Estimation of Robot Dynamics Parameters: Theory and Application

Pradeep K. Khosla

CMU-RI-TR-87-8

Department of Electrical and Computer Engineering
The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

March 1987

This research is based upon the work supported by the National Science Foundation under Grant ECS-8320364.

Table of Contents

1 Introduction	1
2 Newton-Euler Dynamics Model	3
2.1 Newton-Euler Formulation	3
2.2 Transformation of Inertia Tensor	6
3 Reformulation of the N-E Dynamics Model	7
4 Identification Algorithm	8
5 Experimental Implementation	11
5.1 Obtaining the Joint Acceleration	11
5.2 Trajectory Selection	12
5.3 Experimental Results	12
5.4 Identifiable Parameters	15
6 Summary	16

List of Tables

Table 1: Kinematic and Dynamic Parameters 5

Table 2: Experimental Results for the CMU DD Arm II 13

Table 3: Experimental Results (cont'd.) 14

Table 4: Experimental Results (cont'd.) 15

Abstract

This paper presents an algorithm for estimating the dynamics parameters of an N degrees-of-freedom robotic manipulator. In our previous work [8] it was shown that the Newton-Euler model model which is nonlinear in the dynamic parameters can be transformed into an equivalent *modified* model which is linear in dynamic parameters. To introduce our identification algorithm, we cast this *modified* Newton-Euler model in a form wherein the joint torques/forces are expressed as a product of a matrix and a vector of dynamics parameters. The elements of this matrix are a function of the joint variables and the kinematic parameters only. The dynamics parameters are then estimated from this linear (in dynamics parameters) formulation using the least squares estimation method. We have implemented our algorithm and the results of this experimental implementation to estimate the dynamics parameters of the six degrees-of-freedom CMU DD Arm II are presented. The estimated dynamics parameters have also been used to evaluate the effect of dynamics compensation in model-based manipulator control methods.

1. Introduction

The model-based control schemes such as the computed-torque [14] and resolved-acceleration [12] methods incorporate the complete dynamics model of the manipulator in the control law. Further, in order to achieve accurate trajectory tracking these schemes assume that the robot dynamics and kinematics parameters are known accurately. This assumption, however, is hardly valid in practice and there exists a need to develop techniques to estimate the dynamics and the kinematic parameters of a manipulator. In practice, it is also necessary to identify on-line the mass and inertial characteristics of the payload in order to achieve accurate trajectory tracking with varying payload.

Earlier work in identification of robot dynamics concentrated on estimating the mass of the payload. Paul [19] presented two techniques with the assumption that the manipulator is at rest. The first method used the joint torques/forces, and the second method a wrist torque/force sensor. Coiffet [3] extended this technique, for a manipulator at rest, to estimate also the center-of-mass of the payload. By using special test torques and moving only one degree-of-freedom at a time, the moments-of-inertia of the payload can also be estimated. Olsen and Bekey [17] proposed an identification algorithm that was restricted to rotary joint manipulators. Further, it required special test motions that involved rotations about one axis at a time. Recently, the work was expanded to general purpose manipulators and without any trajectory restrictions [18]. However, in estimating the center-of-mass of the links, they assumed that the accelerations due to rotations are insignificant compared to the gravitational and translational acceleration. This restrictive assumption allows one to approximate the N-E dynamic equations such that they are linear in all the dynamics parameters. In this paper, we show that such an approximation is unnecessary because it is possible to obtain a nonlinear transformation that will make the Newton-Euler recursions linear in all the dynamics parameters [8, 7, 16]. Other researchers [15, 1] have also independently proposed algorithms for estimating the dynamics parameters of a manipulator. We outline the salient features of our algorithm and the differences from other algorithms later in this paper.

In this paper, we present an algorithm to estimate the dynamics parameters of a robot from the measurements of its inputs (actuating torques/forces) and outputs (joint positions, velocities and accelerations). To facilitate the identification procedure, we modify the Newton-Euler formulation, through a nonlinear transformation, so that it becomes linear in the dynamic parameters. We then reformulate the backward recursions of the Newton-Euler model to obtain the joint torques/forces as a product of a matrix (that is a function of only the joint variables and the kinematic parameters) and a vector of inertial parameters. This formulation is very suitable from the estimation point-of-view as it represents a set of linear equations that can be solved using standard linear estimation techniques [4, 5, 6]. We have implemented our identification algorithm to estimate the dynamics parameters of the six degrees-of-freedom CMU DD Arm II [7].

Our general-purpose algorithm is suited for both on-line and off-line applications: in off-line identification only one link of the robot is commanded to move for the purpose of parameter estimation, whereas in on-line identification the parameters are estimated while the robot is in motion performing the task in hand. We can adopt the strategy of estimating off-line the dynamics parameters of the robot and then estimating on-line the inertial characteristics of the payload. This procedure improves the robustness of the estimation, decreases the computational requirements, and adapts to varying payloads. We have implemented our identification algorithm to estimate the dynamics parameters of the six degrees-of-freedom CMU DD Arm II [7]. The obtained dynamics parameters were then used to experimentally implement and evaluate the effect of dynamics compensation in model-based schemes [9, 10].

This paper is organized as follows: In section 2, we review the Newton-Euler formulation and identify its properties applicable to robot parameter identification. We then derive, in section 3, our identification procedures for a general-purpose N degree-of-freedom robot. In section 4, we evaluate the performance of our algorithm on the two case study robots. The experimental results for the six degrees-of-freedom CMU DD Arm II are presented in section 5 and finally, in section 6, we draw our conclusions.

2. Newton-Euler Dynamics Model

2.1. Newton-Euler Formulation

The Newton-Euler formulation [13, 2] shown in equations (1)-(9) computes the inverse dynamics (ie., joint torques/forces from joint positions, velocities, and accelerations) based on two sets of recursions: the *forward* and *backward* recursions. The forward recursions (1)-(3) transform the kinematic varables from the base to the end-effector. The initial conditions (for i=0) assume that the manipulator is at rest in the gravitational field. The backward recursions (4)-(9) transform the forces and moments from the end-effector to the base, and culminate with the calculation of the joint torques/forces.

$$\omega_{i+1} = \begin{cases} \mathbf{A}_{i+1}^T[\,\omega_i + \mathbf{z}_o\dot{\theta}_{i+1}] & \text{rotational} \\ \mathbf{A}_{i+1}^T\,\omega_i & \text{translational} \end{cases} \tag{1}$$

$$\dot{\omega}_{i+1} = \begin{cases} \mathbf{A}_{i+1}^T[\dot{\omega}_i + \mathbf{z}_o\ddot{\theta}_{i+1} + \omega_i \times (\mathbf{z}_o\dot{\theta}_{i+1})] & \text{rotational} \\ \mathbf{A}_{i+1}^T\,\dot{\omega}_i & \text{translational} \end{cases} \tag{2}$$

$$\dot{\mathbf{v}}_{i+1} = \begin{cases} \mathbf{A}_{i+1}^T\dot{\mathbf{v}}_i + \dot{\omega}_{i+1} \times \mathbf{p}_{i+1} + \omega_{i+1} \times (\omega_{i+1} \times \mathbf{p}_{i+1}) & \text{rotational} \\ \mathbf{A}_{i+1}^T[\dot{\mathbf{v}}_i + \mathbf{z}_o\ddot{d}_{i+1} + 2\omega_i \times (\mathbf{z}_o\dot{d}_{i+1})] & \\ \quad + \dot{\omega}_{i+1} \times \mathbf{p}_{i+1} + \omega_{i+1} \times (\omega_{i+1} \times \mathbf{p}_{i+1}) & \text{translational} \end{cases} \tag{3}$$

$\omega_0 = \dot{\omega}_0 = \mathbf{v}_0 = \mathbf{0}$ initial Conditions

$\dot{\mathbf{v}}_0 = [g_x\ g_y\ g_z]^T$ gravitational acceleration

$$\dot{\mathbf{v}}_i^* = \dot{\omega}_i \times \mathbf{s}_i + \omega_i \times (\omega_i \times \mathbf{s}_i) + \dot{\mathbf{v}}_i \tag{4}$$

$$\mathbf{F}_i = m_i \dot{\mathbf{v}}_i^* \tag{5}$$

$$\mathbf{N}_i = \mathbf{I}_i \dot{\omega}_i + \omega_i \times (\mathbf{I}_i \omega_i) \tag{6}$$

$$\mathbf{f}_i = \mathbf{A}_{i+1} \mathbf{f}_{i+1} + \mathbf{F}_i \tag{7}$$

$$\mathbf{n}_i = \mathbf{A}_{i+1} \mathbf{n}_{i+1} + \mathbf{p}_i \times \mathbf{f}_i + \mathbf{N}_i + \mathbf{s}_i \times \mathbf{F}_i \tag{8}$$

$$\tau_i = \begin{cases} \mathbf{n}_i^T (\mathbf{A}_i^T \mathbf{z}_o) & \text{rotational} \\ \mathbf{f}_i^T (\mathbf{A}_i^T \mathbf{z}_o) & \text{translational} \end{cases} \tag{9}$$

$\mathbf{f}_{N+1}$: external force at the end-effector.
$\mathbf{n}_{N+1}$: external moment at the end-effector.

From equations (1)-(9), we note the following properties:

1. The Newton-Euler model is *linear* in the classical link inertia tensors $\mathbf{I}_i$.

   This property follows directly from the backward recursions in (5)-(9). The joint torque/force τ_i in (9) is linear in the moment $\mathbf{n}_i$. In the recursion for the moment $\mathbf{n}_i$ in (8), the net moment $\mathbf{N}_i$ exerted on link i appears additively. Finally, the moment $\mathbf{N}_i$ in (6) is linear in the classical link inertia tensor $\mathbf{I}_i$.

2. For rotational joints, the Newton-Euler model is *nonlinear* in the center-of-mass vectors $\mathbf{s}_i$.

   From equations (4) and (5), the net force $\mathbf{F}_i$ is linear in the center-of-mass vector $\mathbf{s}_i$. The vector cross product $\mathbf{s}_i \times \mathbf{F}_i$ in (8) is thereby nonlinear (quadratic) in $\mathbf{s}_i$. Hence, the torque τ_i for a rotational joint in (9) is nonlinear in the center-of-mass vector $\mathbf{s}_i$. It must be noted that for translational joints the center-of-mass vectors appear linearly.

3. The Newton-Euler model is *nonlinear* in the kinematic parameter vectors $\mathbf{p}_i$.

   From equations (3)-(5) and (7), the link force $\mathbf{f}_i$ is linear in the vector, $\mathbf{p}_i$. The vector cross product $\mathbf{p}_i \times \mathbf{f}_i$ in (8) is thereby nonlinear in $\mathbf{p}_i$. Hence, the torque/force τ_i in (9)

Table 1: Kinematic and Dynamic Parameters

m_i — Total mass of link i

τ_i — Joint torque/force at joint i

ω_i and $\dot{\omega}_i$ — Angular velocity and acceleration of the i-th coordinate frame

$\mathbf{v}_i$ and $\dot{\mathbf{v}}_i$ — Linear velocity and acceleration of the i-th coordinate frame

$\mathbf{v}_i^*$ and $\dot{\mathbf{v}}_i^*$ — Linear velocity and acceleration of the center-of-mass of link i

$\mathbf{F}_i$ and $\mathbf{N}_i$ — Net force and moment exerted on link i

$\mathbf{f}_i$ and $\mathbf{n}_i$ — Force and moment exerted on link i by link $i-1$

$\mathbf{p}_i$ — Position of the i-th coordinate frame with respect to the $(i-1)$-th coordinate frame: $\mathbf{p}_i=[a_i \;\; d_i\, sin\alpha_i \;\; d_i\, cos\alpha_i]^T$

$\mathbf{s}_i$ — Position of the center-of-mass of link i: $\mathbf{s}_i=[s_{ix} \;\; s_{iy} \;\; s_{iz}]^T$

$\mathbf{z}_o$ — $=[0 \;\; 0 \;\; 1]^T$

$\mathbf{A}_i$ — Orthogonal rotation matrix which transforms a vector in the i-th coordinate frame to a coordinate frame which is parallel to the $(i-1)$-th coordinate frame:

$$\mathbf{A}_i= \begin{matrix} cos\theta_i & -cos\alpha_i\, sin\theta_i & sin\alpha_i\, sin\theta_i \\ sin\theta_i & cos\alpha_i\, cos\theta_i & -sin\alpha_i\, cos\theta_i \\ 0 & sin\alpha_i & cos\alpha_i \end{matrix}$$

for $i=1,2, \cdots ,N$, where $\mathbf{A}_{N+1} \triangleq \mathbf{E}$.

$\mathbf{I}_i$ — Classical inertia tensor of link i about the center-of-mass of link i (and parallel to the i-th coordinate frame); with principal inertias I_{ixx}, I_{iyy} and I_{izz}; and cross-inertias I_{ixy}, I_{ixz} and I_{iyz}.

is nonlinear in the vector, $\mathbf{p}_i$.

4. The dynamic equations of links $i+1$ through N are independent of the mass m_i and the classical inertia tensor $\mathbf{I}_i$ of link i.

   This physically intuitive property is an immediate consequence of the backward recursions.

The proofs of the above properties are presented by Khosla [8] and have been omitted here for the sake of brevity. In summary, the classical link inertia tensors $\mathbf{I}_i$ and the link masses m_i appear linearly in the Newton-Euler dynamics model, but the link masses are multiplied by linear and/or quadratic functions of the center-of-mass vectors $\mathbf{s}_i$ and nonlinear functions of the joint position variables θ_i. Further, it can be shown that if the Newton-Euler model in equations (1)-(9) is reformulated such that the link inertia tensors are expressed about the link coordinate frames instead of the link center-of-mass coordinate frame, the modified Newton-Euler formulation will be linear in the center of the mass vectors $\mathbf{s}_i$ [7].

2.2. Transformation of Inertia Tensor

Let $C_i=(x_i, y_i, z_i)$ be a Denavit-Hartenberg coordinate frame for link i and let $C_i^* = (x_i^*, y_i^*, z_i^*)$ be a coordinate frame which is fixed at the center-of-mass of link i and whose axes are parallel with those of C_i. From the definition, $\mathbf{s}_i$ is the translational vector from the origin of the link coordinate frame C_i to to the origin of the center-of-the-mass coordinate frame C_i^*.

If $\mathbf{I}_i$ is the classical link inertia tensor about the center-of-mass of link i, the corresponding inertia tensor $\mathbf{I}'_i$ about the link i coordinate frame C_i is computed according to the parallel-axis theorem or Steiner's law:

$$\mathbf{I}'_i=\mathbf{I}_i+m_i(\mathbf{s}_i^T\mathbf{s}_i\mathbf{E}-\mathbf{s}_i\mathbf{s}_i^T) \tag{10}$$

where $\mathbf{E}$ is the 3×3 identity matrix. This transformation of the inertia tensor when substituted in the N-E model absorbs the terms that are quadratic in $\mathbf{s}_i$ thus resulting in the *modified* N-E formulation that is linear in the dynamics parameters [8, 7].

Properties 1 and 2 together with this transformation lay the foundation for our identification algorithms, and Property 4 will be used to derive our off-line identification algorithm.

3. Reformulation of the N-E Dynamics Model

The N-E dynamics model is reformulated by using the nonlinear transformation in 10 and compacting the backward recursions in (6)-(9) into a single matrix-vector equation. To facilitate the reformulation, we introduce the following vector identities:

$$\mathbf{a}\times(\mathbf{b}\times(\mathbf{b}\times\mathbf{a})) = \mathbf{b}\times[\mathbf{a}^T\mathbf{a}\mathbf{E} - \mathbf{a}\mathbf{a}^T]\mathbf{b} \tag{11}$$

$$\mathbf{a}\times(\mathbf{b}\times\mathbf{a}) = [\mathbf{a}^T\mathbf{a}\mathbf{E} - \mathbf{a}\mathbf{a}^T]\mathbf{b} \tag{12}$$

where $\mathbf{a}$ and $\mathbf{b}$ are 3×1 vectors and $\mathbf{E}$ is the 3×3 identity matrix.

Upon substituting (4)-(6) in equation (8), we obtain:

$$\begin{aligned}\mathbf{n}_i = {} & \mathbf{A}_{i+1}\mathbf{n}_{i+1} + \mathbf{p}_i\times\mathbf{f}_i + \mathbf{I}_i\dot{\omega}_i + \omega_i\times(\mathbf{I}_i\omega_i) \\ & + m_i\mathbf{s}_i\times[\dot{\omega}_i\times\mathbf{s}_i + \omega_i\times(\omega_i\times\mathbf{s}_i) + \dot{\mathbf{v}}_i].\end{aligned} \tag{13}$$

The above equation can be further simplified by using the identities (11),(12) and equation (10), and written as:

$$\mathbf{n}_i = \mathbf{A}_{i+1}\mathbf{n}_{i+1} + \mathbf{p}_i\times\mathbf{f}_i + \mathbf{I}'_i\dot{\omega}_i + \omega_i\times(\mathbf{I}'_i\omega_i) + m_i\mathbf{s}_i\times\dot{\mathbf{v}}_i \tag{14}$$

where $\mathbf{I}'_i$ is the inertia of link i expressed about the link coordinate frame. Equation (14) is the equivalent of the backward recursions in (4)-(6) and (8) and will henceforth be used in our development.

As the next step in our reformulation, we define a 6×1 vector $\mathbf{g}_i$ as

$$\mathbf{g}_i = [\,\mathbf{f}_i \quad \mathbf{n}_i\,]^T \tag{15}$$

and combine (7) and (13) in a single matrix-vector equation:

$$\mathbf{g}_i = \mathbf{R}_{i+1}\mathbf{g}_{i+1} + \mathbf{G}_i \tag{16}$$

where $\mathbf{R}_{i+1}$ is the six by six *pseudo-rotation* matrix

$$R_{i+1} = \begin{bmatrix} A_{i+1} & 0 \\ [p_i X] A_{i+1} & A_{i+1} \end{bmatrix}$$

and G_i is defined as:

$$G_i = \begin{bmatrix} F_i \\ N_i' \end{bmatrix}$$

with N_i' being:

$$N_i' = I_i' \dot{\omega}_i + \omega_i X (I_i' \omega_i) + m_i s_i X \dot{v}_i + m_i p_i X [\dot{\omega}_i X s_i + \omega_i X(\omega_i X s_i) + \dot{v}_i]$$

The actuating torques are chosen from the vector g_i as:

$$\tau_i = g_i^T \begin{bmatrix} A_i^T & 0 \\ 0 & A_i^T \end{bmatrix} \begin{bmatrix} \sigma_i z_0 \\ (1-\sigma_i) z_0 \end{bmatrix} \tag{17}$$

where σ_i=1 for a *i-th* rotary joint and σ_i=0 for a *i-th* translational joint. The complete dynamic robot model is given by the forward recursions in equations (1)-(3) and the backward recursion in (16), the selection of torques according to (17).

4. Identification Algorithm

The identification problem is to estimate all of the kinematic and dynamic parameters that affect the link torques/forces. The Denavit-Hartenberg parameters constitute the kinematic parameters, and the link masses, link inertias, and center-of-mass vectors are the dynamic parameters. Improving the kinematic accuracy of a manipualtor by estimating the kinematics parameters has received a lot of attention and algorithms to estimate these have been proposed [21, 20]. Further, these algorithms have also been experimentally implemented to demonstrate the improved kinematic accuracy. Consequently, in our development, we will assume that the kinematics parameters are accurately known and the problem then is to accurately estimate the dynamics parameters.

To develop the identification algorithm, we expand the recursions in equation (16) to obtain explicit expressions for the vector $\mathbf{g}_i$. To facilitate the development, we use the following notation for denoting the cross product of two vectors and the multiplication of a vector by a matrix. If ω_i and $\mathbf{a}$ are 3×1 vectors, then

$\omega_i\times\mathbf{a} = [\omega_i\times]\mathbf{a}$

where

$$[\omega_i\mathrm{x}] = \begin{bmatrix} 0 & -\omega_{iz} & \omega_{iy} \\ \omega_{iz} & 0 & -\omega_{ix} \\ -\omega_{iy} & \omega_{ix} & 0 \end{bmatrix}$$

and the multiplication of the 3×3 classical inertia matrix $\mathbf{I}_i$ with the vector ω_i is denoted as

$\mathbf{I}_i\omega_i = [\bullet\omega_i]\mathbf{I}_i$

where

$$[.\omega_i] = \begin{bmatrix} \omega_{ix} & \omega_{iy} & \omega_{iz} & 0 & 0 & 0 \\ 0 & \omega_{ix} & 0 & \omega_{iy} & \omega_{iz} & 0 \\ 0 & 0 & \omega_{ix} & 0 & \omega_{iy} & \omega_{iz} \end{bmatrix}$$

and

$\mathbf{I}_i = [I_{ixx}\ I_{ixy}\ I_{ixz}\ I_{iyy}\ I_{iyz}\ I_{izz}]^T$

For a six degrees-of-freedom manipulator, assuming that the vector of externally applied forces and moments is zero, and expanding the recursions in equation (16), we obtain:

$$\begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \\ g_5 \\ g_6 \end{bmatrix} = \begin{bmatrix} K_1 & R^1{}_2K_2 & R^1{}_3K_3 & R^1{}_4K_4 & R^1{}_5K_5 & R^1{}_6K_6 \\ 0 & K_2 & R^2{}_3K_3 & R^2{}_4K_4 & R^2{}_5K_5 & R^2{}_6K_6 \\ 0 & 0 & K_3 & R^3{}_4K_4 & R^3{}_5K_5 & R^3{}_6K_6 \\ 0 & 0 & 0 & K_4 & R^4{}_5K_5 & R^4{}_6K_6 \\ 0 & 0 & 0 & 0 & K_5 & R^5{}_6K_6 \\ 0 & 0 & 0 & 0 & 0 & K_6 \end{bmatrix} \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \\ \varphi_4 \\ \varphi_5 \\ \varphi_6 \end{bmatrix}$$

where $\mathbf{R}^i{}_j = \mathbf{R}_{i+1}\mathbf{R}_{i+2}\ldots\ldots\mathbf{R}_j$, the matrix $\mathbf{K}_i$ is:

$$\mathbf{K}_i = \begin{bmatrix} \dot{\mathbf{v}}_i & [\dot{\omega}_i\mathrm{x}] + [\omega_i\mathrm{x}][\omega_i\mathrm{x}] & 0 \\ [\mathbf{p}_i\mathrm{x}]\dot{\mathbf{v}}_i & [-\dot{\mathbf{v}}_i\mathrm{x}] + [\mathbf{p}_i\mathrm{x}][\dot{\omega}_i\mathrm{x}] + [\mathbf{p}_i\mathrm{x}][\omega_i\mathrm{x}][\omega_i\mathrm{x}] & [.\dot{\omega}_i] + [\omega_i\mathrm{x}][\omega_i\mathrm{x}] \end{bmatrix}$$

and the vector ϕ_i is:

$$\phi_i = [m_i \;\; m_i s_{ix} \;\; m_i s_{iy} \;\; m_i s_{iz} \;\; I'_{ixx} \;\; I'_{ixy} \;\; I'_{ixz} \;\; I'_{iyy} \;\; I'_{iyz} \;\; I'_{izz}]^T$$

In the sequel, we will refer to $\mathbf{K}_i$ and ϕ_i as the *kinematic* matrix and vector of dynamics parameters of link i, respectively. An et al. [1] and Mukerjee and Ballard [15] have independently derived a similar form for the kinematic matrix with one important difference. In their derivation, the element (strictly speaking it is a 3 vector) in second row and first column of the matrix $\mathbf{K}_i$ is zero. This implies that the dynamics equations of a manipulator are independent of terms that multiply only the mass element of the dynamics parameter vector (for a rotary joint). We have verified the correctness of our form of the kinematic matrix by symbolically deriving the dynamics equations of a three degrees-of-freedom manipulator using the standard Newton-Euler formulation [13] and comparing the results with those obtained from the modified formulation of this paper.

The identification problem is to obtain the estimates of the elements of ϕ_i (for i=1,2, . . . ,N) and is formulated as the solution of a set of linear equations

$$\tau = \mathbf{M}\phi \tag{18}$$

for each sample point in the trajectory. In the above equation, the 6×1 vector τ and the 60×1 vector ϕ are:

$$\tau = [\tau_1 \;\; \tau_2 \;\; \tau_3 \;\; \tau_4 \;\; \tau_5 \;\; \tau_6]^T$$

$$\phi = [\phi_1 \;\; \phi_2 \;\; \phi_3 \;\; \phi_4 \;\; \phi_5 \;\; \phi_6]^T$$

Further, the matrix $\mathbf{M}$ is formed by selecting the row of $\mathbf{g}_i$ corresponding to τ_i. If the twist angles of the manipulator α_i are assumed to be either 0 or ±90 degrees then equation (17) may be used to select the actuating torques τ_i. The above formulation of the identification problem is particularly suited for computer implementation and is used to estimate the dynamics parameters of the CMU DD Arm II.

5. Experimental Implementation

Experimental implementation of the identification algorithm requires the knowledge of the applied joint torques and the measured joint positions, velocities and accelerations. Each joint of the DD Arm II is instrumented to measure the position and the velocity. The applied joint torques are assumed to be the same as the torques computed from the control law. This assumption is valid because we use current controlled servo motors and has also been verified by experimentation. Further, the joint acceleration is obtained by differentiating the measured velocity.

5.1. Obtaining the Joint Acceleration

The operation of obtaining the derivative of a set of data is inherently noisy because the differentiator essentially behaves like a high pass-filter. And this effect is further accentuated if the measured data is known to have some noise. In such a circumstance a commonly used method is to low-pass filter the measured data and then differentiate the resultant signal. This procedure serves to reduce the noise in the differentiated signal at the cost of incorporating a phase shift and hence the loss of fidelity.

Another method involves using the principle of least-squares for solving the problem of differentiation [11]. In this method, the differentiating filter is designed by fitting a second-order parabola to five conseqeutive points with the assumption that the derivative does not change much during the period of the observations. This assumption is especially true since we sample the position and the velocity of the joints every 2 ms. As the five data points, in general, cannot be guaranteed to lie on a second-order curve, we obtain the coefficients of the parabola by using the principle of least-squares. The resulting filter is described by the following difference equation:

$$f'(x) = \frac{-2f(x-2T)-f(x-T)+f(x+T)+2f(x+2T)}{10T}$$

where the symbol $'$ denotes the derivative and T is the sampling period in seconds. The above filter obtains the derivative of the function $f(x)$ at the point x by using the two immediate neighbors on both sides and thus represents a noncausal operation for real-time implementation. However, if we are able to tolerate a delay of two sampling periods then the filter can be made causal by shifting the data by two sampling instants to obtain:

$$f'(x-2T) = \frac{-2f(x-4T)-f(x-3T)+f(x-T)+2f(x)}{10T}.$$

In the off-line implementation [8] of our identification algorithm, the noncausal nature of the filter presents no problem as all the data is known in advance. In order to obtain the joint acceleration from the measured joint velocity, we experimented with many methods of implementing differentiating filters and found the filter designed on the basis of the principle of least-squares to possess superior noise rejection properties.

5.2. Trajectory Selection

One of the important constituents of identification is the selection of input trajectories for exciting the system. The input trajectory must be such that it allows complete identification of the system. Such a trajectory is known as a *persistently exciting* trajectory [6]. Choosing a persistently exciting trajectory is sufficiently complex and has not been addressed in this research. However, a method to determine if a chosen trajectory is persistently exciting is presented by Khosla [7]. In the experimental implementation, we used this method to ensure that the trajectories chosen for identification of the dynamics parameters were persistently exciting.

5.3. Experimental Results

We implemented the numerical version of the identification algorithm together with the differentiating filter to obtain the estimates of the dynamics parameters of the CMU DD Arm II. The modeled values of the dynamics parameters were chosen to be the initial estimates and the data of a sample trajectory run recorded. We then estimated the dynamics parameters based on the off-line implementation of our algorithm, and these are depicted in Table (2), (3), and (4). Due to space restriction we have listed the estimated parameters of the first two links of the CMU DD Arm II. The estimated parameters of all six links may be found in [7].

The identification experiments were performed with two different, persistently exciting trajectories and two sets of initial values for the modeled dynamics parameters. In all the four experiments, the estimated values of the dynamics parameters were found to be within 5% of the values depicted in Table (2) through (4). This variation is practically

Table 2: Experimental Results for the CMU DD Arm II

Link	Parameter (Dimensions)	Initial Value	Estimated Value
6	I_{6xx} (kg-m^2)	0.000426	0.002092
	I_{6xy} (kg-m^2)	0.0	0.000011
	I_{6xz} (kg-m^2)	0.0	0.000022
	I_{6yy} (kg-m^2)	0.000421	0.001979
	I_{6yz} (kg-m^2)	0.0	0.000010
	I_{6zz} (kg-m^2)	0.000047	0.000310
	$m_6 s_{6x}$ (kg-m)	0.0	-0.000090
	$m_6 s_{6y}$ (kg-m)	0.0	-0.000187
	$m_6 s_{6z}$ (kg-m)	0.002199	0.008709
	m_6 (kg)	0.269	0.90018
5	I_{5xx} (kg-m^2)	0.002018	0.002602
	I_{5xy} (kg-m^2)	0.0	0.000302
	I_{5xz} (kg-m^2)	0.0	-0.000108
	I_{5yy} (kg-m^2)	0.001049	0.001349
	I_{5yz} (kg-m^2)	-0.000092	-0.000070
	I_{5zz} (kg-m^2)	0.001396	0.001211
	$m_5 s_{5x}$ (kg-m)	0.0	0.000981
	$m_5 s_{5y}$ (kg-m)	0.005130	0.006744
	$m_5 s_{5z}$ (kg-m)	-0.016784	-0.019689
	m_5+m_4 (kg)	2.817	3.0895

Table 3: Experimental Results for the CMU DD Arm II (contd.)

Link	Parameter (Dimensions)	Initial Value	Estimated Value
4	I_{4xx} (kg-m^2)	0.023775	0.030765
	I_{4xy} (kg-m^2)	0.0	-0.00100
	I_{4xz} (kg-m^2)	0.0	0.000302
	I_{4yy} (kg-m^2)	0.004055	0.003655
	I_{4yz} (kg-m^2)	0.003083	0.004207
	I_{4zz} (kg-m^2)	0.021652	0.029002
	$m_4 s_{4x}$ (kg-m)	0.0	-0.002402
	$m_4 s_{4y}$ (kg-m)	0.134764	0.160372
	$m_4 s_{4z}$ (kg-m)	0.043011	0.081462
3	I_{3xx} (kg-m^2)	0.014622	0.015192
	I_{3xy} (kg-m^2)	0.0	0.000726
	I_{3xz} (kg-m^2)	0.0	0.000109
	I_{3yy} (kg-m^2)	0.006615	0.006209
	I_{3yz} (kg-m^2)	0.001269	0.001872
	I_{3zz} (kg-m^2)	0.012432	0.014080
	$m_3 s_{3x}$ (kg-m)	0.0	0.015242
	$m_3 s_{3y}$ (kg-m)	-0.039703	-0.132251
	$m_3 s_{3z}$ (kg-m)	-0.012487	-0.040521
	m_3 (kg)	2.801	2.92106

Table 4: Experimental Results for the CMU DD Arm II

Link	Parameter (Dimensions)	Initial Value	Estimated Value
2	I_{2yy} (kg-m^2)	0.264736	0.322156
	$m_2 s_{2x}$ (kg-m)	-1.039971	-1.156482
	$m_2 s_{2z}$ (kg-m)	0.008722	0.008234
	m_2 (kg)	7.894	8.2501
1	I_{1zz} (kg-m^2)	1.193645	1.270784
	$m_1 s_{1x}$ (kg-m)	-5.925900	-6.478305
	m_1 (kg)	19.753000	20.152630

negligible and may be attributed to the noise in the measurements which tends to bias the estimates, and also to the errors in the kinematic parameters which also have a similar effect.

5.4. Identifiable Parameters

Each link of a manipulator is characterized by ten dynamics parameters: the link mass, the six classical inertias and the three elements of the center-of-mass vector. In practice, only a fraction of these ten parameters are identifiable. This is evident from Table 4 wherein only seven of the twenty parameters are identifiable. In general, all the dynamics parameters can be classified into three categories [7]: uniquely identifiable, identifiable in linear combinations and unidentifiable. In order to make the numerical estimation procedure robust it is imperative to categorize the parameters. A procedure to categorize the dynamics parameters, based on the knowledge of the kinematic parameters, is presented by Khosla [7].

6. Summary

In this paper, we have presented the numerical version of the identification algorithm which is suitable from computer implementation point-of-view. Using the nonlinear transformation, we reformulated the Newton-Euler inverse dynamics implementation to obtain the estimation equations. We then used the least squares estimation procedure to obtain the numerical estimates of the dynamics parameters. The identification algorithm together with the experimental implementation is an important step forward, in the area of model-based manipulator control, because it serves to satisfy the fundamental assumption that the model of the manipulator is accurately known. Based on the presented identification algorithm, we have implemented the nonlinear, feedback-based computed-torque scheme and evaluated the effect of dynamics compensation [9, 10].

References

[1] An, C. H., Atkeson, C. G. and Hollerbach, J. M.
Estimation of Inertial Parameters of Rigid Body Links of Manipulators.
In *Proceedings of the 24-th CDC.* Florida, December, 1985.

[2] Brady, M., et al. (editors).
Robot Motion: Planning and Control.
MIT Press, Cambridge, MA, 1982.

[3] Coiffet, P.
Robot Technology. Volume 2: *Interaction With the Environment.*
Prentice-Hall, NJ, 1983.

[4] Deutsch, R.
Estimation Theory.
Prentice-Hall, Englewood-Cliffs, NJ, 1965.

[5] Golub, G. H. and Van Loan, C. F.
Matrix Computations.
The Johns Hopkins University Press, Baltimore, MD , 1983.

[6] Goodwin, G. C. and Sin, K. S.
Adaptive Filtering, Prediction and Control.
Prentice-Hall, Englewood-Cliffs, NJ, 1984.

[7] Khosla, P. K.
Real-Time Control and Identification of Direct-Drive Manipulators.
PhD thesis, Department of Electrical and Computer Engineering, Carnegie-Mellon University, August , 1986.

[8] Khosla, P. K. and Kanade, T.
Parameter Identification of Robot Dynamics.
In Franklin, G. F. (editor), *Proceedings of the 24-th CDC*, pages 1754-1760. Florida, December 11-13, 1985.

[9] Khosla, P. K. and Kanade, T.
Real-Time Implementation and Evaluation of Model-Based Controls on CMU DD ARM II.
In Bejczy, A. K. (editor), *1986 IEEE International Conference on Robotics and Automation.* IEEE, April 7-10, 1986.

[10] Khosla, P. K. and Kanade, T.
Experimental Evaluation of the Feedforward Compensation and Computed-Torque Control Schemes.
In Stear, E. B. (editor), *Proceedings of the 1986 ACC.* AAAC, Seattle, WA, June 18-20, 1986.

[11] Lanczos, C.
Applied Analysis.
Prentice Hall Inc., Englewood Cliffs, N. J., 1956.

[12] Luh, J. Y. S., Walker, M. W. and Paul, R. P.
Resolved-Acceleration Control of Mechanical Manipulators.
IEEE Transactions on Automatic Control 25(3):468-474, June, 1980.

[13] Luh, J. Y. S., Walker, M. W. and Paul, R. P.
On-Line Computational Scheme for Mechanical Manipulators.
Journal of Dynamic Systems, Measurement, and Control 102(2):69-76, June, 1980.

[14] Markiewicz, B. R.
Analysis of the Computed-Torque Drive Method and Comparision with the Conventional Position Servo for a Computer-Controlled Manipulator.
Technical Memorandum 33-601, Jet Propulsion Laboratory, Pasadena, CA, March, 1973.

[15] Mukerjee, A. and Ballard, D. H.
Self-Calibration in Robot Manipulators.
In Fu, K. S. (editor), *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1050-1057. St. Louis, Missouri, March 25-28, 1985.

[16] Neuman, C. P. and Khosla, P. K.
Identification of Robot Dynamics: An Application of Recursive Estimation.
In Narendra, K. S. (editor), *Proceedings of the Fourth Yale Workshop on Applications of Adaptive Systems Theory*, pages 42-49. Yale University, New Haven, CT, May 29-31, 1985.

[17] Olsen, H. B. and Bekey, G. A.
Identification of Parameters in Models of Robots with Rotary Joints.
In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 1045-1049. IEEE, St. Louis. Missouri, March, 1985.

[18] Olsen, H. B. and Bekey, G. A.
Identification of Robot Dynamics.
In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 1004-1011. IEEE, San Francisco, CA, April, 1986.

[19] Paul, R. P.
Robot Manipulators : Mathematics, Programming and Control.
MIT Press, Cambridge, MA, 1981.

[20] Stone, H. W., Sanderson, A. and Neuman, C. P.
Arm Signature Identification.
In *Proceedings of the 1986 IEEE Conference on Robotics and Automation.* IEEE, San Francisco, CA, April, 1986.

[21] Whitney, D. and Lozinski, C. A.
Industrial Robot Calibration Methods and Results.
In *Proceedings of the International Computers in Engineering Conference.* ASME, Atlanta, GA, August, 1984.