\dot{q}_1+\dot{q}_2) & 0 & 0 \\ 0 & 0 & l_2\beta_2 & l_2(\dot{q}_1+\dot{q}_2) & \dot{q}_1+\dot{q}_2 & \dot{q}_1+\dot{q}_2 \end{bmatrix}$$

and

$$\tilde{\Pi}_1 = \begin{bmatrix} 0 & 0 & 0 & * & 0 & 0 \\ 0 & 0 & 0 & * & 0 & 0 \\ 0 & \pi_{32} & \pi_{33} & \pi_{34} & \dot{q}_1+\dot{q}_2 & \dot{q}_1+\dot{q}_2 \end{bmatrix}$$

where

$$\pi_{32} = l_1^2\dot{q}_1 + l_1 c_1 g$$
$$\pi_{33} = l_1(s_2\beta_1 + c_2\beta_2) + l_2\beta_2$$
$$\pi_{34} = (l_2 + l_1 c_2)(\dot{q}_1+\dot{q}_2) - l_1 s_2(\dot{q}_1+\dot{q}_2)^2 + \beta_2.$$

The parameter vector associated with $\tilde{\Pi}_i$ is

$$\theta = [m_2 \quad m_3 \quad m_3 p_x \quad m_3 p_x^2 \quad I_{zz}].$$

Using (85), the corresponding regressor $Y(q, \dot{q}, \dot{q})$ is given by

$$Y = \begin{bmatrix} \pi_{32} & \pi_{33} & \pi_{34} & \dot{q}_1+\dot{q}_2 & \dot{q}_1+\dot{q}_2 \\ 0 & l_2\beta_2 & l_2(\dot{q}_1+\dot{q}_2)+\beta_2 & \dot{q}_1+\dot{q}_2 & \dot{q}_1+\dot{q}_2 \end{bmatrix}.$$

Note that the Y obtained is identical to the one given by (65).

V. Stable Adaptive Motion Control and Its Implementation Using the Regressor

In the first part of this section, a modified version of the Slotine–Li adaptive scheme [5] is presented, and its global stability is shown through a Lyapunov approach. Unlike the algorithm in [5], the modified version enables us to use directly the regressor Y in controller implementation. Another adaptive motion control algorithm that uses Y directly was proposed by Craig *et al.* [10]. Contrary to the algorithm of [10], however, the approach proposed below does not require the use of joint acceleration measurements $\ddot{q}$ and inversion of the estimated mass matrix. In the second part of this section, we present a case study that simulates the two adaptive motion control algorithms where the regressors encountered are evaluated using the iterative algorithm described in Section IV.

A. Modified Slotine–Li Adaptive Control Algorithm

Following the discussion in Section II-A, we define

$$\dot{q}_r = \dot{q}_d - \Lambda\tilde{q} \tag{86}$$

where $\dot{q}_d$ is the desired velocity, $\tilde{q} = q - q_d$, and $\Lambda > 0$, and note that

$$\tilde{H}\dot{q}_r + \tilde{C}\dot{q} + \tilde{G} = Y_u(q, \dot{q}, \dot{q}_r)\tilde{\theta}_u. \tag{87}$$

If a control torque vector τ is assigned as

$$\tau = \hat{H}\dot{q}_r + \hat{C}\dot{q} + \hat{G} - Ks \tag{88}$$

with $s = \dot{q} - \dot{q}_r$ and if $\hat{\theta}_u$ is updated according to

$$\dot{\hat{\theta}}_u = -\Gamma Y_u^T(q, \dot{q}, \dot{q}_r)s, \qquad \Gamma > 0 \tag{89}$$

then for any $s(0), \hat{\theta}_u(0)$, and a bounded $q_d(t)$, there exists a $K > 0$ such that the position and velocity tracking errors converge to zero. To show this, consider the Lyapunov function

$$v = \frac{1}{2}[s^T H s + \tilde{\theta}_u^T \Gamma^{-1}\tilde{\theta}_u] \tag{90}$$

and compute its time derivative along trajectories of (1) as

$$\begin{aligned} \dot{v} &= s^T(H\ddot{q} - H\ddot{q}_r) + \tilde{\theta}_u^T\Gamma^{-1}\dot{\hat{\theta}}_u + \frac{1}{2}s^T\dot{H}s \\ &= s^T(\tau - C\dot{q}_r - G - H\ddot{q}_r) + \tilde{\theta}_u^T\Gamma^{-1}\dot{\hat{\theta}}_u \end{aligned}$$

where the fact that $\dot{H} - 2C$ is skew symmetric has been used. If control (88) and parameter update law (89) are employed, then

$$\begin{aligned} \dot{v} &= -s^T[K - C(q, \dot{q})]s \\ &= -s^T[K - \bar{C}(q, \dot{q})]s \end{aligned} \tag{91}$$

where $\bar{C}(q, \dot{q}) = [C(q, \dot{q}) + C^T(q, \dot{q})]/2$ is a symmetric matrix. By (91) in conjunction with the same argument as was adopted in [5] and [11], it can be shown that both $\tilde{q}$ and $\dot{\tilde{q}}$ converge to zero.

Concerning the algorithm implementation, note that generating a new control torque vector requires that $Y_u(q, \dot{q}, \dot{q}_r)$ be evaluated to update parameter vector $\hat{\theta}_u$ in (89), and then $Y(q, \dot{q}, \dot{q}_r)$ be computed to obtain τ as (88) can be written in the form

$$\tau = Y(q, \dot{q}, \dot{q}_r)\begin{bmatrix} \theta_k \\ \hat{\theta}_u \end{bmatrix} - Ks. \tag{92}$$

As is noted in [5], if one chooses $\Lambda = \lambda I$ in (86) and $K = \lambda\hat{H}(q)$ in (88), then the time derivative of v in (90) along trajectories of (1) is given by

$$\dot{v} = -s^T[\lambda H(q) - \bar{C}(q, \dot{q})]s, \tag{93}$$

provided that the unknown parameters are updated according to

$$\dot{\hat{\theta}}_u = -\Gamma Y_u^T(q, \dot{q}, (\dot{q}_r - \lambda s))s. \tag{94}$$

Since $H(q)$ is uniformly positive definite, $\dot{v}$ in (93) is negative if λ is sufficiently large. Note that the control torque in this case becomes

$$\tau = Y(q, \dot{q}, \dot{q}^*)\begin{bmatrix} \theta_k \\ \hat{\theta}_u \end{bmatrix} \tag{95}$$

where

$$\dot{q}^* = \dot{q}_d - 2\lambda\dot{\tilde{q}} - \lambda^2\tilde{q}. \tag{96}$$

Obviously, (95) and (96) represent a quasi-computed-torque controller, which would be identical to the well-known computed-torque algorithm [6] if $\hat{\theta}_u = \theta_u$.

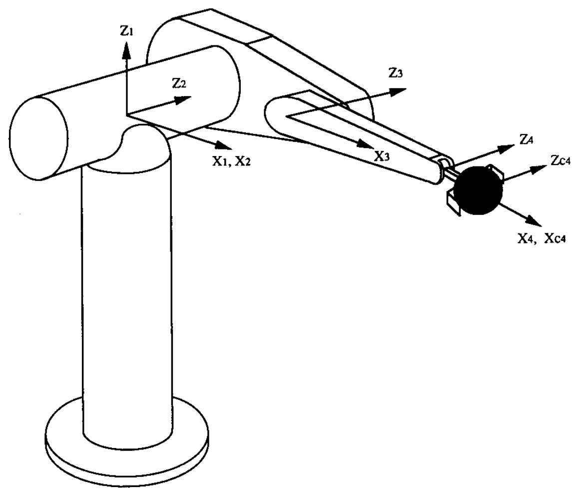


Fig. 3. Simplified PUMA-type 4-DOF robot handling an unknown point-mass load.

B. Case Study

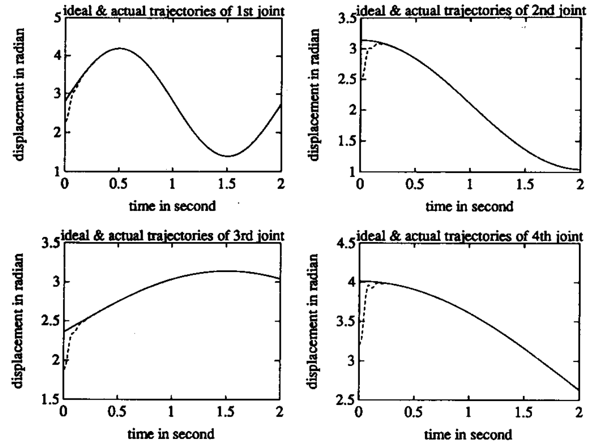
In what follows, we consider a 4-DOF manipulator, shown in Fig. 3,with a geometry identical to the first four links of a PUMA 500 robot. Its dynamic parameters are, however, simplified as follows: all four links are of point-mass with $m_1 = m_2 = m_3 = 2$ kg and $m_4 = 0.5$ kg. For the first three links, the mass center of each link is at its midpoint, and the mass center of link 4 is at its distal end. The robot carries a 0.5-kg point-mass load, tracking in the joint space a trajectory specified by

$$\begin{bmatrix} q_1(t) \\ q_2(t) \\ q_3(t) \\ q_4(t) \end{bmatrix} = \begin{bmatrix} c_1 + a_1 \sin \omega t \\ c_2 + a_2 \cos(\omega t/2) \\ c_3 + a_3 \sin(\omega t/3) \\ c_4 + a_4 \cos(\omega t/4) \end{bmatrix}$$

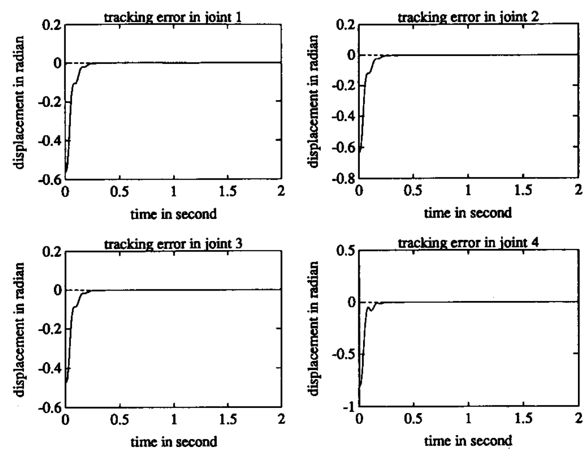
with $c_1 = 8\pi/9, c_2 = 2\pi/3, c_3 = 3\pi/4, c_4 = 15\pi/18, a_1 = 4\pi/9, a_2 = \pi/3, a_3 = \pi/4$, and $a_4 = 8\pi/18$ for $0 \leq t \leq 2$ s. Under these circumstances, the robot can be treated as if $m_4 = 1$ kg and carrying no load.

Now assume that the user does not know the load mass and makes an initial guess of $m_4 = 0.5$ kg. The initial robot configuration is set with 20% relative error in joint displacement for each joint. Both the modified Slotine–Li algorithm and the algorithm of Craig *et al.* [10] are applied to control the robot motion, and their tracking errors as well as parameter estimation errors are shown in Figs. 4 and 5. The regressors $Y(q, \dot{q}, \dot{q}^*)$ and $Y_u[q, \dot{q}, (\dot{q}_r - \lambda s)]$ in the modified Slotine–Li algorithm, and the regressor $Y_u(q, \dot{q}, \dot{q})$ in Craig's algorithm are evaluated using the iterative algorithm proposed in Section IV. From the figures, it is observed that it takes less than 0.3 s for the modified Slotine–Li algorithm and about 0.35 s for Craig's algorithm to see the tracking error converging to an acceptable tolerance. Both algorithms are able to identify the unknown mass of the load, but the modified Slotine–Li algorithm can do it quicker. As mentioned earlier, the control algorithm in Section V-A is computationally more efficient as compared with the algorithm of [10]. In addition, our simulation experience indicates that parameters λ and Γ in the modified Slotine–Li algorithm are less sensitive than their counterparts in Craig's algorithm, which is consistent with the observations made in a force control study [12].

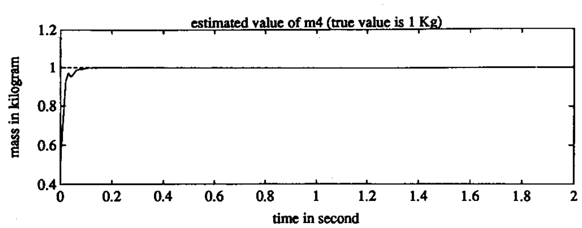
An issue that is often encountered in simulation of robot dynamics is the computation of $\ddot{q}$ given $q, \dot{q}$, and τ. The steps listed below form a regressor version of an approach suggested in [13] for computing $\ddot{q}$.



(a)



(b)



(c)

Fig. 4. Simulation of modified Slotine–Li algorithm. (a) Actual and ideal joint-space trajectories. (b) Tracking errors. (c) Estimated m_4.

Evaluation of $\ddot{q}$ in Simulations

Step 1: Compute $C\dot{q} + G = Y(q, \dot{q}, 0)\theta \equiv Y_0\theta$.

Step 2: Compute $H(q)e_i + C\dot{q} + G = Y(q, \dot{q}, e_i)\theta \equiv Y_i\theta$, for $1 \leq i \leq n$.

Step 3: Compute $H(q) = [(Y_1 - Y_0)\theta \cdots (Y_n - Y_0)\theta]$.

Step 4: $\ddot{q} = H^{-1}(q)(\tau - Y_0\theta)$.

where e_i is the ith column of the identity matrix. By replacing θ in the first three steps with $[\theta_k^T \ \hat{\theta}_u^T]^T, \hat{H}(q)$ in Craig's algorithm can

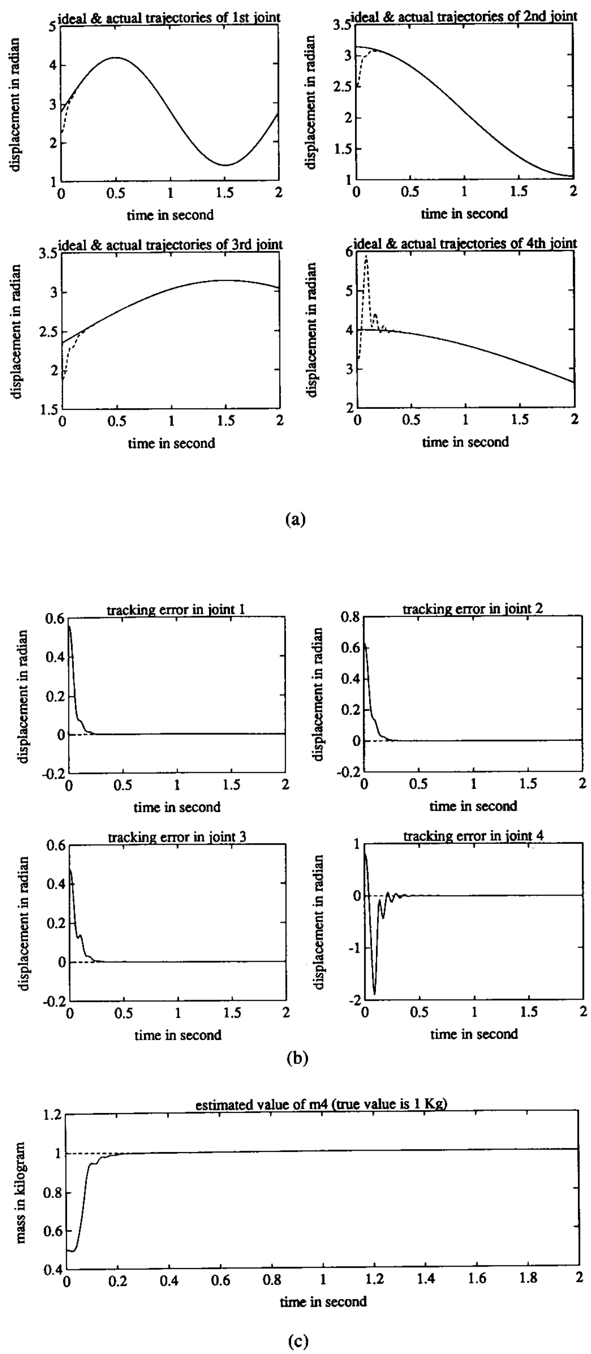


Fig. 5. Simulation of Craig's algorithm. (a) Actual and ideal joint-space trajectories. (b) Tracking errors. (c) Estimated m_4.

also be evaluated using the first three steps.

VI. Conclusion

An attempt has been made to derive a closed-form solution as well as iterative algorithm for symbolic and numerical evaluation of the manipulator regressor. Similar to the Lagrangian and Newton–Euler formulations of robot dynamics, the closed-form solution obtained offers physical insights of the regressor in terms of its relation to link Jacobians of the robot, whereas the proposed iterative algorithm provides an efficient approach to numerical evaluation of the regressor.

Appendix I
Parameter Extraction via Vector Analysis Techniques

Throughout this appendix we assume that $m \in R^{1\times 1}, p = [p_x\, p_y\, p_z]^T, d = [d_1\, d_2\, d_3]^T$, and $x = [x_1\, x_2\, x_3]^T$.

1) Consider

$$m(J_\omega \dot{q} \times Rp)^T g \tag{97}$$

where $J_\omega \in R^{3\times n}, R \in R^{3\times 3}$ and $g \in R^{3\times 1}$ are given. Denoting

$$\theta_1 = [mp_x \quad mp_y \quad mp_z]^T \tag{98}$$

and applying (19) to (97), we have

$$m(J_\omega \dot{q} \times Rp)^T g = \dot{q}^T Y_e \theta_1 \tag{99}$$

where

$$Y_e = [g \times J_{\omega_1} \quad g \times J_{\omega_2} \quad \cdots g \times J_{\omega_n}]^T R$$

is an $n \times 3$ known matrix.

2) Consider the vector given by

$$m[\dot{\omega} \times p + \omega \times (\omega \times p) + \dot{v}] \tag{100}$$

where m and p are parameters to be extracted. Define

$$\theta_2 = [m \quad mp_x \quad mp_y \quad mp_z] \tag{101}$$

and note that

$$\dot{\omega} \times p = \Omega(\dot{\omega})p, \tag{102}$$

where $\Omega(\dot{\omega})$ is a skew symmetric matrix characterizedby

$$\Omega(x) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}. \tag{103}$$

From (20) it follows that

$$\omega \times (\omega \times p) = (p^T \omega)\omega - (\omega^T \omega)p = [U - \|\omega\|^2 I]p,$$

where $U = \omega^T \omega$. Hence

$$m[\dot{\omega} \times p + \omega \times (\omega \times p) + \dot{v}] = [\dot{v}\, \Omega(\dot{\omega}) + U - \|\omega\|^2 I]\theta_2. \tag{104}$$

3) Let

$$\begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \tag{105}$$

be the unknown inertia matrix. Define

$$\theta_3 = [I_{xx} \quad I_{yy} \quad I_{zz} \quad I_{xy} \quad I_{xz} \quad I_{yz}]. \tag{106}$$

It can readily be verified that

$${}^cI_i d = B(d)\theta_3, \tag{107}$$

where

$$B(d) = \begin{bmatrix} d_1 & 0 & 0 & -d_2 & -d_3 & 0 \\ 0 & d_2 & 0 & -d_1 & 0 & -d_3 \\ 0 & 0 & d_3 & 0 & -d_1 & -d_2 \end{bmatrix}. \tag{108}$$

4) Consider the vector product

$$mp \times (\dot{v} + Hp) \tag{109}$$

where H is a 3×3 known matrix. By (102), the first vector product in (109) is

$$mp \times \dot{v} = -\Omega(\dot{v})\theta_2.$$

If we write

$$H = \begin{bmatrix} h_1^T \\ h_2^T \\ h_3^T \end{bmatrix}$$

and define

$$\tilde{\Psi} = \begin{bmatrix} 0 & h_3^T & -h_2^T \\ -h_3^T & 0 & h_1^T \\ h_2^T & -h_1^T & 0 \end{bmatrix}_{3\times 9} \tag{110}$$

then

$$mp \times Hp = \begin{bmatrix} h_3^T(mp_y p) - h_2^T(mp_z p) \\ h_1^T(mp_z p) - h_3^T(mp_x p) \\ h_2^T(mp_x p) - h_3^T(mpyp) \end{bmatrix} = \tilde{\Psi}\begin{bmatrix} mp_x p \\ mp_y p \\ mp_z p \end{bmatrix}_{9\times 1}. \tag{111}$$

By defining

$$\theta_4 = m[p_x^2 \quad p_y^2 \quad p_z^2 \quad p_x p_y \quad p_x p_z \quad p_y p_z]^T \tag{112}$$

(111) can be written as

$$mp \times Hp = \Psi\theta_4, \tag{113}$$

where

$$\Psi = \tilde{\Psi}[e_1 \quad e_5 \quad e_9 \quad e_2 + e_4 \quad e_3 + e_7 \quad e_6 + e_8] \tag{114}$$

and e_i is the ith column of the 9×9 identity matrix. Hence

$$mp \times (\dot{v} + Hp) = \Phi[\theta_1^T \quad \theta_4^T]^T \tag{115}$$

with

$$\Phi = [-\Omega(\dot{v}) \quad \Psi]. \tag{116}$$

5) Finally, let us consider

$$m\left[\frac{\partial}{\partial \dot{q}}(J_v\dot{q} + J_\omega\dot{q} \times {}^0_iRp)\right]^T (J_v\dot{q} + J_\omega\dot{q} \times {}^0_iRp). \tag{117}$$

By (102)

$$\frac{\partial}{\partial \dot{q}}(J_v\dot{q} + J_\omega\dot{q} \times {}^0_iRp) = J_v - \Omega({}^0_iRp)J_\omega. \tag{118}$$

Thus (117) can be written as

$$m[J_v^T J_v\dot{q} + J_v^T(J_\omega\dot{q} \times {}^0_iRp) - J_\omega^T\Omega^T({}^0_iRp)J_v\dot{q} - J_\omega^T\Omega^T({}^0_iRp)(J_\omega\dot{q} \times^0_i Rp)]. \tag{119}$$

By (19):

$$J_v^T(J_\omega\dot{q} \times {}^0_iRp) = (J_v \times J_\omega\dot{q})^T\, {}^0_iRp$$

the second term in (119) is equal to

$$(J_v \times J_\omega\dot{q})^T\, {}^0_iR(mp) \tag{120}$$

where

$$J_v \times J_\omega\dot{q} \equiv [J_{v_1} \times J_\omega\dot{q} \cdots J_{v_n} \times J_\omega\dot{q}]. \tag{121}$$

Since $\Omega(Rp)$ is skew symmetric, $-\Omega({}^0_iRp) = \Omega({}^0_iRp)$. By (102) and (19), the third term in (119) can be written as

$$-J_\omega^T(J_v\dot{q} \times {}^0_iRp) = (J_v\dot{q} \times J_\omega)^T\, {}^0_iRp \tag{122}$$

where

$$J_v\dot{q} \times J_\omega = [J_v\dot{q} \times J_{\omega_1} \cdots J_v\dot{q} \times J_{\omega_n}]. \tag{123}$$

Using (102) and (20), the last term in (119) can be expressed as

$$J_\omega^T\, {}^0_iRp \times (J_\omega\dot{q} \times {}^0_iRp) = \|p\|^2 J_\omega^T J_\omega\dot{q} - (p^T\, {}^0_iRJ_\omega\dot{q})J_\omega^T\, {}^0_iRp. \tag{124}$$

From (119), (120), (122), and (124), it follows that if we define a ten-dimensional vector θ_5 as

$$\theta_5 = m[1 \quad p_x \quad p_y \quad p_z \quad p_x^2 \quad p_y^2 \quad p_z^2 \quad pxpy \quad p_x \quad p_z \quad p_y p_z] = [\theta_2^T \quad \theta_4^T]^T \tag{125}$$

then (117) can be expressed as

$$m\left[\frac{\partial}{\partial \dot{q}}(J_v\dot{q} + J_\omega\dot{q} \times {}^0_iRp)\right]^T (J_v\dot{q} + J_\omega\dot{q} \times {}^0_iRp) = W_1(q, \dot{q})\theta_5 \tag{126}$$

with

$$W_1(q, \dot{q}) = [J_v^T J_v\dot{q} \quad (J_v \times J_\omega\dot{q} + J_v\dot{q} \times J_\omega)^T\, {}^0_iR \quad -D] \tag{127}$$

where

$$D = \tilde{D} - [J_\omega^T J_\omega\dot{q} \quad J_\omega^T J_\omega\dot{q} \quad J_\omega^T J_\omega\dot{q} \quad 0 \quad 0 \quad 0]_{n\times 6} \tag{128}$$

and $\tilde{D}$ is determined by

$$(p^T\, {}^0_iRJ_\omega\dot{q})J_\omega^T\, {}^0_iRp = \tilde{D}[p_x^2 \quad p_y^2 \quad p_z^2 \quad p_x p_y \quad p_x p_z \quad p_y p_z]^T. \tag{129}$$

References

[1] R. Ortega and M. W. Spong, "Adaptive motion control of rigid robots: A tutorial," in *Proc. 27th CDC,* 1988, pp. 1575–1584.

[2] C. H. An, C. G. Atkenson, and J. M. Hollerbach, "Estimation of inertial parameters of rigid body links of manipulators," in *Proc. 24th CDC,* 1985, pp. 990–995.

[3] P. Khosla and T. Kanade, "Parameter identification of robot dynamics," in *Proc. 24th CDC,* 1985, pp. 1754–1760.

[4] C. P. Neuman and P. K. Khosla, "Identification of robot dynamics: An application of recursive estimation," in *Proc. of 4th Yale Workshop on Appl. Adaptive Syst. Theory,* K. S. Narendra, Ed. May 29–31, 1985.

[5] J-J. E. Slotine and W. Li, "Adaptive manipulator control: A case study," *IEEE Trans. Automat. Contr.,* vol. AC-33, no. 11, pp. 995–1003, 1988.

[6] J. J. Craig, *Introduction to Robotics.* Reading, MA: Addison-Wesley, 1989.

[7] R. H. Middleton and G. C. Goodwin, "Adaptive computed torque control for rigid link manipulator," *Syst. Contr. Lett.,* vol. 10, pp. 9–16, 1988.

[8] J-J. E. Slotine and W. Li, "Composite adaptive control of robot manipulators," *Automatica,* vol. 25, no. 4, pp. 509–519, 1989.

[9] J. Y. S. Luh, M. W. Walker, and R. P. Paul, "On-line computational scheme for mechanical manipulators," *Trans. ASME, J. Dyn. Syst., Meas., Contr.,* vol. 102, pp. 69–72, June 1980.

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

[11] J-J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," *Int. J. Robotics Res.,* vol. 6, no. 3, 1987.

[12] W.-S. Lu and H. Meng, "Impedance control with adaptation for robotic manipulators," *IEEE Trans. Robotics Automat.,* vol. 7, no. 3, pp. 408–415, June 1991.

[13] M. W. Walker and D. E. Orin, "Efficient dynamic computer simulation of robotic mechanisms," *Trans. ASME, J. Dyn. Syst., Meas., Contr.,* vol. 104, pp. 205–211, Sept. 1982.

[14] H. Asada and J-J. E. Slotine, *Robot Analysis and Control.* New York: Wiley, 1986.