IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, May 9-10, 1991

# Recursive Computation of Manipulator Regressor and Its Application to Adaptive Motion Control of Robots

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

*Department of Electrical and Computer Engineering*
*University of Victoria*
*Victoria, B.C., V8W 3P6*

**Abstract**

This paper describes a recursive algorithm that generates the manipulator regressor for a generl $n$ d.o.f. robot. As the closed-form formula is derived from the Lagrangian equations of motion, the proposed recursive algorithm is obtained from the conventional Newton-Euler dynamic formulation by using a vector-analysis type technique.

## I. Introduction

Most of the established globally stable algorithms for adaptive motion control of robotic manipulators are derived and implemented using the manipulator regressor, often denoted as $Y(q,\dot{q},\ddot{q})$ or $W(q,\dot{q},\ddot{q})$ [1]. This is due to the fact that the use of manipulator regressor leads to a linear parametrization of the robot dynamics so that a Lyapunov approach may be applied, resulting in a certain *linear* law for updating the parameters. References addressing this linear parameter dependence issue from an identification point of view include [2]-[4]. In order to find $Y$, one may start with the conventional dynamic equations – the second-order nonlinear differential equations of the $n$-dimensional joint displacement vector $q$. By re-defining manipulator parameters and re-grouping various terms in the equations of motion, the regressor may eventually be found [1][5][6]. In [7], a closed-form formula for the regressor is developed for a general $n$ degree-of-freedom (d.o.f.) robot. This is accomplished based on the observation that the unknown part of the Lagrangian of a robot can be separated from the Lagrangian of the known part of the robot. While the formula is general and in a closed-form, its implementation requires extensive computation of various types of link-related Jacobians and, therefore, is numerically not feasible.

This paper describes a recursive algorithm that generates the manipulator regressor for a general $n$ d.o.f. robot. As the closed-form formula is derived from the Lagrangian equations of motion, the proposed recursive algorithm is obtained from the conventional Newton-Euler dynamic formulation [8] by using a vector-analysis type technique. In the next section, we show how the regressor $Y(q,\dot{q},\ddot{q})$ can be used to derive and implement a globally stable adaptive algorithm for unconstrained motion control. In Sec.III, we adopt vector-analysis type techniques to analyze the conventional Newton-Euler dynamic formulation. The analysis in turn leads to an efficient recursive formulation for computing the regressor. One example is included to illustrate the algorithm obtained.

## II. A Stable Adaptive Algorithm for Motion Control

Consider an $n$ d.o.f. rigid robot whose dynamics is described by

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau \tag{1}$$

where $H(q) \in R^{n\times n}$ is the positive definite mass matrix, $C(q,\dot{q})\dot{q}$ represents torques due to the centrifugal and Coriolis forces and is arranged such that $\dot{H} - 2C$ is skew symmetric [1], and $g(q)$ denotes the torque due to gravity. It has been shown that equation (1) can be written in the form of

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

where $Y(q,\dot{q},\ddot{q})$ is an $n \times r$ matrix of known functions, called the manipulator regressor, and $\theta$ is a $r$-dimensional parameter vector.
Denoting

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

with $\theta_k \in R^{r_1}$ the known parameter vector and $\theta_u \in R^{r_2}$ the unknown parameter vector, and partitioning $Y = [Y_k \ Y_u]$ with $Y_k \in R^{n\times r_1}$, $Y_u \in R^{n\times r_2}$, (1) and (2) imply that

$$H\ddot{q} + C\dot{q} + g = Y_k\theta_k + Y_u\theta_u \tag{3}$$

If $\theta_u$ are replaced by its estimate $\hat{\theta}_u$, then

$$\hat{H}\ddot{q} + \hat{C}\dot{q} + \hat{g} = Y_k\theta_k + Y_u\hat{\theta}_u$$

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

$$\tilde{H}\ddot{q} + \tilde{C}\dot{q} + \tilde{g} = Y_u(q,\dot{q},\ddot{q})\tilde{\theta}_u \tag{4}$$

where $\tilde{(*)} = \hat{(*)} - (*)$, and $\tilde{\theta}_u = \hat{\theta}_u - \theta_u$.

To justify the use of $Y_u$ in adaptive motion control, define $\dot{q}_r$ as

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

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

$$\tilde{H}\ddot{q}_r + \tilde{C}\dot{q} + \tilde{g} = Y_u(q,\dot{q},\ddot{q}_r)\tilde{\theta}_u$$

If the control torque $\tau$ is assigned as

$$\tau = \hat{H}\ddot{q}_r + \hat{C}\dot{q} + \hat{g} - Ks \tag{5}$$

with $s = \dot{q} - \dot{q}_r$, and $\hat{\theta}_u$ is updated by

$$\dot{\hat{\theta}}_u = -\Gamma Y_u^T(q,\dot{q},\ddot{q}_r)s, \qquad \Gamma > 0 \tag{6}$$

with sufficiently large gain $K > 0$, then the convergence of the positional and velocity tracking error is guaranteed provided that the desired $q_d$, $\dot{q}_d$ and $\ddot{q}_d$ are bounded. To show this, consider the Lyapunov function

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

and compute its time derivative along the trajectory of (1) as

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

where the fact that $\dot{H} - 2C$ is skew symmetric has been used. If control (5) and parameter update law (6) are employed, then

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

where $\bar{C}(q, \dot{q}) = [C(q, \dot{q}) + C^T(q, \dot{q})]/2$ is a symmetric matrix. Consequently if the gain matrix $K$ is sufficiently large such that $K - \bar{C}$ is positive definite, then $\dot{v} \leq 0$. For example, this will be the case if $K$ is chosen such that its smallest eigenvalue is greater than the 2-norm of $\bar{C}(q, \dot{q})$.

To update parameter vector $\theta_u$ through (6), $Y_u(q, \dot{q}, \ddot{q}_r)$ is needed at each control instant. Furthermore, by writing control torque $\tau$ as

$$\tau = Y_k(q, \dot{q}, \ddot{q}_r)\theta_k + Y_u(q, \dot{q}, \ddot{q}_r)\hat{\theta}_u - Ks \qquad (7)$$

we see that regressor $Y_u(q, \dot{q}, \ddot{q})$ is again needed in addition to the computation of $Y_k(q, \dot{q}, \ddot{q})\theta_k$.

## III. Recursive Computation of $Y_u(q, \dot{q}, \ddot{q})$

For the sake of simplicity, it is assumed in the rest of the paper that the $n$ d.o.f. manipulator is rigidly holding an object, of which the mass, the center of mass and the inertia tensor are the only unknown parameters. Let $\{n\}$ and $\{n+1\}$ be the two parallel frames attached to the gripper and the center of mass of the payload, respectively, and let $m$, ${}^n p_{n+1} = [p_x \ p_y \ p_z]^T = \mathbf{p}$, and

$${}^{Cn+1}I_{n+1} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix}$$

denote the mass, center of mass and the inertia tensor of the payload respectively. Other notation used in what follows is consistent with those adopted in text [6].

We begin our analysis by re-formulating the outward iteration of the conventional Newton-Euler recursion using vector-analysis type techniques.

### Outward Iterations (From base to payload)

Since all robot parameters are assumed to be known, we concentrate our attention on the end effector and the payload. By assumptions, there is no relative motion between $\{n\}$ and $\{n+1\}$, hence

$$\begin{aligned} {}^{n+1}\omega_{n+1} &= {}^n\omega_n \\ {}^{n+1}\dot{\omega}_{n+1} &= {}^n\dot{\omega}_n \\ {}^{n+1}\dot{v}_{n+1} &= {}^n\dot{\omega}_n \times {}^n p_{n+1} + {}^n\omega_n \times ({}^n\omega_n \times {}^n p_{n+1}) + {}^n\dot{v}_n \\ &= (\Omega({}^n\dot{\omega}_n) - \|{}^n\omega_n\|^2 I_3 + {}^nU_n){}^n p_{n+1} + {}^n\dot{v}_n \end{aligned}$$

where, for $x = [x_1 \ x_2 \ x_3]^T$, $\Omega(x)$ is defined by

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

and ${}^nU_n = {}^n\omega_n^n\omega_n^T$ i.e. the outer product of ${}^n\omega_n$ with itself. Furthermore, since the origin of $\{n+1\}$ is the center of mass of the payload,

$$\begin{aligned} {}^{n+1}\dot{v}_{C_{n+1}} &= {}^{n+1}\dot{v}_{n+1} \\ {}^{n+1}F_{n+1} &= m \, {}^{n+1}\dot{v}_{C_{n+1}} \\ &= [{}^n\dot{v}_n \quad \Omega({}^n\dot{\omega}_n) - \|{}^n\omega_n\|^2 I_3 + {}^n U_n]\theta_1 \\ &\equiv A_n\theta_1 \end{aligned}$$

with $\theta_1 = [m \ \ mp_x \ \ mp_y \ \ mp_z]^T$,

$$\begin{aligned} {}^{n+1}N_{n+1} &= {}^{Cn+1}I_{n+1} \, {}^n\dot{\omega}_n + {}^n\omega_n \times {}^{Cn+1}I_{n+1} \, {}^n\omega_n \\ &= B({}^n\dot{\omega}_n)\theta_2 + \Omega({}^n\omega_n)B({}^n\omega_n)\theta_2 \\ &\equiv E_n\theta_2 \end{aligned} \qquad (8)$$

where

$$\begin{aligned} B(x) &= \begin{bmatrix} x_1 & 0 & 0 & -x_2 & -x_3 & 0 \\ 0 & x_2 & 0 & -x_1 & 0 & -x_3 \\ 0 & 0 & x_3 & 0 & -x_1 & -x_2 \end{bmatrix} \\ \theta_2 &= [I_{xx} \ I_{yy} \ I_{zz} \ I_{xy} \ I_{xz} \ I_{yz}]^T \\ E_n &= B({}^n\dot{\omega}_n) + \Omega({}^n\omega_n)B({}^n\omega_n) \end{aligned}$$

### Inward Iterations (From payload to base)

Assume that no contact between the payload and the external environment, we have then zero boundary conditions to start the inward iterations, i.e.

$${}^{n+2}f_{n+2} = 0, \qquad {}^{n+1}n_{n+2} = 0$$

hence

$$\begin{aligned} {}^{n+1}f_{n+1} &= {}^{n+1}F_{n+1} = A_n\theta_1 \\ {}^{n+1}n_{n+1} &= {}^nN_{n+1} = E_n\theta_2 \\ {}^nf_n &= {}^n_{n+1}R^{n+1}f_{n+1} + {}^nF_n \\ &= {}^n_{n+1}RA_n\theta_1 + k.t. \end{aligned} \qquad (9)$$

where $k.t.$ denotes a "known term" meaning that it is independent of any unknown parameters, and

$$\begin{aligned} {}^nn_n &= {}^nN_n + {}^n_{n+1}R^{n+1}n_{n+1} + {}^np_{C_n} \times {}^nF_n \\ &\quad + {}^np_{n+1} \times {}^n_{n+1}R^{n+1}f_{n+1} \\ &= E_n\theta_2 + {}^np_{n+1} \times A_n\theta_1 + k.t. \end{aligned}$$

Note that

$$\begin{aligned} {}^np_{n+1} \times A_n\theta_1 &= m \, {}^np_{n+1} \times \{{}^n\dot{v}_n + [\Omega({}^n\dot{\omega}_n) \\ &\quad - \|{}^n\omega_n\|^2 I_3 + {}^nU_n]^n p_{n+1}\} \\ &= -\Omega({}^n\dot{v}_n)m^n p_{n+1} + \Psi({}^n\omega_n, {}^n\dot{\omega}_n)(m^n p_{n+1}) \\ &= [-\Omega({}^n\dot{v}_n) \quad \bar{\Psi}({}^n\omega_n, {}^n\dot{\omega}_n)]\theta_3 \\ &\equiv \Phi_n\theta_3 \end{aligned}$$

where

$$
\begin{aligned}
H({}^n\omega_n, {}^n\dot{\omega}_n) &= \Omega({}^n\dot{\omega}_n) - \|{}^n\omega_n\|^2 I_3 + {}^n U_n \equiv \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} \\
\Psi({}^n\omega_n, {}^n\dot{\omega}_n) &= \begin{bmatrix} 0 & h_3 & -h_2 \\ -h_3 & 0 & h_1 \\ h_2 & -h_1 & 0 \end{bmatrix}_{3\times 9} \\
\tilde{\Psi}({}^n\omega_n, {}^n\dot{\omega}_n) &= \Psi({}^n\omega_n, {}^n\dot{\omega}_n)[e_1 \;\; e_5 \;\; e_9 \;\; e_2 + e_4 \;\; e_3 + e_7 \;\; e_6 + e_8] \\
e_i &- \text{the } i\text{-th column of the } 9\times 9 \text{ identity matrix} \\
\Phi_n &= [-\Omega({}^n\dot{v}_n) \quad \tilde{\Psi}({}^n\omega_n, {}^n\dot{\omega}_n)] \qquad (10) \\
\theta_3 &= m[p_x \;\; p_y \;\; p_z \;\; p_x^2 \;\; p_y^2 \;\; p_z^2 \;\; p_x p_y \;\; p_x p_z \;\; p_y p_z]^T
\end{aligned}
$$

Thus

$$
\begin{aligned}
{}^n n_n &= [\Phi_n \quad E_n] \begin{bmatrix} \theta_3 \\ \theta_2 \end{bmatrix} + k.t. \\
&= [z_1 \quad \Phi_n \quad E_n] \begin{bmatrix} m \\ \theta_3 \\ \theta_2 \end{bmatrix} + k.t. \\
&\equiv \Pi_n \theta_u + k.t. \qquad (11) \\
\tau_n &= \hat{e}_1^T \Pi_n \theta_u + k.t. \qquad (12)
\end{aligned}
$$

where $z_1 = [0 \;\; 0 \;\; 0]^T$ and $\hat{e}_1 = [0 \;\; 0 \;\; 1]^T$.

Next we compute

$$
\begin{aligned}
{}^{n-1}f_{n-1} &= {}^{n-1}_{n}R\, {}^n f_n + {}^{n-1}_{n}R\, {}^n F_n + {}^{n-1}F_{n-1} \\
&= {}^{n-1}_{n+1}R A_n \theta_1 + {}^{n_1}_{n}R\, {}^n F_n + {}^{n-1}F_{n-1} \qquad (13) \\
{}^{n-1}n_{n-1} &= {}^{n-1}N_{n-1} + {}^{n-1}_{n}R^n n_n + {}^{n-1}p_{Cn-1} \times {}^{n-1}F_{n-1} \\
&\quad + {}^{n-1}p_n \times {}^{n-1}_{n}R\, {}^n f_n \\
&= {}^{n-1}_{n}R \Pi_n \theta_u + {}^{n-1}p_n \times {}^{n-1}_{n+1}R A_n \theta_1 + k.t. \\
&\equiv {}^{n-1}_{n}R \Pi_n \theta_u + A_{n-1}\theta_1 + k.t. \\
&\equiv \Pi_{n-1}\theta_u + k.t. \qquad (14) \\
A_{n-1} &= \Omega({}^{n-1}p_n)\, {}^{n-1}_{n+1}R A_n \qquad (15) \\
\Pi_{n-1} &= [(\text{the first 4 columns of } {}^{n-1}_{n}R\Pi_n) + A_{n-1} \\
&\quad \text{the last 12 columns of } {}^{n-1}_{n}R\Pi_n] \qquad (16)
\end{aligned}
$$

Hence

$$
\tau_{n-1} = \hat{e}_1^T \Pi_{n-1}\theta_u + k.t. \qquad (17)
$$

By inspecting (9) and (11)-(17), we obtain the following

$$
Y_u(q, \dot{q}, \ddot{q}) = \begin{bmatrix} \hat{e}_1^T \Pi_1 \\ \hat{e}_1^T \Pi_2 \\ \vdots \\ \hat{e}_1^T \Pi_n \end{bmatrix} \qquad (18)
$$

where $\Pi_i$ $(1 \le i \le n)$ can be computed recursively as follows.

**Step 1.** Compute

$$
\Pi_{n+1} = [z_1 \quad \Phi_{n+1} \quad E_{n+1}]
$$

where $\Phi_{n+1}$ is defined by (10), i.e.

$$
\begin{aligned}
\Phi_{n+1} &= [-\Omega({}^{n+1}\dot{v}_{n+1}) \quad \tilde{\Psi}({}^{n+1}\omega_{n+1}, {}^{n+1}\dot{\omega}_{n+1})] \\
E_{n+1} &= B({}^{n+1}\dot{\omega}_{n+1}) + \Omega({}^{n+1}\omega_{n+1}) B({}^{n+1}\omega_{n+1}) \\
&= B({}^n\dot{\omega}_n) + \Omega({}^n\omega_n) B({}^n\omega_n) \\
&= E_n
\end{aligned}
$$

and

$$
A_{n+1} = [{}^{n+1}\dot{v}_{n+1} \quad \Omega({}^{n+1}\dot{\omega}_{n+1}) - \|{}^{n+1}\omega_{n+1}\|^2 I_3 + {}^{n+1}\omega_{n+1}^{n+1}\omega_{n+1}^T]
$$

**Step 2.** For $i = n, n-1, \ldots, 1$, compute

$$
\begin{aligned}
\Pi_i &= [(\text{the first 4 columns of } {}^{i}_{i+1}R\Pi_{i+1}) + A_i \\
&\quad \text{the last 12 columns of } {}^{i}_{i+1}R\Pi_{i+1}]
\end{aligned}
$$

with

$$
A_i = \Omega({}^i p_{i+1})\, {}^{i}_{n+1}R A_{n+1}
$$

## IV. An Illustrative Example

Consider a 2 d.o.f. planar arm handling a rectangular bar with unknown but uniform material density as shown in Fig.1. The center of mass of each robot link is assumed to be at its distal end, and the length, mass and moment of inertia of link $i$ $(i = 1, 2)$ are denote by $l_i$, $m_i$, $I_i$, respectively. Hence ${}^3p_{obj}$ and ${}^{c_{obj}}I_3$ assume the form of

$$
{}^3p_{obj} = [p_x \;\; 0 \;\; 0] \quad \text{and} \quad {}^{c_{obj}}I_3 = diag(I_{xx},\, I_{yy},\, I_{zz})
$$

we now follow the steps described in Sec. III to obtain $Y_u$.

**Step 1.** To compute

$$
\Pi_3 = [z_1 \quad \Phi_3 \quad E_3]
$$

note

$$
\begin{aligned}
-\Omega({}^3\dot{v}_3) &= \begin{bmatrix} 0 & \dot{v}_{33} & -\dot{v}_{32} \\ -\dot{v}_{33} & 0 & \dot{v}_{31} \\ \dot{v}_{32} & -\dot{v}_{31} & 0 \end{bmatrix} \\
{}^3\dot{v}_3 &= {}^3_2R({}^2\dot{\omega}_2 \times {}^2p_3 + {}^2\omega_2 \times ({}^2\omega_2 \times {}^2p_3) + {}^2\dot{v}_2) \\
&= \begin{bmatrix} l_1 s_2 \ddot{\theta}_1 - l_2(\dot{\theta}_1 + \dot{\theta}_2)^2 - l_1 c_2 \dot{\theta}_1^2 + g s_{12} \\ l_1 c_2 \ddot{\theta}_1 + l_2(\ddot{\theta}_1 + \ddot{\theta}_2) + l_1 s_2 \dot{\theta}_1^2 + g c_{12} \\ 0 \end{bmatrix} \\
&\equiv [\dot{v}_{31} \;\; \dot{v}_{32} \;\; \dot{v}_{33}]^T
\end{aligned}
$$

with $c_i = cos(\theta_i)$, $s_i = sin(\theta_i)$, $c_{ij} = cos(\theta_i + \theta_j)$, and $s_{ij} = sin(\theta_i + \theta_j)$.

Further we compute

$$
\tilde{\Psi}_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & -\ddot{\theta}_{12} & \dot{\theta}_{12}^2 \\ 0 & 0 & 0 & 0 & -\dot{\theta}_{12}^2 & -\ddot{\theta}_{12} \\ \ddot{\theta}_{12} & \ddot{\theta}_{12} & 0 & 0 & 0 & 0 \end{bmatrix}
$$

and

$$
\Phi_3 = \begin{bmatrix} 0 & 0 & \phi_{13} & 0 & 0 & 0 & 0 & -\ddot{\theta}_{12} & \dot{\theta}_{12}^2 \\ 0 & 0 & \phi_{23} & 0 & 0 & 0 & 0 & -\dot{\theta}_{12}^2 & -\ddot{\theta}_{12} \\ \phi_{31} & \phi_{32} & 0 & \ddot{\theta}_{12} & \ddot{\theta}_{12} & 0 & 0 & 0 & 0 \end{bmatrix}
$$

where

$$
\begin{aligned}
\dot{\theta}_{12} &= \dot{\theta}_1 + \dot{\theta}_2 \\
\ddot{\theta}_{12} &= \ddot{\theta}_1 + \ddot{\theta}_2 \\
\phi_{13} &= -l_2(\ddot{\theta}_1 + \ddot{\theta}_2) - l_1 c_2 \ddot{\theta}_1 - l_1 s_2 \dot{\theta}_1^2 - g c_{12} \\
\phi_{23} &= -l_2(\dot{\theta}_1 + \dot{\theta}_2)^2 + l_1 s_2 \ddot{\theta}_1 - l_1 c_2 \dot{\theta}_1^2 + g s_{12} \\
\phi_{31} &= l_2(\ddot{\theta}_1 + \ddot{\theta}_2) + l_1 c_2 \ddot{\theta}_1 + l_1 s_2 \dot{\theta}_1^2 + g c_{12} \\
\phi_{32} &= l_2(\dot{\theta}_1 + \dot{\theta}_2)^2 - l_1 s_2 \ddot{\theta}_1 + l_1 c_2 \dot{\theta}_1^2 - g s_{12}
\end{aligned}
$$

and

$$E_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & -\ddot{\theta}_{12} & \dot{\theta}_{12}^2 \\ 0 & 0 & 0 & 0 & -\dot{\theta}_{12}^2 & -\ddot{\theta}_{12} \\ 0 & 0 & \ddot{\theta}_{12} & 0 & 0 & 0 \end{bmatrix}$$

Note that $A_3$ is given by

$$\begin{aligned} A_3 &= [{}^3\dot{v}_3 \quad \Omega({}^3\dot{\omega}_3) - \|{}^3\omega_3\|^2 I_3 + {}^3U_3] \\ &= \begin{bmatrix} a_{311} & -\dot{\theta}_{12}^2 & -\ddot{\theta}_{12} & 0 \\ a_{321} & \ddot{\theta}_{12} & -\dot{\theta}_{12}^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

where

$$\begin{aligned} a_{311} &= l_1 s_2 \ddot{\theta}_1 - l_2(\dot{\theta}_1 + \dot{\theta}_2)^2 - l_1 c_2 \dot{\theta}_1^2 + g s_{12} \\ a_{321} &= l_1 c_2 \ddot{\theta}_1 + l_2(\ddot{\theta}_1 + \ddot{\theta}_2) + l_1 s_2 \dot{\theta}_1^2 + g c_{12} \end{aligned}$$

**Step 2.** For $i = 2, 1$,

$$\begin{aligned} \Pi_i &= [(\text{the first 4 columns of } {}^{i}_{i+1}R\Pi_{i+1}) + A_i \\ &\quad \text{the last 12 columns of } {}^{i}_{i+1}R\Pi_{i+1}] \end{aligned}$$

with

$$A_i = \Omega({}^i p_{i+1})\, {}^{i}_{n+1}R A_{n+1}$$

So

$$\Pi_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \pi_{231} & \pi_{232} & \ddot{\theta}_{12} & \ddot{\theta}_{12} \end{bmatrix}$$

where

$$\begin{aligned} \pi_{231} &= l_2^2 \ddot{\Theta}_{12} + l_1 l_2 c_2 \ddot{\theta}_1 + l_1 l_2 s_2 \dot{\theta}_1^2 + l_2 g c_{12} \\ \pi_{232} &= 2 l_2 \ddot{\Theta}_{12} + l_1 c_2 \ddot{\theta}_1 + l_1 s_2 \dot{\theta}_1^2 + g c_{12} \end{aligned}$$

and

$$\Pi_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \pi_{131} & \pi_{132} & \ddot{\theta}_{12} & \ddot{\theta}_{12} \end{bmatrix}$$

where

$$\begin{aligned} \pi_{131} &= (l_1^2 + l_2^2 + 2 l_1 l_2 c_2)\ddot{\theta}_1 + (l_2^2 + l_1 l_2 c_2)\ddot{\theta}_2 \\ &\quad - l_1 l_2 s_2 \dot{\theta}_2 (2\dot{\theta}_1 + \dot{\theta}_2) + g(l_1 c_1 + l_2 c_{12}) \\ \pi_{132} &= 2(l_2 + l_1 c_2)\ddot{\theta}_1 + (2 l_2 + l_1 c_2)\ddot{\theta}_2 \\ &\quad - l_1 s_2 \dot{\theta}_2 (2\dot{\theta}_1 + \dot{\theta}_2) + g c_{12} \end{aligned}$$

Finally

$$\begin{aligned} Y_u &= \begin{bmatrix} \hat{e}_1^T \Pi_1 \\ \hat{e}_1^T \Pi_2 \end{bmatrix} \\ &= \begin{bmatrix} \pi_{131} & \pi_{132} & \ddot{\theta}_{12} & \ddot{\theta}_{12} \\ \pi_{131} & \pi_{132} & \ddot{\theta}_{12} & \ddot{\theta}_{12} \end{bmatrix} \end{aligned}$$

and

$$\tau = Y_u \theta_u + k.t.$$

where $\theta_u = [m \;\; m p_x \;\; m p_x^2 \;\; I_{zz}]^T$.

**Acknowledgement**

The authors are members of the Institute for Robotics and Intelligent Systems (IRIS) and wish to acknowledge the support of the Networks of Excellence Program of the Government of Canada and the participation of PRECARN Associates Inc.

## References

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

[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*, pp.990-995, 1985.

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

[4] C. P. Neuman and P. K. Khosla, "Identification of robot dynamics: An application of recursive estimation", in K. S. Narendra (ed.) *Proc. 4th Yale Workshop on Appl. of Adaptive Syst. Theory*, Yale Univ., 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, *Adaptive Control of Mechanical Manipulators*, Reading, MA: Addison-Wesley, 1988.

[7] W.-S. Lu and Q.-H. Meng, "A closed-form formula for the manipulator regressor", in *Proc. Canadian Conf. ECE*, pp.252-254, Ottawa, Sept. 1990.

[8] J. Y. S. Luh, M. W. Walker, and R. P. Paul, "On-line computational scheme for mechanical manipulator", *Trans. ASME Dynamic Syst., Meas., Contr.*, vol. 102, pp.69-76, June 1980.

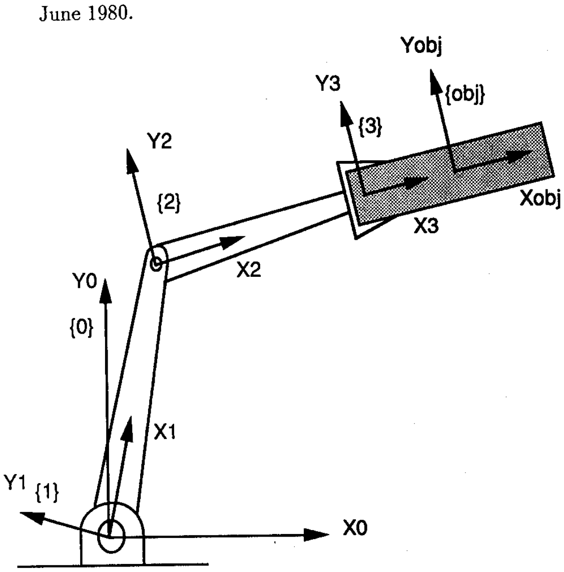


Fig. 1 The 2 d.o.f. planar robot handling a rectangular payload