Dynamic Modeling of Three Links Robot Manipulator (Open Chain) with Spherical Wrist

Hassan Mohammad Alwan¹, Zaid Hikmat Rashid²

Abstract

Dynamic modeling of a robot manipulator is a central problem in an accurate robot control. In this paper; the dynamic equations of motion were derived by using Euler-Lagrange method for a six degree of freedom articulated robot manipulator based on the geometrical jacobian construction for each link and actuator. In addition, friction effects beside the end effector forces that act the environment are considered. A Matlab Simulink plant is developed to embrace the theoretical work and simulate the dynamic response for a designed nonlinear controller Proportional Derivative plus Gravity (PD+G), also a modified controller is applied to reject the disturbances and the internal friction effect where the settling errors were 3.57E-6, 2.09E-7, -3.63E-6, 8.84E-6, -5.39E-8 and -4.39E-5 (deg) for joints one to six respectively. The presented approach can be applicable to solve the dynamic problem of other n-link robot manipulators and achieve a suitable solution for tracking trajectories.

Keywords: Forward Kinematic, Jacobian, Nonlinear Controller, ode45, Solid Works.

1. Introduction

The robot manipulator has a high nonlinearity in dynamics and many inner variable parameters that effect on the dynamic response such as the inertia, Coriolis and friction forces so the precise dynamic model of a robot is an important step to achieve high performance robot control [1]. J. kardos[2] presented a simplified dynamic model for a three degree of freedom (DOF) anthropomorphic robot based on Euler Lagrange method using Matlab Simulink by considering the concentration of the mass of each link into its center of gravity, while Warthik and Wael[3] presented the modeling and control of the LabVolt 5250 robot arm 5 DOF by assuming each link is a homogenous cylinder; H. Al-Qahtani et al [4] presented the dynamics and control of a robot having four links where the Euler Lagrange analysis is carried out for the dynamics modelling; H. Al-Dois et al [5] described an analyzing method of dynamic performance for serial robot manipulators where they presented a numerical example for PUMA 560 as an illustrative case also A. Izadbakhsh [6] presented an...
explicit dynamic model of PUMA 560 robot without any mathematical simplifications and compare numerically the dynamics errors of different inputs with the proposed model of B. Armstrong [7]; Y.D. Patel and P. M. George [8] were used the analytical calculations of Newton Euler analysis for joints torques and observed that around 3% variation compared to Euler Lagrange approaches using Matlab and Pro/engineer software.

In this work, a three links robot arm with spherical wrist (six DOF) is adopted as a case study as shown in Fig.1. The Forward kinematic is presented based on Danivat Hartenberg convention in order to get the positions and orientations vectors from the transformation matrices which are useful in geometrical jacobian construction where the jacobian is one of the most important tool for manipulator characterization in finding singularities[9], determining inverse algorithm, related the joint torques and applied forces, deriving equations of motion and designing operational control schemes[10]. Also the Euler Lagrange energy method is used to derive the equations of motions since it treats the robot as a whole by taking the total kinetic and potential energies of links and actuators to give a more compact and direct model which can be easily edited to add the friction and disturbance torques.

2. Forward Kinematics Modeling

Forward kinematics is the transformation between joint space and the cartesian space to solve the position and orientation of the robot end effector. Denavit Hartenberg (DH) convention computes the forward kinematic by attaching a coordinate frame system at each joint and specifying the four parameters of DH: \( a_i \), \( \alpha_i \), \( d_i \) and \( \theta_i \) where: [11]

- \( a_i \) : (link length) is the distance between \( z_{i-1} \) and \( z_i \) axes along the \( x_i \) axis.
- \( \alpha_i \) : (link twist) is the required rotation of \( z_{i-1} \) to \( z_i \) axes about the \( x_i \) axis.
- \( d_i \) : (joint offset) is the distance between \( x_{i-1} \) and \( x_i \) axes along the \( z_{i-1} \) axis.
- \( \theta_i \) : (joint angle) is the required rotation of \( x_{i-1} \) to \( x_i \) axes about the \( z_{i-1} \) axis.

The transformation matrix of frame \( \{i\} \) relative to previous frame \( \{i-1\} \) is:

\[
\begin{bmatrix}
  c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\
  s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\
 0 & sa_i & ca_i & d_i \\
 0 & 0 & 0 & 1
\end{bmatrix}
\]  

...(1)

And the transformation matrix of nth coordinate frame to base coordinate frame is:

\[
^{0}_{n}T = ^{0}_{1}T T_{1,2} .... T_{n-1,n}
\]

...(2)

The first (3x3) matrix represents rotation matrix of frame \( \{i\} \) relative to frame \( \{i-1\} \) and the fourth column represents the origin of the frame \( \{i\} \) position in frame \( \{i-1\} \). DH parameters of the robot manipulator are defined according to the assigned frames that shown in Fig.2 and they are listed in Table 1[12].

3. Kinematic Jacobian

Jacobian gives the relationship between the joints velocities and the corresponding end effector linear and angular velocities. The end effector linear and angular velocities can be defined as:

![Figure (2): The Attached Coordinate Frame Systems.](image)

<table>
<thead>
<tr>
<th>( i )</th>
<th>( a_i ) (mm)</th>
<th>( \alpha_i ) (deg)</th>
<th>( d_i ) (mm)</th>
<th>( \theta_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>90</td>
<td>201</td>
<td>( q_1 )</td>
</tr>
<tr>
<td>2</td>
<td>390</td>
<td>0</td>
<td>65</td>
<td>( q_2 )</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>90</td>
<td>-65</td>
<td>( q_3 )</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>-90</td>
<td>380</td>
<td>( q_4 )</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>90</td>
<td>0</td>
<td>( q_5 )</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>81</td>
<td>( q_6 )</td>
</tr>
</tbody>
</table>
\[
\begin{bmatrix}
    v_e \\
    \omega_e
\end{bmatrix}
= \begin{bmatrix}
    J_p \\
    J_o
\end{bmatrix}
\begin{bmatrix}
    \dot{q} \\
    \dot{\omega}
\end{bmatrix}
\text{...(1)}
\]

Where:
- \( v_e \) : (3x1) matrix represents the end effector linear velocity in cartesian space.
- \( \omega_e \) : (3x1) matrix represents the end effector angular velocity in cartesian space.
- \( J_p \) : (3xn) jacobian matrix relates the end effector linear velocity to joints velocities.
- \( J_o \) : (3xn) jacobian matrix relates the end effector angular velocity to joints velocities.

The geometry \( i^{th} \) column of jacobian matrix for revolute joint is: [13]
\[
J_{n}\times(p_e - p_{i-1}) \\
J_{o}
\text{...(2)}
\]

Where:
- \( \xi_i \) : unit vector in z-direction is given by third column of the rotation matrix \( ^0R_{i-1} \).
- \( p_e \) : end effector position vector is given by the first three elements of fourth column of transformation matrix \( ^0T \).
- \( p_{i-1} \) : is given by the first three elements of the fourth column of transformation matrix \( ^{i-1}T \).

4. Robot Equations of Motion Derivation

The dynamic model of the manipulator provides a description of the relationship between the joint actuator torque and motion of structure, by Euler-Lagrange formulation the equations of motion can be derived in a systematic way where the lagrange analysis can be defined as:
\[
L = K - U 
\text{...(3)}
\]
Where:
- \( K \) : Total kinetic energy.
- \( U \) : Total potential energy.

And the equations of motion are expressed by:
\[
d \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \xi_i, i=1,\ldots,6 
\text{...(4)}
\]
Where:
- \( \xi_i \) : The generalized coordinate which is represented by the joint variable.
- \( \xi_i \) : The generalized moments this is represented by the joint actuator torque.

The kinetic energy of \( i^{th} \) link and of \( j^{th} \) motor is:
\[
K_i(t) = \frac{1}{2} \sum_{j=1}^{n} b_{ij}(q) \dot{q}_j \sum_{j=1}^{n} b_{ij}(q) \dot{q}_j + \frac{1}{2} \sum_{j=1}^{n} \dot{q}_j \sum_{j=1}^{n} \dot{q}_j 
\text{...(5)}
\]
\[
K_i(m) = \frac{1}{2} \sum_{j=1}^{n} m_{ij}(q) \dot{q}_j \sum_{j=1}^{n} m_{ij}(q) \dot{q}_j + \frac{1}{2} \sum_{j=1}^{n} \dot{q}_j \sum_{j=1}^{n} \dot{q}_j 
\text{...(6)}
\]

Where:
- \( v \) and \( \omega \) : linear and angular velocities of center of masses.
- \( 0_l^T \) : Mass moment in base frame coordinates.

Since
\[
0_v = J_p \dot{q} \text{ and } 0_l = J_p \dot{q} + R \dot{q} 
\text{[13]}
\]

Then for the assigned robot manipulator the total kinetic energy is:
\[
K = \sum_{i=1}^{3} K_i(t) + \sum_{i=2}^{6} K_i(m) 
\text{...(7)}
\]

And the potential energy of \( i^{th} \) link and \( j^{th} \) actuator is:
\[
U_i(t) = -m_i \sum_{j=1}^{n} b_{ij}(q) \dot{q}_j \sum_{j=1}^{n} b_{ij}(q) \dot{q}_j 
\text{...(8)}
\]
\[
U_i(m) = -m_i \sum_{j=1}^{n} b_{ij}(q) \dot{q}_j \sum_{j=1}^{n} b_{ij}(q) \dot{q}_j 
\text{...(9)}
\]

Where:
- \( g \) : gravity acceleration vector in base coordinate frame which is [0 0 -9.81]T.
- \( q_r \) : The distance of \( i^{th} \) link mass center to base coordinates frame.

And the total potential energy for \( n \) link manipulator:
\[
U = \sum_{i=1}^{3} U_i(t) + \sum_{i=2}^{6} U_i(m) 
\text{...(10)}
\]

By noticing that \( U \) doesn’t depend on \( \dot{q} \) then equation (6) can be written as:
\[
\frac{d}{dt} \frac{\partial K}{\partial \dot{q}_i} - \frac{\partial K}{\partial q_i} = \xi_i 
\text{...(11)}
\]

By substituting eqs (11) and (14) into eq. (15), getting that:
\[
\frac{d}{dt} \frac{\partial K}{\partial \dot{q}_i} = \sum_{j=1}^{n} b_{ij}(q) \dot{q}_j + \sum_{j=1}^{n} \dot{q}_j \sum_{j=1}^{n} \dot{q}_j 
\text{...(12)}
\]
\[
\frac{d}{dt} \frac{\partial U_i}{\partial \dot{q}_i} = \sum_{j=1}^{n} b_{ij}(q) \dot{q}_j + \sum_{j=1}^{n} \dot{q}_j \sum_{j=1}^{n} \dot{q}_j 
\text{...(13)}
\]

Also
\[
\frac{\partial K}{\partial \dot{q}_i} = \frac{1}{2} \sum_{j=1}^{n} \sum_{l=1}^{m} \frac{\partial b_{jk}(q)}{\partial \dot{q}_j} \dot{q}_j 
\text{...(14)}
\]

And
\[
\frac{\partial U_i}{\partial \dot{q}_i} = \frac{1}{2} \sum_{j=1}^{n} \sum_{l=1}^{m} \frac{\partial m_{ij}(q)}{\partial \dot{q}_j} \dot{q}_j + \frac{1}{2} \sum_{j=1}^{n} \dot{q}_j \sum_{j=1}^{n} \dot{q}_j 
\text{...(15)}
\]

As a result the equations of motion are:
\[
\sum_{j=1}^{n} h_{ijk}(q) \dot{q}_j + \sum_{j=1}^{n} g_{ijk}(q) \dot{q}_j + G_1(q) = \xi_i 
\text{...(16)}
\]

Where:
\[h_{ijk} = \frac{\partial b_{ij}}{\partial q_k} - \frac{1}{2} \frac{\partial b_{ik}}{\partial q_j} \quad \text{or} \quad h_{ijk} = \frac{1}{2} \left( \frac{\partial b_{ij}}{\partial q_k} + \frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i} \right), \quad b_{ij} = b_{ji}\]

Christoffel symbol first type.

And the final matrix form of equations can be written as:

\[B(q)\ddot{q} + C(q, \dot{q})\dot{q} + F_r + F_c \{\text{sgn}(\dot{q})\} + G^T(q) + J^T_i F_a = \tau \quad \ldots(20)\]

Where:

- \(B(q)\) : Inertia matrix symmetric \((6x6)\).
- \(C(q, \dot{q})\) : Centrifugal and Coriolis forces matrix \((6x6)\).
- \(F_r\) : Viscous friction coefficients, diagonal matrix \((6x6)\).
- \(F_c\) : Coulomb friction (static torque friction), diagonal matrix \((6x6)\).
- \(\text{sgn}(\dot{q})\) : Vector of sign function of joints velocities \((6x1)\).
- \(G\) : Gravitational torques matrix \((1x6)\).
- \(F_e\) : Vector of forces and moments exerted by the end effector on the environment \((6x1)\).
- \(\tau\) : Actuator torque vector \((6x1)\).

As shown in Fig.3, the links and end effector tool characteristics can be written as below.

**Figure (3): Schematic diagram of dynamic model**

Where for links:

\[0_{ri} = 0_T r_i, \quad i=1,2,3\]

\[J_l = \begin{bmatrix} z_0 x(r_{i1} - p_0) & 0 & 0 & 0 & 0 & 0 \\ z_0 y(r_{i2} - p_0) & z_1 x(r_{i2} - p_1) & 0 & 0 & 0 & 0 \\ z_0 z(r_{i3} - p_0) & z_1 y(r_{i3} - p_1) & z_2 x(r_{i3} - p_2) & 0 & 0 & 0 \\ \end{bmatrix}_{6x6}\]

\[J_o = \begin{bmatrix} z_0 & 0 & 0 & 0 & 0 & 0 \\ z_1 & z_2 & z_3 & 0 & 0 & 0 \\ z_4 & z_5 & & & & \end{bmatrix}_{3x6}\]

And for actuators:

\[0_{rm} = 0_T r_m, \quad j=2,3,\ldots,6,\]

\[J_{pm} = \begin{bmatrix} z_0 x(r_{m1} - p_0) & z_1 x(r_{m1} - p_1) & 0 & 0 & 0 & 0 \\ z_0 y(r_{m2} - p_0) & z_1 y(r_{m2} - p_1) & z_2 x(r_{m2} - p_2) & 0 & 0 & 0 \\ z_0 z(r_{m3} - p_0) & z_1 z(r_{m3} - p_1) & z_2 z(r_{m3} - p_2) & z_3 x(r_{m3} - p_3) & 0 & 0 \\ \end{bmatrix}_{6x6}\]

SolidWorks software was used to compute the principle inertia for the compound materials links and for actuators, also reference coordinate system are added in order to measure the distances of masses centers as illustrated in figures 4&5. And the measured
parameters of the manipulator were as listed in tables (4&5).

<table>
<thead>
<tr>
<th>i</th>
<th>( m_i ) (g)</th>
<th>( r_{c_i} ) (mm)</th>
<th>( I_{xx_i} ) (g.mm(^2))</th>
<th>( I_{yy_i} ) (g.mm(^2))</th>
<th>( I_{zz_i} ) (g.mm(^2))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>416.55</td>
<td>([0.0, 20.49, 8.58]T)</td>
<td>1246504.34</td>
<td>1093940.89</td>
<td>531495.71</td>
</tr>
<tr>
<td>2</td>
<td>545.28</td>
<td>([-0.01, 171.76, -3.31]T)</td>
<td>14123528.90</td>
<td>289480.36</td>
<td>14156824.53</td>
</tr>
<tr>
<td>3</td>
<td>276.63</td>
<td>([0.01, 194.95, 0.01]T)</td>
<td>5156951.71</td>
<td>149714.18</td>
<td>5150235.21</td>
</tr>
</tbody>
</table>

Table (5): Robot actuators specifications

<table>
<thead>
<tr>
<th>i</th>
<th>( m_{ai} ) (g)</th>
<th>( r_{a_i} ) (mm)</th>
<th>( I_{xx_{ai}} ) (g.mm(^2))</th>
<th>( I_{yy_{ai}} ) (g.mm(^2))</th>
<th>( I_{zz_{ai}} ) (g.mm(^2))</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1600</td>
<td>([0.3, 5.0, -48.26]T)</td>
<td>2581114.74</td>
<td>2581133.72</td>
<td>693492.85</td>
</tr>
<tr>
<td>3</td>
<td>550</td>
<td>([-0.05, 318.8, -26.13]T)</td>
<td>319734.57</td>
<td>319734.57</td>
<td>132511.82</td>
</tr>
<tr>
<td>4</td>
<td>475.29</td>
<td>([0.0, 6.17, 14.86]T)</td>
<td>1140038.78</td>
<td>983996.69</td>
<td>253437.25</td>
</tr>
<tr>
<td>5</td>
<td>200</td>
<td>([-0.14, 287.03, -4.98]T)</td>
<td>87038.19</td>
<td>87038.19</td>
<td>22060.62</td>
</tr>
<tr>
<td>6</td>
<td>231.15</td>
<td>([-0.05, -18.0, -0.06]T)</td>
<td>134497.11</td>
<td>31671.05</td>
<td>133654.9</td>
</tr>
</tbody>
</table>

Figure (4): The measured inertia and center distance of link 2.

Figure (5): The measured inertia and center distance of actuator 6.
An inverse dynamic method is applied to find the dynamic response of joints variables $q$ & $\dot{q}$. From eq.(22):

$$B(q)\ddot{q} + N = \tau \quad \ldots \text{ (21)}$$

Where

$$N = C(q, \dot{q})[\dot{q}] + F_s[\dot{q}] + F_s[\text{sgn}(\dot{q})] + J_T F_T$$

And from eq.(23)

$$\ddot{q} = B^{-1}(\tau - N) \quad \ldots \text{ (22)}$$

The system is termed open loop system by applying a feed forward torque and closed loop system by applying feedback control torques as illustrated in Fig.6.

Figure (6): Control system I/O

In order to solve the arm dynamic, eq.(24) may be converted to ordinary differential equation system (ODEs). So the state space representation of the robot manipulator multi input multi output (MIMO) system can be written as:

$$\dot{x} = \left[ \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \end{array} \right] + \left[ \begin{array}{c} q \\ q \\ \end{array} \right] + \left[ \begin{array}{c} 0 \\ 0 \\ \end{array} \right] \left[ \begin{array}{c} M^{-1}(\tau - N) \\ \end{array} \right] \quad \ldots \text{ (23)}$$

$$y = [I]_{2n2n} x \quad \ldots \text{ (24)}$$

Where:

$$\{x\} = \{q, \dot{q}\}_{2n1}^T$$

5. Results and discussions:

A Matlab Simulink system is built to embrace the theoretical work as shown in Fig.7.

Figures 8&9 show the joints trajectory for the manipulator collapsing under gravity because the applied torques set to zero and continuous swing due to zero friction (ideal joint) also the 1st joint rotates due to coriolis effect in one direction because the links center masses right a side of base center.

Figure (7): Matlab Simulink system

By applying the simplest form of nonlinear computed torques controller that consider the gravitational force the most effect term at the commanded torques: [14]

$$\tau_i = K_p e_i + K_v \dot{e}_i + G_i \quad \ldots \text{ (25)}$$

termed Propotional Derivative plus Gravity (PD+G) controller. Where :

$$e = (q_d - q) \quad \text{ (error vector)}$$

$$\dot{e} = (\dot{q}_d - \dot{q}) \quad \text{ (error rate vector)}$$

$K_p, K_v$ : diagonal matrix.

let $q_{i1} = \pi/18, q_{i2} = \pi/18, q_{i3}=0, q_{d1} = \pi/9, q_{d2} = \pi/3$ and $q_{d3} = \pi/4$ (initial and desired joints position (rad)).

Considering the actuators are merged with the links to reduce the computations in solving ODE, then the time history of the three links joints for the manual tuning $K_p=200$ & $K_v=75$ were as illustrated in Fig.s (10,11) and table(6), Where the errors completely vanishing in 4 seconds. While for the extracted model with three links and five effective actuators the, step input response, tracking errors and joints torques of six joints were as shown in Fig.s 12, 13 & 14 respectively that show the overall system is respond after 0.2 sec, also by applying the forward analysis, the
cartesian trajectory is shown in Fig.(15); The input and output details are listed in table (7).

**Table (6):** Three joints response due To the PD +G controller torques

<table>
<thead>
<tr>
<th>Time (sec)</th>
<th>q1 (rad)</th>
<th>q1_dot (Rad/s)</th>
<th>q2 (rad)</th>
<th>q2_dot (Rad/s)</th>
<th>q3 (rad)</th>
<th>q3_dot (Rad/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.1745</td>
<td>0</td>
<td>0.1745</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0.3370</td>
<td>0.0322</td>
<td>0.9875</td>
<td>0.1602</td>
<td>0.7318</td>
<td>0.1442</td>
</tr>
<tr>
<td>2</td>
<td>0.3482</td>
<td>0.0022</td>
<td>1.0431</td>
<td>0.0109</td>
<td>0.7818</td>
<td>0.0098</td>
</tr>
<tr>
<td>3</td>
<td>0.3490</td>
<td>0.0002</td>
<td>1.0469</td>
<td>0.0007</td>
<td>0.7852</td>
<td>0.0007</td>
</tr>
<tr>
<td>4</td>
<td>0.3491</td>
<td>0.0000</td>
<td>1.0472</td>
<td>0.0000</td>
<td>0.7854</td>
<td>0.0000</td>
</tr>
<tr>
<td>5</td>
<td>0.3491</td>
<td>0.0000</td>
<td>1.0472</td>
<td>0.0000</td>
<td>0.7854</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

**Figure (10):** Joints response due to the PD + G controller by ode45.  

**Figure (11):** Joints rate due to the PD + Gravity controller by ode45.

**Table (7):** Six joints dynamic characteristics

<table>
<thead>
<tr>
<th></th>
<th>q1(rad)</th>
<th>q2(rad)</th>
<th>q3(rad)</th>
<th>q4(rad)</th>
<th>q5(rad)</th>
<th>q6(rad)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>Pi/18</td>
<td>Pi/3</td>
<td>Pi/6</td>
<td>Pi/4</td>
<td>Pi/3</td>
<td>Pi/9</td>
</tr>
<tr>
<td>desired</td>
<td>Pi/6</td>
<td>Pi/6</td>
<td>Pi/18</td>
<td>Pi/9</td>
<td>Pi/6</td>
<td>Pi/18</td>
</tr>
<tr>
<td>Kp</td>
<td>400</td>
<td>400</td>
<td>400</td>
<td>400</td>
<td>400</td>
<td>400</td>
</tr>
<tr>
<td>Kv</td>
<td>15</td>
<td>25</td>
<td>25</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>

**Figure (12):** Six joints angles response by Simulink  

**Figure (13):** Six joints tracking errors by Simulink  

**Figure (14):** Six joints required torques  

**Figure (15):** End Effector trajectory by Simulink
Fig. 16 shows the disturbance effect greater than friction effect on the tracking trajectory controller, where the external force 250 g is applied in vertical z direction that directly affects joint 2, 3 and 5 and the assumed viscose friction coefficient was 0.02. So it is too important to design a robust controller in order to cancel the combined effect of internal friction and external disturbances.

![Figure (16): friction and disturbance effect](image)

By modifying the commanded torque law in eq.(27) to:

$$\tau_i = Kp \, e_i + Kv \, \dot{e}_i + N_i$$  \hspace{1cm} \text{(26)}

The settling errors back to $3.57\text{E}-6$, $2.09\text{E}-7$, -3.63$\text{E}$-6, 8.84$\text{E}$-6, -5.39$\text{E}$-8 and -4.39$\text{E}$-5 (deg) for joints one to six respectively.

### 6. Conclusions:

In this work, the tool orientation (i.e. $q_4, q_5$ & $q_6$) is neglected at first experience of controller considering the actuators as part of links using Matlab Ode45 solver to avoid memory overflow and the solver was time consuming, whereas Simulink was used for dynamic simulation of the complicated system. The results show the activity of nonlinear controller to track the step input trajectory. $Kp$ and $Kv$ gains were tuned manual to achieve a suitable response with minimum overshoot, the values are differs for each case because the variations between the two models. As the model close to reality, as the controller perform better by reducing the tracking errors tend to zero.

The disturbance forces have more effect than the internal frictions thus a need rise to modify the controller to cancel the perturbations. PD+N controller works fine for that purpose. The presented approach can be applicable to solve the dynamic problem of other n-link robot manipulators to achieve a suitable solution for tracking trajectories taking in account friction effect and end effector external forces.

### 7. References:


