This is a repository copy of *The Dynamic Performance of Robot Manipulators Under Different Operating Conditions*.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/78142/

**Monograph:**

---

**Reuse**
Unless indicated otherwise, fulltext items are protected by copyright with all rights reserved. The copyright exception in section 29 of the Copyright, Designs and Patents Act 1988 allows the making of a single copy solely for the purpose of non-commercial research or private study within the limits of fair dealing. The publisher or other rights-holder may allow further reproduction and re-use of this version - refer to the White Rose Research Online record for this item. Where records identify the publisher as the copyright holder, users can verify any specific terms of use on the publisher’s website.

**Takedown**
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.
The Dynamic Performance of Robot Manipulators
Under Different Operating Conditions

A. Y. ZOMAYA and A. S. MORRIS

Department of Control Engineering
University of Sheffield
Mappin Street
Sheffield S1 3JD

Research Report No. 345

October 1988
Abstract:

The dynamical performance of robot manipulators is greatly affected by the different payloads handled by the end effector (hand). Hence, it is very important, especially for industrial applications, to study the different interconnected relationships between the manipulator joints, speeds, loads, and actuation forces. In this paper, a simplified symbolic Lagrangian representation of the different terms is presented, with emphasis on the coriolis and centripetal effects. The accuracy and computational efficiency of this new formulation is demonstrated by simulation of a Stanford and PUMA 560 manipulator. Useful quantitative measurements and error analysis are also included on the significance of coriolis and centripetal terms under different load and speed conditions.

1. Introduction:

The mathematical formulation of the equation of motion of a robot manipulator divides into the two separate areas of :

(a) the inverse dynamics, which are concerned with finding the forces required to drive the arm through some specified trajectory.

(b) the forward dynamics, which are concerned with calculating the position, velocity and acceleration of each link for a given set of applied forces.

The dynamics consists of a set of differential, coupled, non linear, and matrix oriented representation which describes the behaviour of the robot arm.

Various robot dynamic formulations have been proposed during the past few years. The Lagrange-Euler (LE) [1, 2, 3] has low computational efficiency but a very well structured and systematic representation that allows for different control applications. The Recursive Lagrangian [4] has better computation time but destroys the structure of the equations. The Generalized D' Alambert formulation [5] has a fair representation with some computation improvements. The Newton-Euler (NE) [6, 7, 8, 9] has a very efficient computational representation with very untidy recursive equations. Tabulation dependent schemes [12] have very serious difficulties owing to the enormous computer memory storage requirements. Other approaches include the dynamic equations of Kane [13] and the use of parallel processing and advanced computer architectures to reduce the computation-time [14, 15, 16, 17]. Of the previous methods, the most commonly used are the (LE) and (NE). The interaction and equivalence between these schemes has been shown by Silver [11] and Turney et.al. [18].
In this paper, a simplified symbolic mathematical description of the dynamics based on the (LE) will be presented accompanied by execution-time results. Also for the first time, exact quantification is given of what effect the inertial parameters of coriolis and centripetal forces actually have as the load and speed are varied. This allows for a much clearer understanding in any particular circumstance of the limits of operating speed at which the value of the dynamic forces are fairly valid. The analysis is based on a set of compiled data of two robot arms (stanford, PUMA 560) to facilitate the robot dynamic performance problem.

2. The Lagrangian:

The importance and usefulness of the (LE) rise from its simple, algorithmic and highly structured equation based on lagrangian mechanics which is derived from energetic principles.

The set of equations can be written in a compact form which is the final outcome from solving (LE):

\[ F = D(v) \ddot{v} + C(v) \dot{v} + G(v) \quad (1) \]

Where,

\[ D : n \times n \text{ matrix which represent the coupling and effective inertia terms, position and acceleration dependent.} \]

\[ C : n \times n \times n \text{ matrix which represent the centripetal and coriolis terms, position and velocity dependent.} \]

\[ G : n \text{ - dimensional vector representing the gravity loading effects, position dependent.} \]

\[ v, \dot{v}, \ddot{v} : \text{position, velocity and acceleration.} \]

\[ n : \text{degree of freedom.} \]

\[ F : \text{force (prismatic joint) or torque (revolute joint).} \]

The very general form of eq.(1) is very important in state space and modern control applications [19, 20, 21]. Because of the changing geometrical configuration of the robot as it moves, the inertial parameters of the robot are time-varying. The inertial and gravity terms affect the servo stability and positioning accuracy of the arm. The coriolis and centripetal terms contribute little to the dynamic forces at low operational speeds but become highly significant at high speeds. Normal practice in industrial robots is therefore to limit their operational speed such that this problem is not encountered.

The use of (LE) is very useful but cannot be utilised in real-time control without further simplifications. Because of that, many attempts had been made to reduce the order of computations [3, 22, 23, 24, 25, 26, 27]. The nomenclature used in [1, 3] which is based on the Denavit-Hartenberg
conventions |10| will be used in our discussion, Eq. (1) might be written in an alternate form:

\[
F_i = \sum_{j=1}^{n} P_{ij} \dot{q}_j + \sum_{j=1}^{n} \sum_{k=1}^{n} P_{ijk} \dot{q}_j \dot{q}_k + P_i
\]  

(2)

where

- \(P_{ii}\), effective inertia at joint (i)
- \(P_{ij}\), coupling inertia between joint (i) and (j)
- \(P_{ij} = \sum_{\ell=\max(i,j)}^{n} \frac{\text{tr} \left( \frac{\partial H_{\ell}}{\partial q_j} \right)}{\frac{\partial q_i}{\partial q_\ell}} \) 

(3)

- \(P_{ij}\), Coriolis forces at joint (i) due to velocity at joint (j)
- \(P_{ijk}\), Coriolis forces at joint (i) due to velocities at joint (i) and (k).

\[
P_{ijk} = \sum_{\ell=\max(i,j,k)}^{n} \frac{\text{tr} \left( \frac{\partial^2 H_{\ell}}{\partial q_j \partial q_k} \right)}{\frac{\partial q_i}{\partial q_\ell}} \left( \frac{\partial H_{\ell}}{\partial q_i} \right)
\]  

(4)

- \(P_i\), gravity loading vector

\[
P_i = \sum_{\ell=1}^{n} -m_{\ell} \mathbf{g}^T \left( \frac{\partial H_{\ell}}{\partial q_i} \right) \left( \frac{\partial \mathbf{r}_{\ell}}{\partial q_i} \right)
\]  

(5)

- \(m_{\ell}\), mass of link \(\ell\)
- \(\mathbf{r}_{\ell}\), centre of mass of link \(\ell\) according to its own coordinate.

\(\mathbf{H}_{\ell}\), 4 x 4 link transformation or denavit-hartenberg matrices.

\(\mathbf{g}\), gravitational effects vector.

and \(\mathbf{J}\) is a pseudo inertia matrix, which is luckily for most industrial manipulator, and in our case the stanford and PUMA 560, is of the form,

\[
\mathbf{J}_{ij} = \begin{bmatrix}
-I_{xx\ell} + I_{yy\ell} + I_{zz\ell} & 0 & 0 & m_{\ell} x_{\ell} \\
2 & I_{xx\ell} - I_{yy\ell} + I_{zz\ell} & 0 & m_{\ell} y_{\ell} \\
0 & 0 & I_{xx\ell} + I_{yy\ell} - I_{zz\ell} & m_{\ell} z_{\ell} \\
m_{\ell} x_{\ell} & m_{\ell} y_{\ell} & m_{\ell} z_{\ell} & m_{\ell}
\end{bmatrix}
\]
3. Simplification of the Lagrangian Formulation:

3.1 Inertial and Gravity Terms:

The effective and coupling inertia terms of eq. (3) have been shown in [1] to be:

$$P_{ij} = \sum_{k=\max(i,j)}^{n} \text{tr} \left( \dot{\delta}_{ij} \dot{\delta}_{ik} \right)$$

(7)

Where \( \dot{\delta}_{ij} \) is the differential translation and rotation transformation matrix of joint \( k \) with respect to the \( i \)th joint coordinate given by

$$\dot{\delta}_{ij} = \begin{bmatrix}
0 & -\delta_{ix} & \delta_{iy} & \delta_{iz} \\
\delta_{ix} & 0 & \delta_{ix} & \delta_{iy} \\
-\delta_{iy} & \delta_{ix} & 0 & \delta_{iz} \\
0 & 0 & 0 & 0
\end{bmatrix}$$

(8)

Where \( \delta_{ij} = \begin{bmatrix}
-i_{kx} & i_{ky} & i_{kz} \\
-i_{kx} & i_{ky} & i_{kz} \\
-i_{kx} & i_{ky} & i_{kz}
\end{bmatrix} \)

revolute joint

prismatic joint

prismatic joint

By expanding eq. (7) to reduce the multiplication by zero operations and to give insight into more customization of the dynamics [27] which depends mainly on the arm architecture to lead to more simplifications.

Assume a matrix \( E \) such that:

$$E = \begin{bmatrix}
e_{ij} & 0 \\
0 & e
\end{bmatrix}, e: 3 \times 3 \text{ matrix}$$

using the trace operator will give:

$$P_{ij} = \sum_{k=\max(i,j)}^{n} \sum_{m=1}^{3} e_{mm}$$

(9)

Where \( e_{mm} \) is given as,

$$-4-$$
\[
\begin{align*}
= & \sum_{j=1}^{3} \left[ \begin{array}{c} \delta_{iy} \\ \delta_{iz} \\ \delta_{jy} \\ \delta_{jz} \\ \delta_{ix} \\ \delta_{jx} \end{array} \right] \cdot \left[ \begin{array}{c} \delta_{iy} \\ \delta_{iz} \\ \delta_{jy} \\ \delta_{jz} \\ \delta_{ix} \\ \delta_{jx} \end{array} \right] + 5 \sum_{j=2}^{3} \left[ \begin{array}{c} \delta_{ix} \\ \delta_{iz} \\ \delta_{jx} \\ \delta_{jz} \\ \delta_{ix} \\ \delta_{jx} \end{array} \right] \\
& + \sum_{j=4}^{6} \left[ \begin{array}{c} \delta_{ix} \\ \delta_{iy} \\ \delta_{iz} \\ \delta_{jz} \\ \delta_{ix} \\ \delta_{jx} \end{array} \right] + \sum_{j=4}^{6} \left[ \begin{array}{c} \delta_{ix} \\ \delta_{iy} \\ \delta_{iz} \\ \delta_{jz} \\ \delta_{ix} \\ \delta_{jx} \end{array} \right] + \sum_{j=4}^{6} \left[ \begin{array}{c} \delta_{ix} \\ \delta_{iy} \\ \delta_{iz} \\ \delta_{jz} \\ \delta_{ix} \\ \delta_{jx} \end{array} \right] \\
& + \sum_{j=4}^{6} \left[ \begin{array}{c} \delta_{ix} \\ \delta_{iy} \\ \delta_{iz} \\ \delta_{jz} \\ \delta_{ix} \\ \delta_{jx} \end{array} \right] + \sum_{j=4}^{6} \left[ \begin{array}{c} \delta_{ix} \\ \delta_{iy} \\ \delta_{iz} \\ \delta_{jz} \\ \delta_{ix} \\ \delta_{jx} \end{array} \right] + \sum_{j=4}^{6} \left[ \begin{array}{c} \delta_{ix} \\ \delta_{iy} \\ \delta_{iz} \\ \delta_{jz} \\ \delta_{ix} \\ \delta_{jx} \end{array} \right]
\end{align*}
\]

where \(|A|\): determinant of \(A\), \(\cdot\): scalar multiplication.

The gravity loading vector is given in [1] to be:

\[
P_i = \sum_{k=1}^{n} m_{ik} \delta_{jx} \quad (10)
\]

where

\[
\frac{i-1}{g} = \left\{ \begin{array}{cccc}
g.o & g.n & o & o \\
o & o & o & -g.a
\end{array} \right\} \text{ rotational joint}
\]

For a 6-dof arm at a given set of position variables, 36 elements of the matrix \([P_{ij}]\) should be evaluated, but due to symmetry the number reduces to 21 elements. For the gravity loading vector 6 elements should be calculated [1,3,26].

### 3.2 Coriolis and Centripetal Effects:

These terms have great importance in high speed operations which is the case in many industrial applications. Eq. (4) can be simplified to give a reduced order of computation [28].

According to the mathematical identity,

\[
\frac{2}{3} A = \frac{2}{3} \left( \frac{\partial A}{\partial x} \right) \quad A: \text{matrix; } x,y, \text{ scalar variables}
\]

Eq. (4) can be manipulated as follows;

\[
\frac{\partial H_i}{\partial q_i} = H_i \frac{\partial^2 A_i}{\partial \delta q_i} 
\]

Eq. (11.1)
\[
\begin{pmatrix}
\frac{\partial^2 H^T_k}{\partial q_j \partial q_k} \\
\frac{\partial^2 H^T_k}{\partial q_j}
\end{pmatrix}
= \frac{\partial}{\partial q_j} \left( \frac{\partial H^T_k}{\partial q_k} \right) = \frac{\partial}{\partial q_j} \left( H^T_k \Delta^T_k \right)
\]

so,
\[
\frac{\partial^2 H^T_k}{\partial q_j \partial q_k} = \frac{\partial}{\partial q_j} \left( \frac{\partial H^T_k}{\partial q_k} \right) = \frac{\partial}{\partial q_j} \left( H^T_k \Delta^T_k \right)
\]

expanding will result in;
\[
\frac{\partial^2 H^T_k}{\partial q_j \partial q_k} = \left( \frac{\partial H^T_k}{\partial q_j} \right) \Delta^T_k + H^T_k \left( \frac{\partial \Delta^T_k}{\partial q_j} \right)
\]

neglect the second term of eq. (12) will yield,
\[
\frac{\partial^2 H^T_k}{\partial q_j \partial q_k} = \left( \frac{\partial H^T_k}{\partial q_j} \right) \Delta^T_k
\]

substituting (11.1) into (13),
\[
\frac{\partial^2 H^T_k}{\partial q_j \partial q_k} = H^T_k \Delta^T_k \Delta^T_k
\]

now substituting (11.2) and (14) into eq. (4) gives a better form for simulation purposes;
\[
P_{ijk} = \sum_{k=\max(i,j,k)}^{n} \text{tr} \left( H^T_k \Delta^T_k \Delta^T_k \right)
\]

Eq. (15) could be simplified further; premultiplying and postmultiplying by \( H^T_k \) and \( H^T_k \) respectively will effect the rotation part only, hence the trace operator will remain unchanged, eq. (15) will reduce to,
\[
P_{ijk} = \sum_{k=\max(i,j,k)}^{n} (\Delta^T_k \Delta^T_k \Delta^T_k)
\]

now expanding eq. (16) and assuming a matrix \( u \) such that:
\[
u = \Delta^T_k \Delta^T_k \Delta^T_k
\]

the matrix \( u \) will have the same form of matrix \( E \), i.e.
\[
u = \begin{bmatrix}
u_{11} & 0 & \cdots & 0 \\
0 & \ddots & \cdots & 0 \\
& & \ddots & \vdots \\
0 & 0 & \cdots & 0
\end{bmatrix}
\]

the trace operator will give:
\[
P_{ijk} = \sum_{k=\max(i,j,k)}^{n} \sum_{m=1}^{3} u_{mm}
\]
where \( \sum_{m=1}^{3} u_{mm} \) is given as,

\[
\begin{vmatrix}
\delta_{jx} & \delta_{ky} & \delta_{kz} \\
\delta_{iy} & \delta_{iz} & 1 \\
\end{vmatrix} + \begin{vmatrix}
\delta_{ix} & \delta_{iy} & \delta_{kz} \\
\delta_{ix} & \delta_{iy} & 1 \\
\end{vmatrix} + \begin{vmatrix}
\delta_{ix} & \delta_{iy} & \delta_{kz} \\
\delta_{ix} & \delta_{iy} & 1 \\
\end{vmatrix}
\]

\[
+ \begin{vmatrix}
\delta_{jz} & \delta_{kx} & \delta_{ky} \\
\delta_{jz} & \delta_{kx} & 1 \\
\end{vmatrix}
\]

\[
+ \begin{vmatrix}
\delta_{jx} & \delta_{kz} & \delta_{ky} \\
\delta_{jx} & \delta_{kz} & 1 \\
\end{vmatrix}
\]

\[
+ \begin{vmatrix}
\delta_{ix} & \delta_{iy} & \delta_{kz} \\
\delta_{ix} & \delta_{iy} & 1 \\
\end{vmatrix}
\]

\[
+ \begin{vmatrix}
\delta_{jx} & \delta_{kz} & \delta_{ky} \\
\delta_{jx} & \delta_{kz} & 1 \\
\end{vmatrix}
\]

\[
+ \begin{vmatrix}
\delta_{ix} & \delta_{iy} & \delta_{kz} \\
\delta_{ix} & \delta_{iy} & 1 \\
\end{vmatrix}
\]

\[
+ \begin{vmatrix}
\delta_{jx} & \delta_{kz} & \delta_{ky} \\
\delta_{jx} & \delta_{kz} & 1 \\
\end{vmatrix}
\]

\[
+ \begin{vmatrix}
\delta_{ix} & \delta_{iy} & \delta_{kz} \\
\delta_{ix} & \delta_{iy} & 1 \\
\end{vmatrix}
\]

For a 6-DOF robot arm at a given set of position variables, 216 elements of the matrix \([P_{ijk}]\) should be evaluated, but due to symmetry and other simplifying elements \([3,26]\) such as,

1. \( P_{ijk} = -P_{kji}; \quad i, k \neq j \) (reflexive coupling)

2. \( P_{ijk} = P_{ikj} \)
were used to reduce the number to 56 elements. Computational results are given in section 5.

4. The Load and No-Load Conditions:

4.1 Problem:

The previously derived symbolics will be used in the following discussion to re-emphasis the importance of the coriolis and centripetal forces in effecting the dynamic performance of robot manipulators under different conditions.

In the case of no-load the previous equations can be used directly. In case of load conditions the problem becomes much more complicated because of the dynamic interaction between the load and the arm \([33, 34, 35, 36, 39]\). It's well known that the last link of a robot arm contributes the most complex configurational dynamics, and with a load in the hand the problem will get complicated. In our study and for simplicity reasons the load will be assumed to be a cube which fits exactly in the hand with its centre of mass at the origin of the hand. This assumption will alter only the pseudo-inertia matrix of the last link \((\text{link } \ell)\). In case of eq. (3) and eq. (4) the pseudo-inertia matrix will change to,

\[
\mathbf{J}_n^{\text{load}} = \begin{bmatrix}
\frac{-K_{xx}^2 + K_{yy}^2 + K_{zz}^2}{m_n + m_{\text{load}}} & 0 & 0 & x_\ell \\
0 & 0 & \frac{K_{xx}^2 - K_{yy}^2 + K_{zz}^2}{2} & y_\ell \\
0 & \frac{K_{xx}^2 + K_{yy}^2 - K_{zz}^2}{2} & 0 & z_\ell \\
x_\ell & y_\ell & z_\ell & 1
\end{bmatrix}
\]  

(18)

and for eq. (10), the mass of link \((n)\) will be \((m_n + m_{\text{load}})\).

4.2 Quantification of Neglecting Coriolis/Centripetal Forces:

The error that results from neglecting the coriolis and centripetal effects was computed under different speeds and pay-loads.

The formula used to compute the error is given by,

\[
\text{Error} = \left| \frac{f_2 - f_1}{f_2} \right| \times 100\%
\]
Where

\[ f_2 : \text{joint forces including coriolis and centripetal effects.} \]
\[ f_1 : \text{joint forces excluding coriolis and centripetal effects.} \]
\[ |\cdot| : \text{absolute value.} \]

Two robot models were selected, the Stanford arm \([1,2,3,26]\) and the PUMA 560 \([37,38]\) to give a broader range of data.

As an example, each joint position parameter is chosen as \(q_i = 0.4\) radians \((i = 1,2,\ldots,6)\) for all the cases considered in our examples.

The resulting joint forces and torques were calculated for the different joint velocities, accelerations and loads. The velocities, accelerations and loads were chosen to give realistic simulation results, whilst maintaining consistency with the existing robot models.

In the case of load, the pseudo-inertia matrix of the two robot models will have the following form:

\[
\begin{bmatrix}
0.00059 & 0 & 0 & 0 \\
0 & 0.00059 & 0 & 0 \\
0 & 0 & 0.0516 & 0.1554 \\
0 & 0 & 0.1554 & 1
\end{bmatrix}
\]

\[
\begin{bmatrix}
0.000444 & 0 & 0 & 0 \\
0 & 0.000444 & 0 & 0 \\
0 & 0 & 0.0029 & 0.032 \\
0 & 0 & 0.032 & 1
\end{bmatrix}
\]

All the required data for the dynamics simulation can be found in published literature. For the Stanford arm the data is in \([1]\) and the PUMA 560 in \([37]\).

The result of the simulation are given in the next section.

5. Computational Results:

5.1 Execution time of the Simplified Symbolics:

Symbolic representation can lead to a better understanding and simplification of robot dynamic equations. This will enhance the development of efficient computational algorithms and programs \([29,30,31]\).

A very efficient FORTRAN program was written to test the computing time of the symbolics derived in this paper. The model chosen was a 6-dof Stanford arm and the computation time was calculated in two different ways, i.e.
(I) Including the cost of computing all the pre-required terms such as $H_1', \frac{\partial H_1}{\partial q_1}$ etc.

(II) Using the argument of Hollerbach [4] to exclude the computing time of the terms mentioned in (I) because of their dependency on the arm configuration.

The program was executed on a SUN workstation (32), and computation time are recorded in table 1.

<table>
<thead>
<tr>
<th>average CPU time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
</tr>
<tr>
<td>II</td>
</tr>
</tbody>
</table>

Table 1. CPU execution time

5.2 Quantification results and error analysis:

A set of compiled data of two robot arms (stanford, PUMA 560) was produced to study the dynamic performance of a robot manipulator when subject to different pay loads and speeds. The error committed at each joint when coriolis and centripetal effects are neglected was calculated. The results are recorded in table 2, 3, 4,

\[ \text{case (1):} \]

\[ \text{Load} = 0.0 \text{ Kg (no-load)} \]

<table>
<thead>
<tr>
<th>joint velocity &amp; acceleration $\dot{q}_1$ (rad/s)</th>
<th>Error in calculating joint forces (%)</th>
<th>Stanford</th>
<th>PUMA 560</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\dot{q}_1$ (rad/s)</td>
<td>$E_1$</td>
<td>$E_2$</td>
<td>$E_3$</td>
</tr>
<tr>
<td>0.5</td>
<td>15</td>
<td>2.6</td>
<td>0.5</td>
</tr>
<tr>
<td>1.0</td>
<td>35.4</td>
<td>7.07</td>
<td>2.1</td>
</tr>
<tr>
<td>1.5</td>
<td>65</td>
<td>11.8</td>
<td>5.0</td>
</tr>
<tr>
<td>2.0</td>
<td>109.7</td>
<td>16.3</td>
<td>9.0</td>
</tr>
<tr>
<td>2.5</td>
<td>190</td>
<td>20.5</td>
<td>14.3</td>
</tr>
<tr>
<td>3.0</td>
<td>364</td>
<td>24.4</td>
<td>20.7</td>
</tr>
</tbody>
</table>

Table 2.
**Case (2):**

Load = 1.0 Kg

<table>
<thead>
<tr>
<th>joint velocity &amp; acceleration $\dot{\theta}_1$ (rad/s)</th>
<th>Error in calculating joint forces (%)</th>
<th>Stanford</th>
<th>PUMA 560</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$E_1$</td>
<td>$E_2$</td>
<td>$E_3$</td>
</tr>
<tr>
<td>0.5</td>
<td>12.8</td>
<td>4.0</td>
<td>0.9</td>
</tr>
<tr>
<td>1.0</td>
<td>23.0</td>
<td>11.6</td>
<td>3.7</td>
</tr>
<tr>
<td>1.5</td>
<td>30.6</td>
<td>19.7</td>
<td>8.52</td>
</tr>
<tr>
<td>2.0</td>
<td>37.0</td>
<td>27.2</td>
<td>15.1</td>
</tr>
<tr>
<td>2.5</td>
<td>42.3</td>
<td>34.0</td>
<td>23.0</td>
</tr>
<tr>
<td>3.0</td>
<td>47.0</td>
<td>39.5</td>
<td>32.0</td>
</tr>
</tbody>
</table>

**Table 3.**

**Case (3):**

Load = 2.0 Kg

<table>
<thead>
<tr>
<th>joint velocity &amp; acceleration $\dot{\theta}_1$ (rad/s)</th>
<th>Error in calculating joint forces (%)</th>
<th>Stanford</th>
<th>PUMA 560</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$E_1$</td>
<td>$E_2$</td>
<td>$E_3$</td>
</tr>
<tr>
<td>0.5</td>
<td>30.3</td>
<td>4.5</td>
<td>1.2</td>
</tr>
<tr>
<td>1.0</td>
<td>46.5</td>
<td>13.7</td>
<td>5.0</td>
</tr>
<tr>
<td>1.5</td>
<td>56.6</td>
<td>23.4</td>
<td>11.08</td>
</tr>
<tr>
<td>2.0</td>
<td>64</td>
<td>32.3</td>
<td>19.2</td>
</tr>
<tr>
<td>2.5</td>
<td>69</td>
<td>40</td>
<td>29.0</td>
</tr>
<tr>
<td>3.0</td>
<td>72.3</td>
<td>46.2</td>
<td>38.3</td>
</tr>
</tbody>
</table>

**Table 4.**

The previous three cases were plotted to show the unpredictable non-linear changes in the dynamic performance, as shown in Fig 1,2,3.
Concluding Remarks:

The dynamic equations used to model robot manipulators consist of three types of effects of equal importance, the inertial, the coriolis and the centripetal, and the gravity terms. A simplified symbolic representation based on the lagrangian for the dynamics has been presented. FORTRAN programs were written to verify the derivation and computational time for a 6-DOF manipulator which are recorded in table 1. The derived symbolics were used to perform an error analysis study, and for the first time quantified results have been produced to measure the effect of neglecting the coriolis and centripetal terms on the dynamic performance of robot manipulators under different payloads and speeds. The analysis has been performed on both the Stanford and PUMA 560 arms and a set of compiled numerical data is presented in tables (2,3,4). Graphical representation of the data is given in Fig (1,2,3) to help in further illustration of the results.
References:


Fig. 1  Case 1

(A)

ERROR vs. SPEED at NO LOAD (STANFORD)

- Joint 1
- Joint 2
- Joint 3
- Joint 4
- Joint 5
- Joint 6

ERROR (\%)

SPEED (rad/sec)
(b)

ERROR vs. SPEED at NO LOAD (PUMA 560)

- Joint 1
- Joint 2
- Joint 3
- Joint 4
- Joint 5
- Joint 6

SPEED (rad/sec)

ERROR (°)

0
60
120
180
240
300

0
0.5
1
1.5
2
2.5
3
Fig. 2  Case 2

ERROR vs. SPEED at LOAD = 1.0 Kg (STANFORD)
ERROR vs. SPEED at LOAD = 1.0 Kg (FUKA 560)

- Joint 1
- Joint 2
- Joint 3
- Joint 4
- Joint 5
- Joint 6

SPEED (rad/sec)
Fig. 3. Case 3.

ERROR vs. SPEED at LOAD = 2.0 Kg (STANFORD)
ERROR vs. SPEED at LOAD = 2.0 Kg (PUMA 560)