ABSTRACT:
Current industrial robots are made very heavy to achieve high stiffness which increases the accuracy of their motion. However this heaviness limits the robot speed and in masses the required energy to move the system. The requirement for higher speed and better system performance makes it necessary to consider a new generation of light weight manipulators as an alternative to today’s massive inefficient ones. Light weight manipulators require less energy to move and they have larger payload abilities and more maneuverability. However due to the dynamic effects of structural flexibility, their control is much more difficult. Therefore, there is a need to develop accurate dynamic models for design and control of such systems.

This paper presents the flexibility and Kineto-Elasto dynamic analysis of robot manipulator PUMA-560. Based on the distributed parameter method, the generalized motion equations of robot manipulator with flexible links are derived. The final formulation of the motion equations can be used to model general complex elastic manipulators with nonlinear rigid-body and elastic motion in dynamics and can be used in the flexibility analysis of robot manipulators and spatial mechanisms. Manipulator end-effectors path trajectory, velocity and accelerations are plotted. Joint torques is to be determined for each joint trajectory. Using joint torques, static loading due to link’s masses, masses at joints, and payload, the PUMA 560 arms elastic deformations are to be found.

Key words : Flexibility Analysis, Elastic Deformations, Puma 560; Robot manipulator; Dynamics; Kinetic Analysis; Ansys 12.0; Mat Lab

INTRODUCTION
Machines are used in a variety of applications: pick-and-place operations, welding, machining, etc. Such machines can be divided into two units, the physical mechanism composed of links and actuators, and the control system. The number of actuators present in the mechanical system depends on the number of independent machine axes, or degrees of freedom. For example, a typical articulated six degree of Freedom manipulator contains six rotating actuators. A five-axis high-speed CNC machining centre would contain three linear actuators and two rotating actuators. A motion task given to the machine must ultimately be represented as a reference signal, which is sent to the control system. The control system acts to make the machine track the reference signal by activating the appropriate actuators. If the reference signal changes too quickly, given the dynamic limitations of the machine, the tracking of the reference signal will be poor, regardless of the control system design.

Computer algorithms are designed to calculate an appropriate reference signal based on the desired task path and time-related limits (such as speed and acceleration). This reference signal is the trajectory, and can be defined as a locus of points in operational or joint space on which a time-law has been specified. The generation of an appropriate trajectory is the problem that is being investigated in this thesis. The path along which the trajectory is defined can be point-to-point; namely, the machine is required to move between the two points but is not given any fixed intermediate path. This type of path is useful in manipulator pick-and-place operations. A path can also be completely specified through use of geometric functions. This type of path is commonly used in CNC machining applications or in manipulator applications when obstacles are present, or when it is necessary to ensure that the end effector follows as specific path.

RESEARCH OBJECTIVES
The objective of this work is to provide a method of generating smooth, near time-optimal trajectories on-line. Therefore, the trajectory generation
algorithm must be computationally simple. The trajectories must take into account the physical limitations of the machine, that is, not only their torque limitations but also their torque rate limitations. This work aims to produce a trajectory that will improve the motion time and the trajectory tracking of a standard industrial machine whether it is applied to the machine through a simplified controller or a more complex controller. The trajectory is to be planned such that it could be used to provide a time-law for any type of path, described in any space, provided that the corresponding path jerk, acceleration and velocity limits are derived. A further objective is to apply this technique to a robotic manipulator.

Robot classification

**ROBOT WORK ENVELOPES BASED ON MAJOR AXES**

<table>
<thead>
<tr>
<th>Robot</th>
<th>Axes 1</th>
<th>Axes 2</th>
<th>Axes 3</th>
<th>Total revolute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cartesian</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>0</td>
</tr>
<tr>
<td>Cylindrical</td>
<td>R</td>
<td>P</td>
<td>P</td>
<td>1</td>
</tr>
<tr>
<td>Spherical</td>
<td>R</td>
<td>R</td>
<td>P</td>
<td>2</td>
</tr>
<tr>
<td>SCARA</td>
<td>R</td>
<td>R</td>
<td>P</td>
<td>2</td>
</tr>
<tr>
<td>Articulated</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>3</td>
</tr>
</tbody>
</table>

P = prismatic  R = revolute

This assumption is true only if the elastic deformation of robot arm is far small compared with the size of the link cross-section. The rigid-body dynamic modeling based on this method is suitable for robots with Low speed, light load and low accuracy. With the Development of the robot with high speed, high precision and heavy load, the effect of elastic deformation must be taken into consideration. Elastic compensation inserted in coordinates of robotic programming to get exact end-effector’s path. A comparison of paths is to be plotted. Also variation of torques is to be plotted after considering elastic compensation so that the end effector’s path is corrected to improve the repeatability with the increased accuracy.

**PATH DEFINITION**

A manipulator task may be defined as a series of way-points, through which the manipulator should pass (unconstrained motion). However, it is often desirable to specify the path between the way-points to avoid obstacles present in the workspace, for example, or to facilitate a task such as spray-painting or welding. In robot manipulations, such tasks are generally described by straight-line motions in task space joined by polynomial spines at the way-points. The sp lines pass through or near the way-points so that the manipulator does not have to stop at each way-point. The objective in this constrained motion is to accurately track the path. Trajectory planning methods for both types of path specification have been developed and will be discussed below. In this thesis it is assumed that the path to be followed by the manipulator can be parametrised in terms of a single variable. The path is given in terms of the actuator positions (joint-space specification) or in terms of the placement of the tool tip in the work space (task-space specification).

![Fig. PROGRAMMING UNIVERSAL MANIPULATOR ARM 560](image)

**PROBLEM FORMULATION:**

The main tasks of the project

- For a given End-effector path, Joint trajectories are to be determined (Kinematics)
- Joints torques are to be determined for each joint trajectories (Dynamics), Smoothening the joint trajectories to avoid discontinuities.
Using joint torques, static, loading due to link’s masses, masses at joints, and payload, the PUMA 560 arms elastic deformations are to be found by using ANSYS-12.0 software package.

Elastic compensation inserted in coordinates of robotic programming to get exact end-effectors path. A comparison of paths is to be plotted. Also variation of torques is to be plotted after considering elastic compensation.

**KINEMATIC ANALYSIS:**

\[ \theta_1 = \tan^{-1}\left(\frac{y}{x}\right) \]

\[ r = \sqrt{x^2 + y^2} \]

\[ C = \sqrt{y^2 + (x - D_1)^2} \]

\[ A = \cos^{-1}\left(\frac{L_1^2 + C^2 + (L_2^2)}{2L_1C}\right) \]

\[ B = \sin^{-1}\left(\frac{L_3}{L_2}\right) \sin A \]

\[ \theta_2 = \tan^{-1}\left(\frac{r - D_1}{r}\right) - B \]

\[ \theta_3 = A + B \]

**CO-ORDINATES**

<table>
<thead>
<tr>
<th>( X ) (meters)</th>
<th>0.10</th>
<th>0.16</th>
<th>0.22</th>
<th>0.29</th>
<th>0.3</th>
<th>0.4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4</td>
<td>8</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>( Y ) (meters)</td>
<td>0.18</td>
<td>0.18</td>
<td>0.18</td>
<td>0.18</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>2</td>
<td>8</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( Z ) (meters)</td>
<td>0.15</td>
<td>0.15</td>
<td>0.16</td>
<td>0.16</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>2</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>( \theta_1 ) DEGREES</td>
<td>60.9</td>
<td>47.0</td>
<td>38.2</td>
<td>31.6</td>
<td>26.0</td>
<td>23.0</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>9</td>
<td>5</td>
<td>8</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>( \theta_2 ) DEGREES</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>118.</td>
<td>114.</td>
<td>108.</td>
<td>101.</td>
<td>94.</td>
<td>86.</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>3</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>( \theta_3 ) DEGREES</td>
<td>100.</td>
<td>99.6</td>
<td>97.6</td>
<td>92.3</td>
<td>84.</td>
<td>80.</td>
</tr>
<tr>
<td></td>
<td>02</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>
stored in a mode FRONTIER workflow and vice versa.

Mat lab program

\[
m_2 = 40; \quad m = \text{mass (kg)}
\]

\[
m_3 = 14.5; \quad L = \text{length (m)}
\]

\[
m_{23} = 2.5; \quad G = \text{gravity (m/s}^2)\]

\[
m_{34} = 5; \quad T = \text{time (sec)}
\]

\[
L_2 = 0.432; \quad L_3 = 0.432;
\]

\[
M_a = 0.5*m_2 + m_{23} + m_3 + m_{34};
\]

\[
I_c = (m_3 + 4*m_{34})*L_3^2;
\]

\[
I_a = (0.25*m_2 + m_{23} + m_3 + m_{34})*L_2^2;
\]

\[
G = 10; \quad I_b = (m_3 + 2*m_{34})*L_2*L_3;
\]

\[
T = 0:0.01:5; \quad M_b = m_3 + 2*m_{34};
\]

**THETA AND ITS DIFFERENTIATION**

\[
\theta_1 = 0.027*T^4 + 0.39*T^3 - 3*T^2 + 16*T - 0.0031;
\]

Plot (T, \theta_1); label ('Time [Sec]'); label ('Base rotation [Degrees]'); grid;

\[
\theta_1d = 0.108*T^3 + 1.17*T^2 - 6*T + 16;
\]

Plot (T, \theta_1d); x label('Time [Sec]'); y label('Base angular velocity [Degrees/sec]'); grid;

\[
\theta_1dd = 0.324*T^2 + 2.34*T;
\]

Plot (T, \theta_1dd); x label('Time [Sec]'); y label('Base angular acceleration [Degrees/sec^2]'); grid;

\[
T_11 = (I_a*cos(\theta_2))^2 + I_b*cos(\theta_2)*cos(\theta_2 + \theta_3) + 0.25*I_c*cos(\theta_2 + \theta_3)^2; * \theta_1dd;
\]

\[
T_12 = ((I_a*sin(2*\theta_2) + I_b*sin(2*\theta_2 + \theta_3)) + 0.25*I_c*sin(2*\theta_2 + 2*\theta_3)) * \theta_1d; * \theta_1dd;
\]

\[
T_13 = (I_b*cos(\theta_2)*sin(\theta_2 + \theta_3) + 0.25*I_c*sin(2*\theta_2 + 2*\theta_3)) * \theta_1d; * \theta_3d;
\]

\[
T_1 = T_11 - T_12 - T_13; \text{plot}(T,T_1); \text{x label}('Time [Sec]'); \text{y label}('Base Torque [N.m]'); \text{grid};
\]

\[
\theta_2 = 0.012*T^3 - 0.23*T^3 + 1.7*T^2 + 2.3*T + 0.028;
\]

Plot (T, \theta_2); x label ('Time [Sec]'); label ('Link1 rotation [Degrees]'); grid;

\[
\theta_2d = 0.048*T^3 - 0.69*T^2 + 3.4*T + 2.3;
\]

Plot (T, \theta_2d); x label('Time [Sec]'); label('Link1 angular velocity [Degrees/sec]'); grid;

\[
\theta_2dd = 0.0144*T^2 - 1.38*T + 3.4;
\]

Plot (T, \theta_2dd); x label('Time [Sec]'); label('Link1 angular acceleration [Degrees/sec^2]'); grid;

\[
\theta_3 = 0.0021*T^4 - 0.11*T^3 + 1.6*T^2 - 1.3*T + 0.041;
\]

Plot (T, \theta_3); x label('Time [Sec]'); label('Link2 rotation [Degrees]'); grid;

\[
\theta_3d = 0.0084*T^3 - 0.33*T^2 + 3.2*T - 1.3;
\]

Plot (T, \theta_3d); x label('Time [Sec]'); label('Link2 angular velocity [Degrees/sec]'); grid;

\[
\theta_3dd = 0.0252*T^2 - 0.066*T + 3.2;
\]

Plot (T, \theta_3dd); x label('Time [Sec]'); label('Link2 angular acceleration [Degrees/sec^2]'); grid;

\[
T_1 = 0.012*T^3 - 0.23*T^3 + 1.7*T^2 + 2.3*T + 0.028;
\]

Plot (T, \theta_2); x label ('Time [Sec]'); label ('Link1 rotation [Degrees]'); grid;

\[
\theta_2d = 0.048*T^3 - 0.69*T^2 + 3.4*T + 2.3;
\]
International Journal of Engineering Technology, Management and Applied Sciences

www.ijetmas.com  March 2017, Volume 5 Issue 3, ISSN 2349-4476

T23=(0.5*Ib*sin(theta3).*theta3d^2-Ib*sin(theta3).*theta2d).

*theta3d+Ma*L2*cos(theta2)+0.5*Mb*L3*cos(theta3)*g;

T2=T21+T22

T23;plot(T,T2);xlabel('Time[Sec]');ylabel('Link1 Torque[N.m]');grid

T31=(0.25*Ic+0.5*Ib*cos(theta3).*theta2d^2;+(0.25*Ic).*theta3d);*

T32=(0.5*Ib*cos(theta2)+0.25*Ic*cos(theta2+theta3).*sin(theta2+theta3)).*theta1d^2;*

T33=(0.5*Ib*sin(theta3).*theta2d^2+(0.5*Mb*L3*cos(theta2+theta3)*g));*

T3=T31+T32+T33;plot(T,T3);xlabel('Time[Sec]');ylabel('Link2 Torque[N.m]');grid

**FINITE ELEMENT ANALYSIS**

In the finite element method, the actual continuum or body of matter, such as a solid, liquid, or gas, is represented as an assemblage of subdivisions called finite elements. These elements are considered to be interconnected at specified joints called nodes or nodal point. The nodes usually lie on the element boundaries where adjacent elements are considered to be connected. Since the actual variation of the field variable (e.g., displacement, stress, temperature, pressure, or velocity) inside the continuum is not known, we assume that the variation of the field variable inside a finite element can be approximated by a simple function. These approximating functions (also called interpolation models) are defined in terms of the values of the field variables at the nodes. When field equations (like equilibrium equations) for the whole continuum are written, the new unknowns will be the nodal values of the field variable. By solving the field equations, which are generally in the form of matrix equations, the nodal values of the field variable will be known. Once these are known, the approximating functions define the field variable throughout the assemblage of elements. The solution of a general continuum problem by the finite element method always follows an orderly step-by-step process. With reference to static structural problems, the step-by-step procedure can be stated as follows:

**Step (i):** Discretization of the structure The first step in the finite element method is to divide the structure or solution region into subdivisions or elements. Hence, the structure is to be modeled with suitable finite elements. The number, type, size, and arrangement of the elements are to be decided.

**Step (ii):** Selection of a proper interpolation or displacement model Since the displacement solution of a complex structure under any specified load conditions cannot be predicted exactly, we assume some suitable solution within an element to the unknown solution. The assumed solution must be simple from a computational stand point, but it should satisfy certain convergence requirements. In general, the solution or the interpolation model is taken in the form of a polynomial.

**Step (iii):** Derivation of element stiffness matrices and load vectors From the assumed displacement model, the stiffness matrix [K (e)] and the load vector P (e) of element e are to be derived by using either equilibrium conditions or a suitable variation principle.

**Step (iv):** Assemblage of element equations to obtain the overall equilibrium equations Since the
structure is composed of several finite elements, the individual element stiffness matrices and load vectors are to be assembled in a suitable manner and the overall equilibrium equation as formulated as

\[ [K] \Phi = \bar{F} \]

**Step (v): Solution for the unknown nodal displacements**

The overall equilibrium equations have to be modified to account for the boundary conditions of the problem.

**Step (vi): Computation of element strains and stresses:** the known nodal displacements, if required, the element strains and stresses computed by using the necessary equations of solid or structural mechanics.

**AXIAL DISPLACEMENTS:** The nodal displacements are \( q_1 \) and \( q_7 \) and a linear displacement model leads to the stiffness matrix (corresponding to the axial displacement) as

\[ [k_{a}^{(e)}] = \iiint_{V} [B]^T [D] [B] \, dV = \frac{AE}{l} \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} q_1 \\ q_7 \end{bmatrix} \]

where \( A, E, \) and \( l \) are the area of cross section, Young's modulus and length of the element, respectively. Notice that the elements of the matrix \( [k_{a}^{(e)}] \) are identified by the degrees of freedom indicated at the top and right-hand side of the matrix.

**TOTAL ELEMENT STIFFNESS MATRIX**

The stiffness matrices derived for different sets of independent displacements can now be compiled (superposed) to obtain the overall stiffness matrix of the frame element as

**GLOBAL STIFFNESS MATRIX**

It can be seen that the 12 • 12 stiffness matrix given with respect to the local \( x, y, z \) coordinate system. Since the nodal displacements in the local and global coordinate systems are related by the relation

\[
\begin{bmatrix}
q_1 \\
q_2 \\
q_3 \\
q_4 \\
q_5 \\
q_6 \\
q_7 \\
q_8 \\
q_9 \\
q_{10} \\
q_{11} \\
q_{12}
\end{bmatrix} =
\begin{bmatrix}
l_{12} & m_{12} & n_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_{12} & m_{12} & n_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_{12} & m_{12} & n_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_{12} & m_{12} & n_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_{12} & m_{12} & n_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_{12} & m_{12} & n_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_{12} & m_{12} & n_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_{12} & m_{12} & n_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_{12} & m_{12} & n_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_{12} & m_{12} & n_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_{12} & m_{12} & n_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_{12} & m_{12} & n_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

**TRANSFORMATION MATRIX**

We shall derive the transformation matrix \( [\lambda_1] \) between the local and global coordinate systems in two stages. In the first stage, we derive a transformation matrix \( [\lambda_1] \) between the global coordinates \( XYZ \) and tile coordinates \( x, y, z \) by assuming the \( z \) axis to be parallel to the \( XZ \) plane

\[
\begin{bmatrix}
x \\
y \\
z
\end{bmatrix} = [\lambda_1] \begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix}
\]

and

\[
\begin{bmatrix}
x \\
y \\
z
\end{bmatrix} = [\lambda_2] \begin{bmatrix}
\bar{x} \\
\bar{y} \\
\bar{z}
\end{bmatrix}
\]
by assuming that the local coordinate system \((xyz)\) can be obtained by rotating the \((xyz)\) system about the \(x\) axis by an angle \(\theta\). Thus, the desired transformation between the \(x\ y\ z\) system and the \(XYZ\) system can be obtained as

\[
[A] = [\lambda_2][\lambda_1]
\]

\[
\begin{bmatrix}
\frac{x}{y} \\
\frac{z}{z}
\end{bmatrix} = [\lambda]
\begin{bmatrix}
\frac{X}{X} \\
\frac{Y}{Y} \frac{Z}{Z}
\end{bmatrix}
\]

Since the unit vector \(k\) (which is parallel to the \(z\) axis) is normal to both the unit vectors \(J\) (parallel to the \(Y\) axis) and \(i\) (parallel to the \(x\) axis), we have, from vector analysis

\[
\hat{k} = \frac{i \times j}{\|i \times j\|} = \frac{1}{d} \begin{bmatrix}
i_x & j_x & k_x \\
i_y & j_y & k_y \\
i_z & j_z & k_z
\end{bmatrix}
\]

\[
d = (l_{o_x}^2 + n_{o_x}^2)^{1/2}
\]

Thus, the direction cosines of the \(z\) axis with respect to the global \(XYZ\) system are given by

\[
l_{o_z} = -\frac{n_{o_z}}{d}, \quad m_{o_z} = 0, \quad n_{o_z} = \frac{l_{o_x}}{d}
\]

To find the direction cosines of the \(y\) axis, we use the condition that the \(y\) axis (unit vector \(j\)) is normal to the \(x\) axis \(i\) and to the \(z\) axis \(k\).

\[
j = \hat{k} \times i = \begin{bmatrix}
i_x & j_x & k_x \\
i_y & j_y & k_y \\
i_z & j_z & k_z
\end{bmatrix}
\]

\[
= \frac{1}{d} \begin{bmatrix}
i(-l_{o_x}m_{o_z}) - j(-n_{o_z} - l_{o_z}^2) + k(-m_{o_x}n_{o_z})
\end{bmatrix}
\]

Thus, the direction cosines of the \(y\) axis are given by

\[
l_{o_y} = \frac{-l_{o_x}m_{o_z}}{d}, \quad m_{o_y} = \frac{n_{o_z}^2 + l_{o_z}^2}{d}, \quad n_{o_y} = \frac{-m_{o_x}n_{o_z}}{d}
\]

Thus, the \([\lambda]\) matrix is given by

\[
[\lambda] = \begin{bmatrix}
l_{o_x} & m_{o_x} & n_{o_x} \\
l_{o_y} & m_{o_y} & n_{o_y} \\
l_{o_z} & m_{o_z} & n_{o_z}
\end{bmatrix}
\]

\[
= \begin{bmatrix}
l_{o_x}/d & m_{o_x}/d & n_{o_x}/d \\
-l_{o_x}m_{o_z}/d & (l_{o_z}^2 + n_{o_z}^2)/d & -m_{o_x}n_{o_z}/d \\
-l_{o_z}/d & -n_{o_z}/d & l_{o_z}/d
\end{bmatrix}
\]

**Expression for \([\lambda_1]\)**

When the principal cross-sectional axes of the frame element \((xyz)\) axes) are arbitrary, making an angle \(\alpha\) with the \(x\ y\ z\) axes (\(x\) axis is same as \(x\) axis), the transformation between the two systems can be expressed as
\[
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{z}
\end{bmatrix} =
\begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \alpha & \sin \alpha \\
0 & -\sin \alpha & \cos \alpha
\end{bmatrix}
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{z}
\end{bmatrix} = [\lambda_2]
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{z}
\end{bmatrix}
\]

\[
[\lambda_2] =
\begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \alpha & \sin \alpha \\
0 & -\sin \alpha & \cos \alpha
\end{bmatrix}
\]

**FE MODEL**

**LIST OF DISPLACEMENTS**

<table>
<thead>
<tr>
<th>SL No</th>
<th>(U_x) in m</th>
<th>(U_y) in m</th>
<th>(U_z) in m</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.17036*10^{-4}</td>
<td>0.18</td>
<td>0.87724*10^{-5}</td>
</tr>
<tr>
<td>2</td>
<td>-0.23950*10^{-5}</td>
<td>0.18</td>
<td>0.13107*10^{-5}</td>
</tr>
<tr>
<td>3</td>
<td>-0.38199*10^{-3}</td>
<td>0.18</td>
<td>-0.8653*10^{-3}</td>
</tr>
<tr>
<td>4</td>
<td>0.14442*10^{-3}</td>
<td>0.18</td>
<td>0.10662*10^{-3}</td>
</tr>
<tr>
<td>5</td>
<td>-0.19450*10^{-3}</td>
<td>0.18</td>
<td>0.30069*10^{-3}</td>
</tr>
<tr>
<td>6</td>
<td>-0.67817*10^{-3}</td>
<td>0.18</td>
<td>-0.18229*10^{-2}</td>
</tr>
</tbody>
</table>

**LOADING CONDITIONS**

<table>
<thead>
<tr>
<th>Position</th>
<th>Joint torque at node 2 (MZ in Nm)</th>
<th>Joint torque at node 4</th>
<th>Mass at joint (at node 2)</th>
<th>Mass at joint (at node 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>111.4</td>
<td>681.2</td>
<td>25</td>
<td>50</td>
</tr>
<tr>
<td>2</td>
<td>723.3</td>
<td>-104.4</td>
<td>25</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>-156.1</td>
<td>51.21</td>
<td>25</td>
<td>50</td>
</tr>
<tr>
<td>4</td>
<td>651.2</td>
<td>-176.3</td>
<td>25</td>
<td>50</td>
</tr>
<tr>
<td>5</td>
<td>1385</td>
<td>744.6</td>
<td>25</td>
<td>50</td>
</tr>
<tr>
<td>6</td>
<td>-1175</td>
<td>-424.8</td>
<td>25</td>
<td>50</td>
</tr>
</tbody>
</table>

**DEFORMED SHAPES**

*Position 1*

*Position 2*

*Position 3*

*Position 4*
COMPARATIVE RESULTS:

Base link path (Equation-1)

Link1 path (Equation-2)

Link2 path (Equation-3)

CONCLUSION

This paper presented the Kineto - Elasto dynamic analysis of robot manipulator PUMA560. The end-effector holding an object and passing through a considered path trajectory, the co-ordinate positions of the end-effector is considered at five positions. Inverse kinematic analysis has been performed for finding the corresponding link positions. Dynamic analysis is performed to find the velocities, accelerations and joint torques for moving the end-effector in the considered path trajectory with the help of MATLAB-2008a software Using joint torques, static, loading due to link’s masses, masses
at joints, and payload, the PUMA 560 arms elastic deformations are found by using ANSYS-12.0 software package. Elastic compensation inserted in the co-ordinates of robotic programming to get exact end-effectors path. A comparison of path trajectories and variation of torques is plotted after considering elastic compensation. It is suggested that by compensating the joints torque variations in the robotic programming, the trajectory path of the end-effector will be accurate than the specified repeatability of ±0.1 mm in the manual of PUMA-560.

REFERENCES


7) E Barbieri,. And U.Ozguner, "Unconstrained andconstrained model expansions for a flexible slewing link", *AMSE Journal of Dynamic Systems, Measurement and Control,* Vol. 110, No.4,