

Kineto-Elasto Dynamic Analysis of Robot Manipulator Puma-560

D. Pratap, Dr. Y. V. Mohan Reddy

1 (Mechanical Engineerig,G.Pullu Reddy Engineerig College (Autonomous) /J Ntu Anantapur , India)

2 (Mechanical Engineerig,G.Pullu Reddy Engineerig College (Autonomous) /J Ntu Anantapur , India)

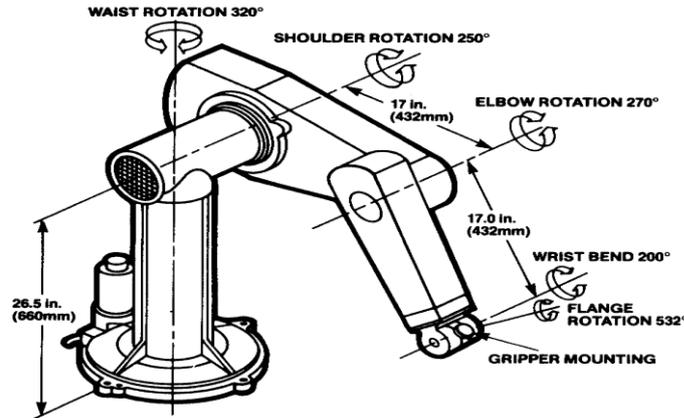
Abstract: Current industrial robots are made very heavy to achieve high Stiffness which increases the accuracy of their motion. However this heaviness limits the robot speed and in masses the required energy to move the system. The requirement for higher speed and better system performance makes it necessary 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 project presents the flexibility and Kineto - Elasto dynamic analysis of robot manipulator considering deflection. 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 is used to model general complex elastic manipulators with nonlinear rigid-body and elastic motion in dynamics and it can be used in the flexibility analysis of robot manipulators and spatial mechanisms. Manipulator end-effector path trajectory, velocity and accelerations are plotted. Joint torques is to be determined for each joint trajectory (Dynamics) .Using joint torques, static loading due to link's masses, masses at joints, and payload, the robot arms elastic deformations are to be found by using ANSYS-12.0 software package. Elastic compensation is inserted in coordinates of robotic programming to get exact end-effectors path. A comparison of paths trajectory of the end-effector is to be plotted. Also variation of torques is plotted after considering elastic compensation. These torque variations are included in the robotic programming for getting the accurate end-effect or's path trajectory.

Keywords: Manipulator, Speed, Accuracy,, Path trajectory, FEA analysis

I. 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.



1.11 General Specifications

Performance:

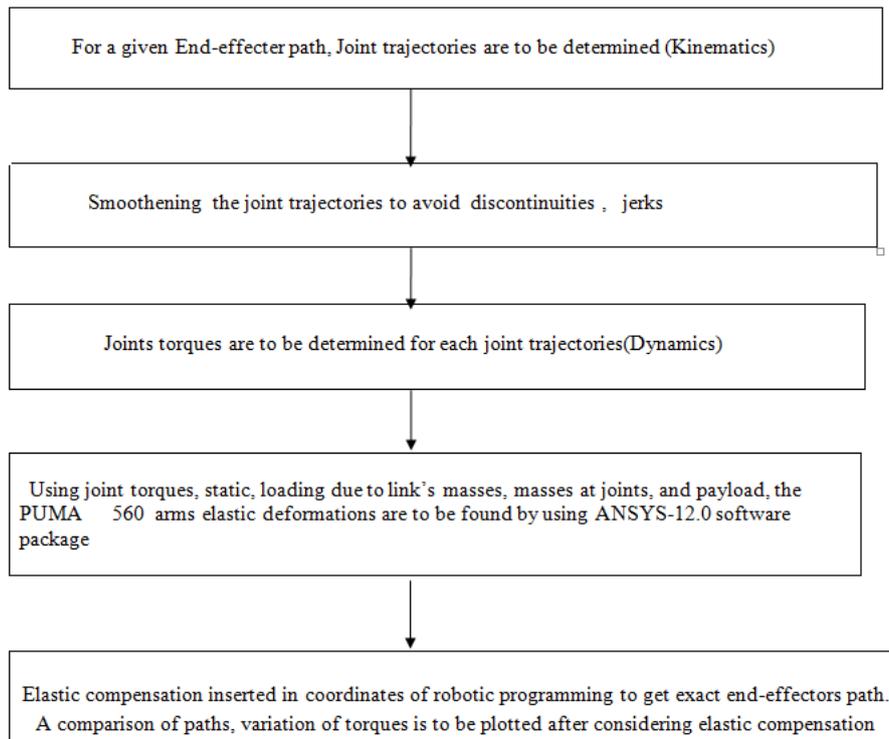
Repeatability	± 0.004in. (±0.1mm)
Load capacity	5.5 lbs. (2.5 kg)
Straight line velocity	20 in/s max. (0.5 m/s max.)

II. Flow Chart

2.1 Problem Formulation

Flow chart

The main tasks of the project



III. Literature Survey

Numerous investigations have been carried out in the fields of Dynamics of flexible robot manipulator and its deflections, velocity, acceleration, torque, variations and are described below

MOH.KHAIRUDIN in 2010 discussed about the modeling of a flexible link manipulator using lagrangian technique in conjunction with the assumed mode method. A pay load is added to the tip of the outer link, while hub inertias are included at the actuator joints. The lagrangian approach is used to derive the dynamic model of the structure, the dynamic model of the flexible link manipulator verified using matlab. The

emphasis has been set on obtaining accurate equation of motion that display the most relevant aspects of the coupling between rigid and flexible dynamics.

A.AHMADI NADOOSHAN in 2008 discussed about the modeling and vibration of a flexible link manipulator with tow flexible link and rigid joints are investigated which can include an arbitrary number of flexible links. Hamilton principle and finite element approach is proposed to model the dynamics of flexible manipulators the links are assumed to be deflection due to bending. The association between elastic displacements of links is investigated, took in to the account the coupling effects of elastic motion and rigid motion. The dynamic behaviour due to the flexibility of links is demonstrated through numerical simulation. The rigid-body motion and elastic deformation are seperated by linearizing the equations of motion around rigid body reference path. Simulation results are shown on for both position and force trajectory tracking tasks in the presence of varying parameters. The proposed method can be used in both dynamic simulation and controller design.

E. Abele in 2008 discussed about different implementations of elastic joint models of industrial robots are described and compared established in ADAMS and SimMechanics. The models are intended to be used for path prediction under process force load due to Roboforming and high speed cutting, respectively. The computational results have been compared and showed good agreement. In experiments of robot forming and robot milling the measured and simulated path deviations according to the process force are compared. The experiments are descriped and the results are discussed within the paper as a basis of a next step model based compensation of the path deviation.

Sinan Kilicaslan in 2009 proposed robot manipulators with flexible links, a new method is proposed to control their constrained end-effectors motions and the associated contact forces and/or moments. The equations of motion are separated into two parts that represent the pseudo static equilibrium and the deviations from it. The feed forward part of the control input based on the pseudo static equilibrium is determined algebraically and the feedback part of the control input for the stabilization of the deviations is obtained by means of a state-variable feedback law using measurements of the contact forces and/or moments, the strains in the links, the joint variables, and the end-effectors position and velocity. The feedback gain matrix is determined online by a continuously updated pole placement process. The method is demonstrated by means of a planar two-link robot with a flexible forearm which is constrained to move on a cylindrical surface. Furthermore, in order to investigate the effects of the modeling discrepancies caused by using lower order models, a "sub model controller" is designed using a model with only the first mode and it is applied to the same system with the two-mode model.

IV. Matlab

4.1 Basics Of Matlabmatlab (matrix laboratory) is a numerical computing environment and fourth-generation programming language. Developed by Math Works, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, and Fortran.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the symbolic engine, allowing access to computing capabilities. An additional package, Simulink, adds graphical multi-domain simulation and Model-Based Design for dynamic and embedded systems.

In 2004, MATLAB had around one million users across industry and academia. MATLAB users come from various backgrounds of engineering, science, and economics. MATLAB is widely used in academic and research institutions as well as industrial enterprises.

Vectors/Matrices

MATLAB is a "Matrix Laboratory", and as such it provides many convenient ways for creating vectors, matrices, and multi-dimensional arrays. In the MATLAB vernacular, a vector refers to a one dimensional ($1 \times N$ or $N \times 1$) matrix, commonly referred to as an array in other programming languages. A matrix generally refers to a 2-dimensional array, i.e. an $m \times n$ array where m and n are greater than or equal to 1. Arrays with more than two dimensions are referred to as multidimensional arrays.

Structures

MATLAB supports structure data types. Since all variables in MATLAB are arrays, a more adequate name is "structure array", where each element of the array has the same field names. In addition, MATLAB supports dynamic field names (field look-ups by name, field manipulations etc). Unfortunately, MATLAB JIT does not support MATLAB structures, therefore just a simple bundling of various variables into a structure will come at a cost.

Function Handles

MATLAB supports elements of lambda-calculus by introducing function handles, or function references, which are implemented either in .m files or anonymous/nested functions.

Secondary Programming

MATLAB also carries secondary programming which incorporates the MATLAB standard code into a more user friendly way to represent a function or system.

Classes

MATLAB supports classes, however the syntax and calling conventions are significantly different than in other languages, because MATLAB does not have reference data types. For example, a call to a method objects. Method (); cannot normally alter any variables of object variable. To create an impression that the method alters the state of variable, MATLAB toolboxes use evalin () command, which has its own restrictions.

Object Oriented Programming

MATLAB's support for object-oriented programming includes classes, inheritance, virtual dispatch, packages, pass-by-value semantics, and pass-by-reference semantics.

Interfacing With Other Languages

MATLAB can call functions and subroutines written in the C programming language or FORTRAN. A wrapper function is created allowing MATLAB data types to be passed and returned.

Elastic compensation co-ordinates

10	0.4200	0.18	0.1800	23.1986	-126.08	141.960
9	0.3844	0.18	0.1767	25.0894	-132.72	152.752
8	0.3489	0.18	0.1733	27.290	-138.23	161.870
7	0.3133	0.18	0.1700	29.876	-142.77	169.474
6	0.2778	0.18	0.1667	32.9432	-146.48	175.700
5	0.2422	0.18	0.1633	36.616	-149.52	180.679
4	0.2067	0.18	0.1600	41.054	-152.07	184.546
3	0.1711	0.18	0.1567	46.50	-154.28	187.44 2

2	0.1356	0.18	0.1533	53.018	-156.28	189.50 2
1	0.10	0.18	0.15	60.945 4	-158.13	190.85 1
S.No	X(M)	Y(M)	Z(M)	θ_1 (Deg)	θ_2 (Deg)	θ_3 (Deg)

Table 4.1.1 co-ordinates

4.2 Mat lab program

```

m2 = 40;
m3 = 14.5;
m23 = 2.5;
m34 = 5;
L2 = 0.432;
L3 = 0.432;
Ma = 0.5*m2+m23 + m3 + m34;
G = 10; Ib = (m3+2*m34)*L2*L3;
m = mass (kg)
L = length (m)
G = gravity (m/s2)
T = time (sec)
Ic = (m3+4*m34)*L3^2;
Ia = (0.25*m2+m23+m3+m34)*L2^2;
Mb = m3+2*m34;
    
```

4.3 Linear Order Equation For Base

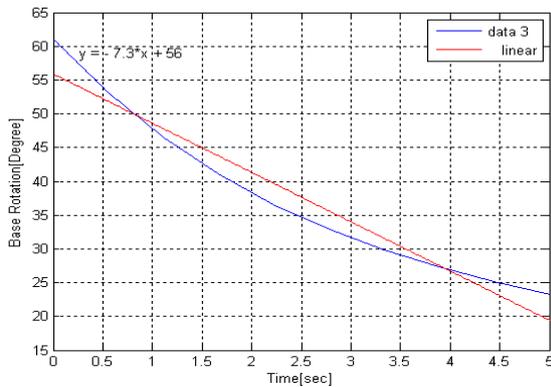


Fig.1 Base Rotation

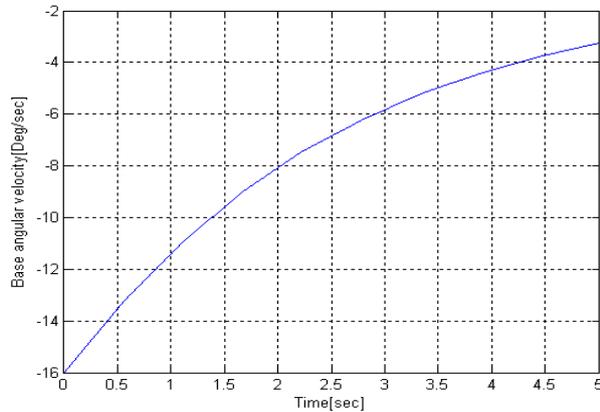


Fig.2 Base Angular Velocity

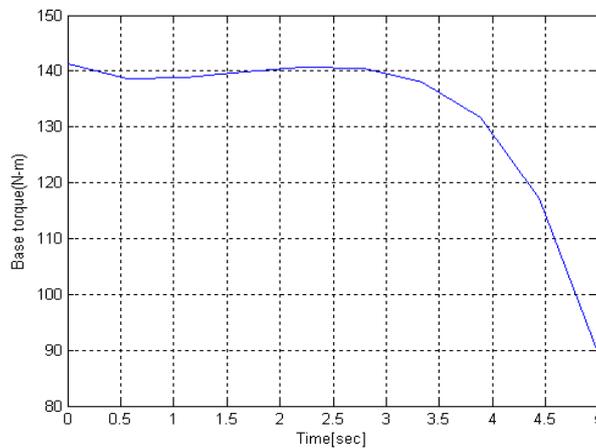


Fig.3 Base Torque

V. Fea Analysis

ANSYS Ver. XI is used for finite element Analysis of the Robot Manipulator

5.1 Element types:

5.1.1 BEAM188 Element Description

3-D Linear Finite Strain Beam: BEAM188 is suitable for analyzing slender to moderately stubby/thick beam structures. This element is based on Timoshenko beam theory. Shear deformation effects are included.

BEAM188 is a linear (2-node) or a quadratic beam element in 3-D. BEAM188 has six or seven degrees of freedom at each node, with the number of degrees of freedom depending on the value of KEYOPT(1). When KEYOPT (1) = 0 (the default), six degrees of freedom occur at each node. These include translations in the x, y, and z directions and rotations about the x, y, and z directions. When KEYOPT (1) = 1, a seventh degree of freedom (warping magnitude) is also considered. This element is well-suited for linear, large rotation, and/or large strain nonlinear application

5.2 Graphical user interface process

a) Preferences > structural

b) Element type>Add

1. BEAM- 188
2. MPC184 REVOLUTE JOINT

a) Real constants

For BEAM- 188

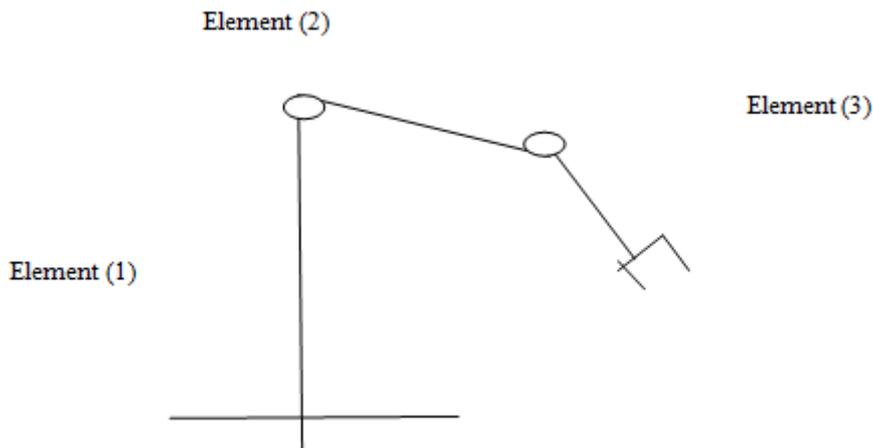
1. Link1
 $R_i = 0.0331\text{m}$
 $R_o = 0.0496\text{ m}$
2. Link2
 $h = 0.189\text{ m}$
 $b = 0.063\text{ m}$
3. Link3
 $h = 0.189\text{ m}$
 $b = 0.063\text{ m}$

d) Material properties

Young's Modulus= $2e11\text{ N/m}^2$
 Density= 7800 kg/m^3
 Poisson's ratio= 0.3

e) FE Model

- create > (Node 1)
 (Node 2, 3)
 (Node 4, 5)
 (Node 6)



Boundary conditions (Degrees of Freedom)At node 1 : All degrees of freedom fixed
 At node 2 & 3: UX, UY, UZ, ROTX and ROTY=0

At node 4 & 5: Revolute joint

Loading conditions

Position	Joint torque at node 2 (MZ in Nm)	Joint torque at node 4 (MZ in Nm)	Mass at joint (atnode4) (in Nm)	Mass at joint (at node 6) (in Nm)
1	172.6	-183.9	-25	-50
2	167	-107	-25	-50
3	162.4	-51.23	-25	-50
4	155.3	-14.8	-25	-50
5	139.2	10.7	-25	-50

Table 5.2.1 loading conditions

Processing Solution ctrl> ok
Solve > current L.S > ok

Post processing

a) List of Displacements

SL.No	U_x (in m)	U_z (in m)
1	-0.4953×10^{-4}	-0.36103×10^{-4}
2	-0.55871×10^{-5}	-0.10547×10^{-4}
3	-0.34072×10^{-3}	-0.60530×10^{-2}

Table 5.2.2 List of Displacements

5.3.1 Deformed shape

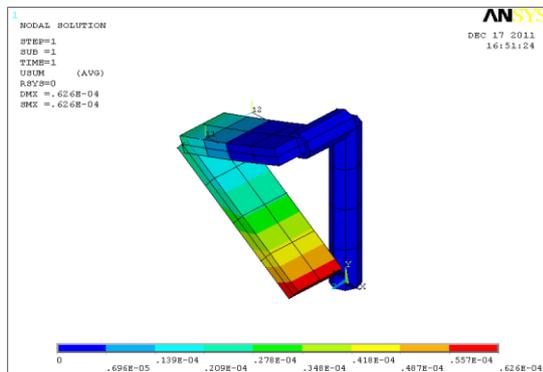


Figure 5.3.1 Deformed shape

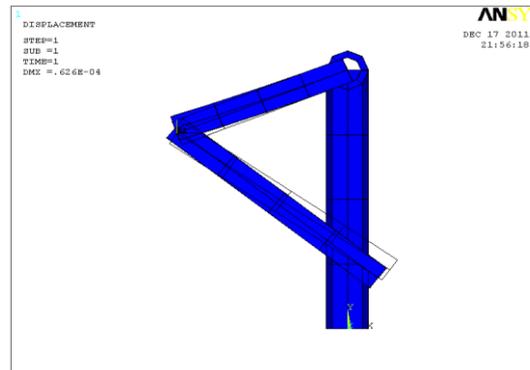


Figure 5.3.2 Position 1

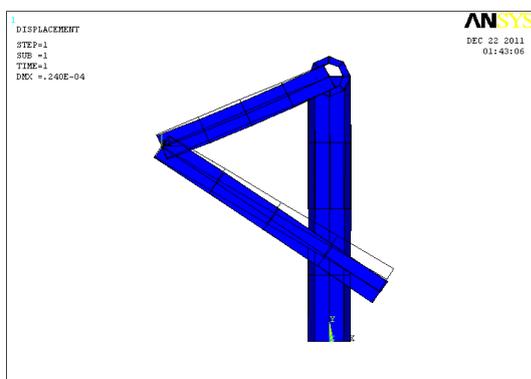


Figure 5.3.4 Position 3

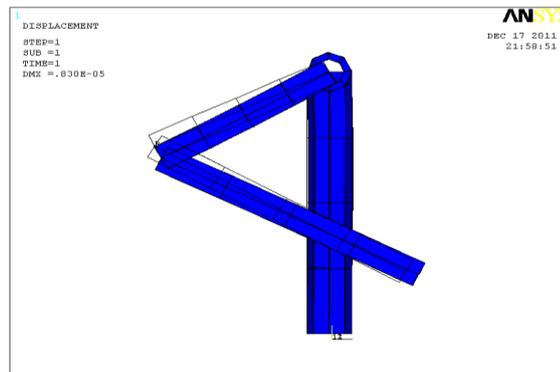


Figure 5.3.3 Position 2

VI. Comparative results

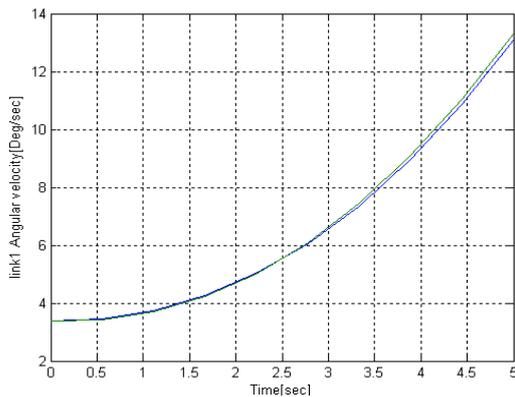


Fig.1 Angular Velocity

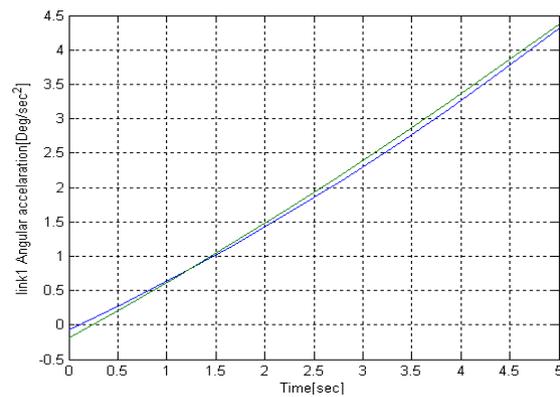


Fig.2 Angular Acceleration

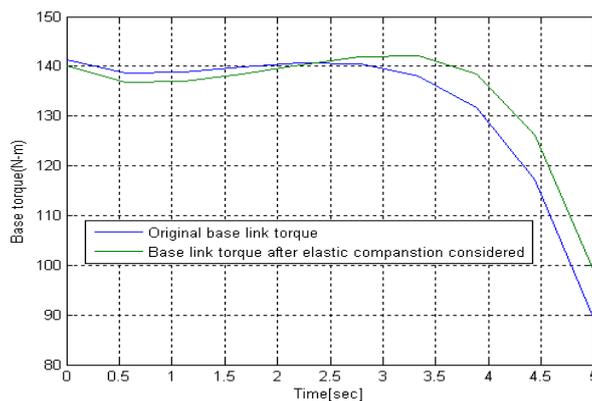


Fig.3 Base Torque

VII. Conclusion

This project presented the Kineto-Elasto dynamic analysis of robot manipulator. The end-effectors holding an object and passing through a considered path trajectory, the co-ordinate positions of the end-effectors is considered at ten 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-effectors 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 Robot manipulator 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-effectors will be accurate than the specified repeatability of ± 0.1 mm in the manual of PUMA-560.

7.1 Future Scope

- 1 The robotic programming language can be modified and that can be incorporated and tested with puma robot for getting accurate path trajectory of the end effectors according to the result obtained in this theses work.
2. This analysis may be further extended by considering the inertia effect due to the speed increased in the robot arm. Feed back control for position, velocity and acceleration can be incorporated in the analysis.

References

- [1] Moh.khairudin, "Dynamic modeling of a flexible link manipulator robot "university negeri, Yogyakarta, Indonesia.2010 ISSN 1693-6930.
- [2] A.Ahmadi nadooshan, E.Abedi, s.salehi,"Dynamic modeling of two flexible link manipulator",2008International journal of aero space and mechanical engineering.
- [3] B.Subudhi, A.S morris "Dynamic modeling,simulation and controlling of a manipulator with flexible links and joints",2008Robotics and atonomous systems.257-270.
- [4] K.S. Fu, R.C. Gonzalez, and C.S.G. Lee, Robotics Control, Sensing, h i o n , and Intelligence,;McGraw-Hill, 1987.