Manipulator Dynamics Using the Extended Zero Reference Position Description

K. Kazerounian, K.C. Gupta

University of Illinois at Chicago

U. S. Army Research Office
Post Office Box 12211
Research Triangle Park, NC 27709

USARO

Approved for public release; distribution unlimited.

Robotics, Manipulator Dynamics, Zero Position, Cam Type Motions, MASP Software, Simulation

A simplified description of robotic manipulator is in terms of its zero reference position. It requires the specification of the joint axes directions and the coordinates of points locating the joint axes in the base coordinate system. This description can be learned quickly and is not prone to the errors of interpretation. It has previously been used to derive closed form inverse kinematic solutions for simple manipulators as well as to develop efficient numerical solutions for general manipulators. This paper develops manipulator dynamics in an
extended zero reference position description. The recursive Newton-Euler formulations for the problems of inverse and direct dynamics are presented in this paper.
MANIPULATOR DYNAMICS USING THE
EXTENDED ZERO REFERENCE POSITION DESCRIPTION

K. Kazemian
Assistant Professor
Department of Mechanical Engineering
University of Connecticut
Storrs, Connecticut 06269

K.C. Gupta
Professor
Department of Mechanical Engineering
University of Illinois at Chicago
Chicago, Illinois 60680

ABSTRACT

A simplified description of robotic manipulator is in terms of its zero reference position. It requires the specification of the joint axes directions and the coordinates of points locating the joint axes in the base coordinate system. This description can be learned quickly and is not prone to the errors of interpretation. It has previously been used to derive closed form inverse kinematic solutions for simple manipulators as well as to develop efficient numerical solutions for general manipulators. This paper develops manipulator dynamics in an extended zero reference position description. The recursive Newton-Euler formulations for the problems of inverse and direct dynamics are presented in this paper.

1. INTRODUCTION

The dynamics of robot arms has been considered by several investigators [1-10]. A common formulation is based upon the Denavit and Hartenberg kinematic description of spatial chains [11-13] and the use of either 4x4 matrices or 3x3 matrices along with 3x1 column vectors for kinematic and dynamic analyses [1-8]. Lagrangian [1,2,7], recursive Lagrangian [5,8], as well as recursive Newton-Euler [3,4,6] approaches have been used for dynamic formulations. Methodologies leading to explicit computations of actuator forces or torques have been considered by using dual matrices [9] and Kane's dynamical equations [10].

The aforementioned 4x4 matrices have twelve non-trivial entries. In multiplying two 4x4 matrices, multiplications with the trivial elements (zero and one) can be avoided during the programming stages, or the concept of matrix partitioning can be used to achieve some computational economy. Ignoring such possibilities, references [5,8] report that the 3x3 matrix and 3x1 vector based Lagrangian formulation is more than twice as efficient than the 4x4 matrix based Lagrangian formulation. Among other approaches, explicit methods appear to be the most efficient but these become dependent upon the configuration of the manipulator. For configuration independent formulations, the recursive Newton-Euler formulations appear to be most efficient.

An alternate kinematic description of robot arms, called the zero reference position description, has been used by Gupta [14]. It can be mastered quickly and it is not prone to the errors of interpretation by the user. Special cases such as when the adjacent joint axes become nearly parallel (e.g. VI.I due to manufacturing errors) do not create any problems in this representation. The zero reference position method has been used for closed form [14,15] as well as iterative [16] inverse kinematic robot solutions.

In this paper, which is based upon reference [17], we develop formulations for robot dynamics by using an extended zero reference position description. These formulations include: inverse dynamics which is the problem of determining actuator drive forces or torques to sustain the specified end-effector motion (section 3); and direct dynamics (or simulation) which is the problem of determining the end-effector motion resulting from the application of specified forces or torques by the actuators (section 4). In view of their relative efficiency, only the recursive Newton-Euler formulation using the zero reference position method is discussed here; reference [17] should be consulted for the details of the Lagrangian formulation.

2. ZERO REFERENCE POSITION DESCRIPTION

In the zero position description [14,15], a suitable configuration of the manipulator is designated as the zero reference position where all of the joint variables have zero values (Fig. 1). In this zero reference position the unit vectors along the revolute or prismatic joints (U0k along the kth joint, k = 1 to 6) as well as the position vector of a point on the axis of each joint (0ok, k = 1 to 6) are given in the base coordinate system. In Fig. 1, k = 1 for a typical revolute joint and k = j for a typical prismatic joint. In addition, the position vector

Figure 1. The zero reference position description for typical revolute axis i, prismatic axis j and end-effector.
Poh of a reference point h on the hand and two perpendicular unit vectors through the point h (preferably an axial and a transverse unit vector, \( u_{ok} \) and \( u_{ot} \)) are also given. All of the joint variables are set to zero at this reference position. The unit vectors \( s_k \) (k = 1 to 6), \( u_{ok} \) and \( u_{ot} \), and position vectors \( \theta_k \) (k = 1 to 6) and \( p_k \) completely define the kinematic structure of the manipulator. At a general position, the governing kinematic equations of the manipulator can be written as follows [14-17].

\[
6 \prod_{k=1}^{\infty} [D(q, s_k, u_{ok}, Q_{ok})] = [D_{P}] 
\]

The 4x4 matrix \([D_{P}]\) represents the displacement of the hand from its zero reference position to the current position. The current position of the hand is normally a part of the trajectory specification. The 4x4 matrix \([D(q_k, s_k, u_{ok}, Q_{ok})]\) represents a rotation of amount \( \theta_k \) and a translation of amount \( s_k \) with respect to the invariant line vector \((u_{ok}, Q_{ok})\), i.e., the vector \( Q_{ok} \) passing through the point \( o_k \).

In a partitioned form, this matrix can be written as follows [13].

\[
[D(q, s_k, u_{ok}, Q_{ok})] = \begin{bmatrix} \mathbf{R}(\theta_k, u_{ok}) \\ \mathbf{0} \end{bmatrix} 
\]

where

\[
\mathbf{R}(\theta_k, u_{ok}) = [1 + [u_{ok}]^2 \sin \theta_k + [u_{ok}]^2 (1 - \cos \theta_k)] 
\]

\[
[u_{ok}] = \begin{bmatrix} 0 & -u_{ok}^x & u_{ok}^y \\ u_{ok}^x & 0 & -u_{ok}^z \\ -u_{ok}^y & u_{ok}^z & 0 \end{bmatrix} 
\]

\[
\mathbf{R}(\theta_k, u_{ok}) = \begin{bmatrix} \cos \theta_k u_{ok}^x + \sin \theta_k u_{ok}^z & \cos \theta_k u_{ok}^y - \sin \theta_k u_{ok}^x & \cos \theta_k u_{ok}^z + \sin \theta_k u_{ok}^y \\ -\sin \theta_k u_{ok}^x + \cos \theta_k u_{ok}^z & \cos \theta_k u_{ok}^x + \sin \theta_k u_{ok}^z & \cos \theta_k u_{ok}^y - \sin \theta_k u_{ok}^z \\ -\sin \theta_k u_{ok}^x - \cos \theta_k u_{ok}^z & \cos \theta_k u_{ok}^x - \sin \theta_k u_{ok}^z & \cos \theta_k u_{ok}^y + \sin \theta_k u_{ok}^x \end{bmatrix} 
\]

If the kth joint is prismatic, then \( s_k = 0 \); if the kth joint is revolute, then \( q_k = 0 \).

An extension of the above description [17] for dynamic analysis is as follows. The unit vectors \( u_{ok} \) for points on the joint axes, reference body vector \( b_0, b_{k+1} \) (k = 1 to 6) are defined such that \( b_0, b_{k+1} \) is the body vector of the link \( k+1 \) and it connects the center of the kth joint to the center of the \((k+1)\)th link. The position vector of the joint center on the kth axis in the zero reference position can be computed by adding the body vectors \( b_{g2}, b_{g3}, ..., b_{gk+1} \) to the unit vectors \( u_{ok} \). The unit vectors \( u_{ok} \) and \( u_{ot} \), and the body vectors \( b_{gk} \) (k = 2 to 7) completely define the kinematic structure of the manipulator. A correspondence among the joint variables using the aforementioned zero reference position description or the common D-H description can be established easily [14,16].

For dynamic formulation, additional data concerning the dynamic properties of the manipulator is also defined in the zero reference position. Coincident points \( p_k \) and \( p_{k+1} \) are defined at the center of the kth joint such that \( p_k \) is the body point of the kth link and \( p_{k+1} \) is the body point of the \((k+1)\)th link. The following quantities are then defined at the zero position.

- \( \mathbf{e}_{ok} \) body vector of the kth link from \( p_{k-1} \) to the center of mass \( \mathbf{g}_k \)
- \( \mathbf{d}_{ok} \) body vector of the kth link from the center of mass \( \mathbf{g}_k \) to \( p_k \) (note that \( \mathbf{e}_{ok} = \mathbf{d}_{ok} + \mathbf{g}_{ok} \))
- \( w_k \) weight of the kth link
- \( \mathbf{I}_{ok} \) symmetric inertia matrix of the kth link about the translated base coordinate
- \( \mathbf{G}_k \) system through the mass center \( \mathbf{g}_k \) when the arm is at zero positions

The vectors \( u_{ok}, b_{k+1}, s_{k+1} \) and \( q_{k+1} \) and the time varying inertia matrix \( \mathbf{I}_{ok} \) at the current (non-zero) configuration are computed as follows:

\[
\{u_{ok}\} = [R_k] \{u_{ok}\} 
\]

\[
\{b_{k+1}\} = [s_k] \{b_{0, k+1}\} 
\]

\[
\{s_{k+1}\} = [s_k] \{s_{0, k+1}\} 
\]

\[
\{d_{k+1}\} = [r_k] \{d_{0, k+1}\} 
\]

\[
\mathbf{G}_k = [r_k] \mathbf{G}_{0, k+1} [r_k]^t 
\]

where \([R_k]\) is the rotation matrix of the link \( k+1 \) from its zero position to the current position. It is computed as follows:

\[
\mathbf{R}_k = \prod_{i=1}^{k} [R(\theta_i, u_{oi})] 
\]

In equation (3), the rotation matrix \([R(\theta_i, u_{oi})]\) is the principal 3x3 minor of the 4x4 displacement matrix \([D(q, s, u_{ok}, Q_{ok})]\) contained in equation (1b); it represents a rotation \( \theta_i \) about the axis \( u_{oi} \).

3. INVERSE DYNAMICS - ACTUATOR DRIVE FORCES OR TORQUES

The kth joint variable is defined as \( q_k \) such that for a revolute joint \( q_k = \theta_k \), and for a prismatic joint \( q_k = s_k \). It is assumed that the solution of the problem of the inverse kinematics is available. Thus for the specified rotation matrix of the hand \( \mathbf{R}_n \), position vector of a reference point \( h \) on the hand, velocity \( \mathbf{v}_h \) of the point \( h \), and angular velocity \( \mathbf{w}_h \) or skew symmetric matrix \([\mathbf{w}_h]\) of the hand, the corresponding joint values \( \mathbf{Q} \), \([\mathbf{Q} = (q_1, q_2, ..., q_6)]\) and joint rates \( \dot{\mathbf{Q}} \) are available.

For inverse kinematics, the following equation, which is used to compute the joint rates \( \dot{\mathbf{Q}} \), is differentiated.
In equation (4), \( [J] \) is the 6x6 velocity Jacobian matrix of the manipulator and its elements are computed by using the matric notation. Equation (4a) is differentiated to obtain equation (4b).

\[
[J] (\dot{q}) + [J] (\ddot{q}) = \left\{ \begin{array}{c}
\dot{q}h \\
\ddot{q}h 
\end{array} \right\}
\] (4b)

where \( \dot{q}_h \) and \( \ddot{q}_h \) are linear and angular accelerations of the hand and \( [J] \) is a 6x6 matrix whose elements are the derivatives of the elements of the Jacobian matrix \([J]\). A recursive method to compute \( [J] \) is discussed in reference [17]. Equation (4b) is then solved for the joint accelerations \( \ddot{q} \).

When the velocity and acceleration of the hand as well as the corresponding data at the joint level (i.e. \( Q, \dot{Q}, \ddot{Q} \)) are known, the dynamics of the manipulator can be formulated as follows. A recursive process for the computation of the actuator forces (or torques) starts from the 7th link. At each step, the angular velocity and acceleration of a particular link, and the linear acceleration of its mass center are computed. These values are then used to compute the inertia force and inertia moment acting on the link. The joint forces and torques are then computed by writing the dynamical equilibrium equations (D'Alembert's principle) for that link.

In particular, when the recursive process is at the kth link \((k = 7, 6, 5, \ldots, 2)\), the computation is as follows

\[
\omega_k = \begin{cases} 
\omega_k^{k-1} - \dot{q}_k u_k & \text{if the kth joint is revolute} \\
\omega_k^{k+1} & \text{if the kth joint is prismatic}
\end{cases}
\]

Equation (5) is differentiated to compute the angular acceleration \( \ddot{q}_k \).

\[
\ddot{q}_k = \begin{cases} 
\dot{q}_k - \ddot{q}_k u_k - \ddot{q}_k \omega_k \times u_k & \text{if the kth joint is revolute} \\
\ddot{q}_k^{k+1} & \text{if the kth joint is prismatic}
\end{cases}
\]

The next step is the computation of the acceleration of the mass center of the kth link. Before that, however, two vectors \( p_{k-1}^k G \) and \( G_p k^{-1} \) (Fig. 2) are defined as follows.

![Diagram of two adjacent links k and k+1](image)

Figure 2. Two adjacent links k and k+1.

\[
G_{k+1} = \frac{s_k}{c_k} \text{ if the kth joint is revolute}
\]

\[
G_{k+1} = \frac{d_k}{c_k} \text{ if the kth joint is prismatic and the collar is on the kth link}
\]

\[
G_{k+1} = \frac{d_k + q_k u_k}{c_k} \text{ if the kth joint is prismatic and the collar is on the (k+1)th link}
\]

It should be noted that the actuator forces (or torques) do not depend on which link carries the collar of the prismatic joint. The joint reactions, however, are affected by this fact.

The acceleration of the mass center \( \ddot{G}_k \) is now obtained from the acceleration of the mass center \( \ddot{G}_{k+1} \) by using the relation for two points in the same body (equations 9, 11) and also for coincident points belonging to different bodies (equation 10). For prismatic joints, note the Coriolis acceleration term in equation 10.

\[
s_{k+1} = s_k - \frac{u_k}{p_{k+1}^k G_{k+1}} \quad \text{if the kth joint is revolute}
\]

\[
s_k = \begin{cases} 
\ddot{q}_k u_k & \text{if the kth joint is revolute} \\
- \ddot{q}_k \omega_k \times u_k & \text{if the kth joint is prismatic}
\end{cases}
\]

At this point the linear acceleration of the mass center \( \ddot{G}_k \) as well as the angular velocity and acceleration of the kth link are known and the inertia force and moment acting on this link are computed in the base coordinate system as follows (see Appendix).

\[
\ddot{G}_k = \frac{s_k}{c_k} \quad \text{if the kth joint is revolute}
\]

\[
\ddot{G}_k = \frac{d_k}{c_k} \text{ if the kth joint is prismatic}
\]

\[
\ddot{G}_k = \frac{d_k + q_k u_k}{c_k} \text{ if the kth joint is prismatic}
\]

where \( s_k \) is the inertia force acting on the kth link and \( c_k \) is the mass of the kth link.

\[
[H] = -[G] \frac{[G]}{[G]} (\ddot{G}_k) - [G] \frac{[G]}{[G]} (\ddot{G}_k)
\]
Figure 3 shows the kth link with all reactive, gravitational, and inertia forces and moments acting on the link.

Equations (5-16) constitute a recursive set of relations for computing all of the actuator forces or torques. This formulation requires that body vectors \( \mathbf{b}_k \), \( \mathbf{b}_k \), \( \mathbf{c}_k \), \( \mathbf{d}_k \) as well as the time varying inertia matrix \( [I^k] \) be computed at the current position (eq. (2)). This process also requires the computation of the rotation matrix \( [R_k] \) in equation (3). Computation of these quantities in equations (2) and (3) involves a large number of arithmetic operations. The algorithm is relieved from these excessive computations as follows. Let us define superscripted vectors

\[
\mathbf{G}^* = \mathbf{G}^* - \mathbf{G}^* - \mathbf{G}^* - \mathbf{G}^* - \mathbf{G}^* - \mathbf{G}^* - \mathbf{G}^* - \mathbf{G}^*
\]

by premultiplying the corresponding vectors

\[
\mathbf{G}_k, \mathbf{G}_k, \mathbf{G}_k, \mathbf{G}_k, \mathbf{G}_k, \mathbf{G}_k, \mathbf{G}_k, \mathbf{G}_k
\]

by the 3x3 rotation matrix \( [R_{k-1}]^t \) defined by equation (3). Equation (3) is also rearranged as follows

\[
[R_{k-1}]^t = [R(b_k, u_{ok})] [R_k]^t
\]

In light of the above definitions, all of the equations (5-16) are premultiplied by the matrix \( [R_{k-1}] \) as follows

\[
\mathbf{G}_k = [R(b_{k-1}, u_{ok})] \mathbf{G}_k + u_{ok} X \mathbf{u}_{ok}
\]

By premultiplying the corresponding vectors

\[
\mathbf{G}_k, \mathbf{G}_k, \mathbf{G}_k, \mathbf{G}_k, \mathbf{G}_k, \mathbf{G}_k, \mathbf{G}_k, \mathbf{G}_k
\]

By premultiplying the corresponding vectors

\[
\mathbf{G}_k, \mathbf{G}_k, \mathbf{G}_k, \mathbf{G}_k, \mathbf{G}_k, \mathbf{G}_k, \mathbf{G}_k, \mathbf{G}_k
\]
eq (16): \[
\begin{align*}
T_{k-1} &= T_{k-1} - u_{o,k-1} \\
N_{k-1} &= N_{k-1} - u_{o,k-1}
\end{align*}
\] (29)

Equations (18-29) are then used instead of equations (5-16) for the recursive computation of actuator forces (or torques). In modified equations the computation of the body vectors and inertia matrices of links at their current position is not required; only the vectors and inertia matrices defined in the zero reference position are used.  

The efficiency of the above formulation is directly related to the total number of arithmetic (i.e. multiplications m, additions a) and trigonometric (t) operations. The formulation presented in this section has the following computational complexity.

\[
r = (136 m + 118 a + 2t) + p(139 m + 118 a + 2t)
\]

where \( r \) is the number of revolute joints and \( p \) is the number of prismatic joints. For a 6-R manipulator, the computational complexity equals 816 m + 708 a + 12 t. As a comparison, the Newton-Euler formulation of the reference \([4]\) requires 851 m + 739 a + 12 t computations.

In a numerical example, the above inverse dynamic formulation is used to compute the actuator torques which maintain a specified trajectory for a 6-R manipulator (Fig. 4). Tables 1 and 2 contain kinematic and dynamic description of this manipulator in its zero reference position shown in Fig. 4.

Figure 4. The zero reference position of an industrial manipulator (Table 1).

The trajectory of the hand is specified as follows. The point \( h \) of the hand is to move on a circle of radius 6" in a plane parallel to the base YZ plane and its center located on the base X axis at \( x = 34" \). The axial unit vector \( u_{h} \) makes an angle of \( \frac{\pi}{3} \) radians with the intersecting X axis, while the transverse unit vector \( u_{t} \) remains tangent to the above-mentioned circle. Therefore

\[
\begin{align*}
X_{h} &= 34 \\
Y_{h} &= 6 \sin \psi \\
Z_{h} &= 6 \cos \psi \\
u_{a} &= \begin{pmatrix}
\cos \frac{\pi}{3} \\
-\sin \frac{\pi}{3} \\
\sin \frac{\pi}{3}
\end{pmatrix}
\end{align*}
\] (30)

Angle \( \psi \) changes from 0 to 2\( \pi \), where \( \psi = 0 \) indicates the top point on the circle. For a smooth start and stop of the hand on the trajectory, the \( \psi(t) \) variation is selected according to the following cam type scheme \([22-23]\). The total time \( T \) for the trajectory is divided into a half-cycloidal start \( (C1) \) of duration \( T_{1} \), a constant velocity segment \( (T_{2}) \) and a half-cycloidal stop \( (C2) \) of duration \( T_{3} \). The transfer \( \psi \) are determined to insure continuity of \( \psi \), while \( \dot{\psi} \) is already continuous. This leads to the start segment angle change \( \psi_1 = 2\pi T_{1}/(T_{1} + T_{2}) \), constant velocity segment angle change \( \psi_2 = 4\pi T_{2}/(T_{1} + T_{2}) \) and stop segment angle change \( \psi_3 = 2\pi T_{3}/(T_{1} + T_{2}) \). For \( T_{1} = T_{2} = T_{3} = 3 \) sec., the following equations for \( \psi, \dot{\psi}, \ddot{\psi} \) are obtained.

\[
\begin{align*}
0 &< t < 3 \\
\dot{\psi} &= \frac{\pi}{18} \sin \frac{\pi t}{3} \\
\ddot{\psi} &= \frac{\pi}{6} - \frac{\pi}{6} \cos \frac{\pi t}{3} \\
\dddot{\psi} &= \frac{\pi}{6} - \frac{\pi}{2} \sin \frac{\pi t}{3}
\end{align*}
\] (32)

\[
\begin{align*}
3 &< t < 6 \\
\dot{\psi} &= 0 \\
\ddot{\psi} &= \frac{\pi}{3} \\
\dddot{\psi} &= \frac{\pi}{2} + \frac{\pi}{3} (t - 3)
\end{align*}
\] (33)

\[
\begin{align*}
6 &< t < 9 \\
\dot{\psi} &= \frac{\pi}{18} \sin \frac{\pi}{3} (9 - t) \\
\ddot{\psi} &= \frac{\pi}{6} - \frac{\pi}{6} \cos \frac{\pi}{3} (9 - t) \\
\dddot{\psi} &= \frac{\pi}{2} + \frac{\pi}{6} (t - 6) + \frac{\pi}{2} \sin \frac{\pi}{3} (9 - t)
\end{align*}
\] (34)

Then the velocity and acceleration specifications of the hand are as follows

\[
\begin{align*}
\omega_{h} &= \begin{pmatrix}
\dot{\psi} \\
0
\end{pmatrix}
\end{align*}
\] (35)
Table 1 Link and joint information for a PUMA type manipulator in zero-position

<table>
<thead>
<tr>
<th>k</th>
<th>Type of the kth joint</th>
<th>$\bar{u}_{0,k}$</th>
<th>$\bar{c}_{0,k+1}$</th>
<th>$\bar{a}_{0,k+1}$</th>
<th>$\bar{b}_{0,k+1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R</td>
<td>(0,0,1)</td>
<td>(0,5,0)</td>
<td>(0,5,0)</td>
<td>(0,10,0)</td>
</tr>
<tr>
<td>2</td>
<td>R</td>
<td>(0,1,0)</td>
<td>(0,2,0)</td>
<td>(9,-2,0)</td>
<td>(17,0,0)</td>
</tr>
<tr>
<td>3</td>
<td>R</td>
<td>(0,1,0)</td>
<td>(0,0,-9)</td>
<td>(0,0,-8)</td>
<td>(0,0,-17)</td>
</tr>
<tr>
<td>4</td>
<td>R</td>
<td>(0,0,1)</td>
<td>(0,0,-1)</td>
<td>(0,0,1)</td>
<td>(0,0,0)</td>
</tr>
<tr>
<td>5</td>
<td>R</td>
<td>(0,-1,0)</td>
<td>(0,-1,0)</td>
<td>(0,1,0)</td>
<td>(0,0,0)</td>
</tr>
<tr>
<td>6</td>
<td>R</td>
<td>(0,0,-1)</td>
<td>(0,0,-3)</td>
<td>(0,0,-2)</td>
<td>(0,0,-5)</td>
</tr>
</tbody>
</table>

$\bar{u}_c = (0,-1,0)$

$\bar{u}_a = (0,0,-1)$

Table 2 Mass and inertia information for the manipulator of Table 1 in zero-position

<table>
<thead>
<tr>
<th>k</th>
<th>$W_k$ (lb)</th>
<th>$(I_{xx})_k$ (lb-in-sec)$^2$</th>
<th>$(I_{yy})_k$</th>
<th>$(I_{zz})_k$</th>
<th>$(I_{xy})_k$</th>
<th>$(I_{xz})_k$</th>
<th>$(I_{yz})_k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>10</td>
<td>0.230</td>
<td>0.005</td>
<td>0.230</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>0.069</td>
<td>1.453</td>
<td>1.394</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>1.405</td>
<td>1.585</td>
<td>0.034</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0.001</td>
<td>0.001</td>
<td>0.0001</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0.001</td>
<td>0.0001</td>
<td>0.001</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>0.069</td>
<td>0.069</td>
<td>0.01</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

After solving the inverse kinematics problem for the joint $\theta_1, \theta_2, \theta_3$, the joint actuator forces or torques are computed by using equation (18-22). Figures (5-7) show the variations of $Q, \dot{Q}$, and $\dot{Q}$ along $V_1.6$ the trajectory ($0 \leq t < 9$ sec). In Figures 5a and 5b, the joint variable $q_4$ (i.e. $\theta_4$) makes a complete rotation (2π), while the other joints ($\theta_1 = \theta_2, \theta_5 = \theta_6$) return to their starting values. The joint variable $q_3$ (\(\theta_3\)) experiences very small changes using the execution of this particular trajectory. The computed drive torques for the joint actuators are plotted in Figures 5a and 5b. The joint actuators 2 and 3 are most affected by the gravitational loading. The values of the joint torques at the beginning and the end of the trajectory correspond to their static equilibrium values.

In the various numerical examples, it was observed that the inverse dynamic computations of the joint drive torques took approximately 0.003 CPU seconds per set when these were programmed in double precision Fortran on an IBM 3081.

The details of the inverse dynamics in the zero reference position representation by using the Lagrangian formulation are presented in reference [17]. Although these are too involved to present here, the development is analogous to that in reference [8].
Figure 5(a)

Figure 5(b)

Figure 6(a)

Figure 6(b)
Figure 7(a)

Figure 7(b)

Figure 8(a)

Figure 8(b)
A. DIRECT DYNAMICS (SIMULATION)

Direct dynamics or simulation is the problem of determining the position, velocity and acceleration of the hand when the values of the joint actuator forces (or torques) are known as functions of time. It is now discussed in the context of the zero reference position description.

In general the equations of motion for a 6 D.O.F. manipulator can be written as

\[ \mathbf{H}(\mathbf{q}) \dot{\mathbf{q}} + \left( \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \right) \dot{\mathbf{q}} + \left( \mathbf{g}(\mathbf{q}) \right) = \mathbf{f} \]  

(39)

where

\( \mathbf{H}(\mathbf{q}) = 6 \times 6 \) non-singular, symmetric inertia matrix

\( \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = 6 \times 1 \) vector containing "centripetal" and "Coriolis" effects

\( \mathbf{g}(\mathbf{q}) = 6 \times 1 \) vector containing gravity and end-effector loading effects

\( \mathbf{f} = 6 \times 1 \) vector of actuator forces (or torques)

In the problem of direct dynamics, the joint forces \( \mathbf{f} \) are known. Also, at the current integration step, the manipulator state variables \( \mathbf{q} \) and \( \dot{\mathbf{q}} \) are available. The linear system in equation (39) is solved for \( \ddot{\mathbf{q}} \). The joint accelerations \( \ddot{\mathbf{q}} \) are then integrated numerically to compute the next state variables \( \dot{\mathbf{q}} \) and \( \mathbf{q} \). The Newton-Euler formulation is utilized to define the matrix \( \mathbf{H} \) and vectors \( \mathbf{C} \) and \( \mathbf{g} \) in equation (39). The formulation to define these equations is similar to that of the problem of inverse dynamics in section 3, except for the following modifications.

1. Since the position, velocity and acceleration of the hand are not known, the recursive process to compute the angular velocities and accelerations of the links and the linear accelerations of their mass centers starts from the base link (1st link) to the hand (7th link).

2. The joint accelerations \( \ddot{\mathbf{q}} \) are not known. Therefore the terms which are affected by the joint accelerations are defined as linear functions of \( \mathbf{q} \). These terms for the \( k \)th link are: \( \mathbf{C}_{\mathbf{q}k}, \mathbf{C}_{\dot{\mathbf{q}}k}, \mathbf{F}_{\mathbf{q}k}, \mathbf{M}_{\mathbf{k}}, \mathbf{R}_{\mathbf{k}}, \mathbf{T}_{\mathbf{k}} \) and the derived actuator force (torque) \( \mathbf{f}_k \) (29). In other words these vectors in data storage have the dimensions of \( 3 \times 7 \) (\( \mathbf{f}_k \) has the dimensions \( 1 \times 7 \)). The columns 1-6 represent the coefficients of \( \dot{\mathbf{q}}_1 \) to \( \dot{\mathbf{q}}_6 \), and the seventh column represents the constant terms.

When the above modifications are incorporated in equations (18-27), the equations (18-29, 22-24) are rearranged as follows to change the recursion.

\[ \mathbf{\omega}_{k+1} = \begin{cases} \mathbf{R}^t(\mathbf{q}_k \omega_{ok}) \mathbf{\omega}_k & \text{if the kth joint is revolute} \\ \mathbf{R}^t(\mathbf{q}_k \omega_{ok}) \mathbf{\omega}_k & \text{if the kth joint is prismatic} \end{cases} \]  

(40)

\[ \mathbf{\omega}_{k+1} = \begin{cases} \mathbf{R}^t(\mathbf{q}_k \omega_{ok}) \left[ \mathbf{\omega}_k + \hat{\mathbf{q}}_{k-1} \times \mathbf{u}_{ok} \right] & \text{if \( k \neq 1 \)} \\ \mathbf{R}^t(\mathbf{q}_k \omega_{ok}) \mathbf{\omega}_k & \text{if \( k = 1 \)} \end{cases} \]  

(41)

These equations (40-41, 20-21, 42-44) are used for \( k \) from 2 to 7 to compute \( \mathbf{\omega}_1, \mathbf{\omega}_2 \) and \( \mathbf{\omega}_k \) for all of the links. Equations (25-28) are then used to compute the six joint actuator forces (or torques) \( \mathbf{f}_k \) as linear functions of \( \mathbf{q}_k \). The system of linear equations (39) is then completely defined by using the known values of the actuator forces on the right hand side. On the left hand side on this equation, there is a \( 6 \times 7 \) matrix (the \( k \)th row is \( \mathbf{f}_k \)) where the first six columns correspond to the matrix \( \mathbf{H} \) and the seventh column corresponds to the vectors \( \mathbf{C} \) and \( \mathbf{g} \) in equation (39). To reduce the number of arithmetic operations needed to define and solve this system of equations, the following observations were utilized.

1. \( \mathbf{C}_{\mathbf{q}k}, \mathbf{C}_{\dot{\mathbf{q}}k}, \mathbf{P}_k \) and \( \mathbf{M}_k \) are functions of \( \mathbf{q}_i \), \( i = 1, 2, \ldots, k-1 \). Therefore the columns \( k \), \( k+1 \), \( \ldots \), 6 of the corresponding \( 3 \times 7 \) arrays are null columns. The vector and scalar operations on these null columns are avoided.

2. The inertia matrix \( [\mathbf{H}]_{6 \times 6} \) is a symmetric matrix and therefore only the lower triangular part of this matrix needs to be computed. In actual implementation this means that vector and scalar operations on columns \( k, k+1, \ldots, 6 \) of all \( 3 \times 7 \) arrays in equations (25-28) are avoided.

3. Since \( [\mathbf{H}] \) is a symmetric matrix, an efficient method such as triangular decomposition (24) can be used to solve the system of equations (39).

The formulation discussed above defines 6 second order, ordinary differential equations as follows

\[ \ddot{\mathbf{q}} = f(\mathbf{q}, \dot{\mathbf{q}}, t) \]  

(44)

Simulation is basically the numerical solution of the initial value problem involving 6 second order differential equations (44). In terms of the state variables \( (\mathbf{q}, \dot{\mathbf{q}}) \) this system becomes a system of 12 first order differential equations. An efficient predictor-corrector integration scheme is used in the computer program to compute the state variables \( \dot{\mathbf{q}} \) and \( \ddot{\mathbf{q}} \) (17).

The number of operations required in the simulation process prior to the integration step equals 2468 multiplications, 1879 additions and 12 trigonometric evaluations. In the numerical examples presented, it was observed that each cycle of simulation including the integration step, took approximately 0.018 CPU on an IBM 3081 using the double precision Fortran.

Based upon the zero reference position description of robot arms, the inverse kinematics, inverse dynamics and direct dynamics (simulation) have been incorporated into a general purpose FORTRAN computer.
program MASP - Manipulator Analysis and Simulation Package which is listed in reference [17].

5. CONCLUSION

The inverse kinematics of manipulators by using the zero reference position method has been discussed in references [14-16]. In this work the zero reference position analysis method has been extended to formulate the problems of dynamics for general manipulators. A computationally efficient formulation for inverse dynamics based upon recursive Newton-Euler equations has been developed. This formulation is then rearranged and modified to solve the problem of direct dynamics (simulation) for general industrial robots.

6. ACKNOWLEDGEMENTS

The financial support of this research under the U.S. Army Research Office Contract 20125 EC/DAAG29-83-K0144 is gratefully acknowledged.

7. REFERENCES


B. APPENDIX

Consider the generalized Euler equations in a centroidal body system [18-21].

\[
\begin{align*}
\{M_k\}_{tb} &= \left[ R \right] \{M_k\}_{tb} \\
\{C_k\}_{tb} &= \left[ R \right] \{C_k\}_{tb}
\end{align*}
\]

Let \([R]\) be the rotation matrix which relates the coordinates in the body system (subscript \(b\)) to those in the translated base coordinate system (subscript \(tb\)) located at the center of mass \(C\).

\[
\begin{align*}
\{M_k\}_{tb} &= \left[ R \right] \{M_k\}_{tb} \\
\{C_k\}_{tb} &= \left[ R \right] \{C_k\}_{tb}
\end{align*}
\]
\[ \{ \dot{\omega}_k \}_{tb} = - \{ \Omega_k \}_{tb} \{ \Omega^C_k(t) \}_{tb} \{ \omega_k \}_{tb} - \{ \Omega^C_k(t) \}_{tb} \{ \omega_k \}_{tb} \]

or simply, in the translated base coordinate system

\[ \{ \dot{\omega}_k \} = - \{ \Omega_k \} \{ \omega^C_k(t) \} \{ \omega_k \} - \{ \omega^C_k(t) \} \{ \omega_k \} \]

Although the forms of the generalized Euler's equations are similar in the body system and the translated base coordinate system [18], the inertia matrix is time invariant in the former while it is a function of time in the latter.
MANIPULATOR DYNAMICS USING THE EXTENDED ZERO REFERENCE POSITION DESCRIPTION
by K. Kazerounian, K. C. Gupta

K. KAZEROUMAN
Dr. Kazem Kazerounian started his college education in Iran. He later continued and received his BS degree in Mechanical Analysis and Design from University of Illinois at Chicago in the fields of Design, Mechanisms and Robotics. Currently he is an assistant professor of Mechanical Engineering at the University of Connecticut.

K. C. GUPTA
MS '71 (Case), PhD '74 (Stanford). At University of Illinois at Chicago since 1974; Assistant Professor 1974-79; Associate Professor 1979-84; Professor 1984--; Director of Graduate Studies 1982-84. Associate Editor, ASME Journal of Mechanical Design 1981-82; Papers Review Chairman (North America), 1982 ASME Mechanisms Conference; Editorial Advisory Board, ASME Applied Mechanics Review. Member, ASME Mechanisms Committee 1981-86; Member, ASME Mechanisms Subcommittee on Honors, 1982; Chairman ASME Mechanisms Subcommittee on Robots and Manipulators, 1984-86. Member ASME. Merit Scholarship 1964-69; Best Paper Award and Proctor and Gamble Award of Merit, 1978 ASME Mechanisms Conference; Henry Hess Award (for technical literature), ASME 1979. Listings in Who's Who in the Midwest; Who's Who in Engineering. Approximately 25 journal papers and 25 conference papers in mechanism design, robotics and design optimization. Phone (312)996-3427 or 5317.

Presented at the 9th Applied Mechanisms Conference
ANNOUNCING
THE
9th APPLIED MECHANISMS
CONFERENCE

A University Extension Program of the College of
Engineering, Architecture and Technology of
Oklahoma State University

October 28, 29, & 30, 1985 Kansas City, Missouri

VOLUME TWO

Sponsored By
OSU Engineering Extension
and the School of Mechanical and Aerospace Engineering

Oklahoma State University