

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Micro-Robot Management

Wael A. Al-Tabey

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/39240>

1. Introduction

Micro-Robots have been and continue to be essential components of medical field. Micro-Robots are used in medical field to surgical applications. Knowledge of the kinematics, dynamics and trajectory planning of these Micro-Robots is most important for their design and control. MATLAB is a modern tool that has transformed the mathematical calculations methods because MATLAB not only provides numerical calculations but also facilitates analytical calculations using the computer. The present textbook uses MATLAB as a tool to solve problems from micro-robots. The intent is to show the convenience of MATLAB for micro-robots analysis. Using example problems the MATLAB syntax will be demonstrated. MATLAB is very useful in the process of deriving solutions for any problem in micro-robots. The chapter includes a most problem of micro-robots for surgical applications that are being solved using MATLAB. The programs are available at the end of this chapter.

Robots are widely used in medical field for getting minimally invasive surgery efficiently and accurately. Minimally invasive surgery is an innovative approach that allows reducing patient trauma, postoperative pain and recovery time [1]. The kinematic and dynamic analysis of the robot, for any applications whether in medical field or another, are very important. The direct aim is to properly select the workspace and the actuator size. Frumento [1] designed a minimally invasive robot for heart surgery. They concentrated mainly on the kinematic analysis and the workspace of the robot. Tsai and Hsu [3] investigated a parallel surgical robot having six degrees of freedom. They studied the kinematics only, to obtain the workspace of surgical robot and control it using Fuzzy Logic control. Miller and Christensen [4] analyzed the dynamics of the Multi-rigid-body robot using Newton's second law of motion and used the results to design the controller. Featherstone and Orin [5] investigated the robot dynamics and used Newton-Euler technique to obtain the equation and algorithms of robot motion. Wang, et al [6] designed the dimensional synthesis of 6-DOF Micro-surgery manipulator (Micro Hand). They studied the kinematics and the workspace of the manipulator they considered by an Optimal design

method. Alici and Shirinzadeh [7] obtained the singularity loci of parallel manipulators exemplified by a 3-DOF spherical parallel manipulator. They utilized the inverse kinematics, and Jacobian matrices to obtain the velocity equation of the actuator and the end effector. The determinants of the manipulator Jacobian matrices were evaluated for a specified set of geometric parameters. Finally, the singularity loci of the manipulator in Cartesian space were generated. Ben-Horin, et al [8] analyzed the kinematics and dynamics of parallel robot consisting of three plenary actuated links [6-DOF parallel manipulator]. They investigated by direct kinematic analysis and obtained the dynamics equations and algorithms using Newton's second law and presented the system performance. Bonnifait and Garcia [9] studied the 6-DOF dynamic localization of an outdoor mobile robot. They obtained the robot dynamics equations and algorithms using the numerical method (Taylor method) and used the azimuth and elevation angles of known landmarks, collected by a rotating linear camera to obtain the simulation results of the robot. They finally presented and analyzed the results of real experiments, performed with an outdoor mobile robot. Abdellatif and Heimann [10] investigated the inverse dynamics equations of 6-DOF fully parallel manipulators using the Lagrangian formalism. With respect to a proposed set of generalized coordinates and velocities they obtained the final form with respect to the robot's active coordinates. Attention was paid to the transformation of the sub chains dynamics. Finally, a systematic study of the resulting computational effort was presented and discussed with respect to results of other methods and approaches of other researches. Zhu, et al [11] investigated the kinematic and dynamic modeling for a newly developed parallel robot with the proposed Tau configuration and used the inverse kinematics method to obtain the kinematic equations of end effector of 3-DOF parallel robot. The dynamic modeling of 3-DOF parallel robot was derived by analytical solutions, which were verified by both numerical simulation and actual experiments. This analytical approach enabled the real-time control of this parallel robot with high positioning accuracy. Gouliaev and Zavrazhina [12] investigated the dynamic and kinematic control of the spatial movements of a flexible multi-link manipulator. They focused on the dynamics of a flexible multi-link manipulator by Euler-Bernoulli method, so that each element was in a compound motion. A technique for the numerical construction of solutions for an essentially non-linear hybrid-type system of constituent equations was proposed. They used the linear control method. Eliodoro and Serna [13] investigated the inverse dynamics of flexible robots. They focused on the new and general technique for solving the inverse dynamics of flexible robots. The proposed method finds the joint torques that must be applied by the actuators to obtain a specified end-effector trajectory. The inverse dynamics of flexible robots are derived by using the Euler-Bernoulli beam theory and Lagrange's equations. The finite element method was utilized to discretize space variables. They finally established the global dynamic equations of the robot. Kinematic constraints were introduced in the dynamic equations by means of a penalty formulation. The system performance was drawn for different tip trajectories.

Martins et al [14] examined an adaptive controller to guide a unicycle-like mobile robot during trajectory tracking. They concentrated, mainly on the kinematic analysis to design

the adaptive controller using Lyapunov theory. The kinematic equations of mobile robot were found by inverse kinematics method to obtain the desired values of the linear and angular velocities, which represented the input to adaptive controller. Valero et al [15] investigated a trajectory planner for 3-DOF industrial robots that have to operate in workspaces with obstacles. They found the workspace modeling analytically using the differential equations of all joints angle and solved them in Cartesian coordinates system. The trajectory planner for 3-DOF industrial robots they found by workspace model using the finite element method for joints and end effector. Integrating all elements, the function of the trajectory planner for 3-DOF industrial robots was obtained. Finally they minimized the objective function of the trajectory planner to attain the minimum time. Geng [16] investigated the dynamics and trajectory planning of a planar flipping robot with two feet and one-leg robot. Geng made a dynamic analysis to obtain the trajectory planning using Lagrange's formulation. Geng presented the simulation results for joints kinetic variables with time. Alessandro and Vanni [17] investigated a technique for optimal trajectory planning of robot manipulators to minimize the jerk. They concentrated on the trajectory planning of robot manipulators. In order to get the optimal trajectory, an objective function composed of two terms was minimized. The first term was proportional to the total execution time, and the second was proportional to the integral of the squared jerk. Finally, they minimized the time of trajectory planning to minimize energy (or actuator effort) and jerk of manipulator joints. They represented the Simulation results for joints kinetic variables and jerk against the minimized time. Pires et al [18] tested a manipulator trajectory planning with multiple objectives and obstacle avoidance (MOEA). It is a non-trivial optimization problem. They concentrated on the trajectory planning of 2-DOF robot manipulators using MOEA method and simulated for several ranges of joint angle. Finally they gave the simulation results for joints kinetic variables with minimized time. Chettibi et al [19] studied the problem of minimum cost trajectory planning for 2-DOF industrial manipulators. They studied the optimal control via direct parameter optimization of joint positions then they concentrated on the trajectory planning of 2-DOF industrial manipulators for six ranges of joint angle. Finally, they represented the simulation results for joints kinetic variables, jerk and torque with minimized time, to obtain the optimum parameters of joint positions and minimum cost trajectory planning. Alessandro and Vanni [20] investigated a method for smooth trajectory planning of robot manipulators. In order to ensure that the resulting trajectory is smooth enough they used the same method used by them in reference. They also presented in their work a new method to obtain the smooth jerk by composing the overall trajectory with respect to other trajectory optimization techniques.

1.1. Introductory remarks on robots for medical field applications

Robots in medicine are recent entry, beginning as generic instrumental aides and aiming at specialized duties once technology sophistication enables effective settings. Several classifications are used, mainly, dressing taxonomy by means of the expected accomplishments [1]:

1. Patients and disabled aid: bed automation, walking assistants, delivery servants, etc.
2. Laboratory support: clinical testers, radiation therapies assistants, etc.
3. Soundness care: pace-makers, health-monitors, drugs dosing up suppliers, etc.
4. Surgery help: surgeon’s servants, remote effectors, autonomous actors, etc.

Moreover, roughly speaking, the example taxonomy distinguishes extra corporeal fixtures, mainly derived out of conventional technologies, from in-body active devices, generally requiring invasive actions, thus, critically dependent on micro-mechanics and nanotechnologies. With focus on surgery robots, four kinds of tasks are, generally, considered:

1. Organ inspection: cerebral probing, laparoscopic monitoring, etc.
2. Organ nursing or repair: internal anastomosis, obstruction relief, etc.
3. Organ removal: cysts excision, lymph node dissection, etc.
4. (Artificial) organ implant: prosthesis insertion, etc.

Surgical robotic systems are commonly classified according to the degree of direct control the surgeon has over the machine [2]. Under this classification there are three principal types of robots Table 1:

1. Autonomous: performs a preoperative plan without any immediate control from the surgeon.
2. Surgical Assist Device: surgeon and robot share control.
3. Teleoperator: function of the robot completely controlled by the surgeon.

Type of system	Examples	Function(s)	Clinical Discipline(s)
Autonomous	Robodoc	Percutaneous renal needle placement	Orthopedic surgery
	PAKY-RCM	Prostatectomy	Urology
	ProBot	Hip surgery	Urology
Surgical Assist	Caspar	Voice controlled telescope	Orthopedic surgery
	AESOP	Stereotactic	Multiple
	NeuroMate	Neurosurgery	Neurosurgery
Teleoperator	Acrobat	Knee arthroplasty	Orthopedic surgery
	PUMA (Programmable Universal Machine for Assembly)	Multiple	Multiple
	Da Vinci	Multiple	Multiple
	Zeus	Multiple	Multiple
	Neurobot	Multiple	Neurosurgery

Table 1. Classification of robotic surgical systems

The systems in study at this present work are Zeus and Da Vinci robotic arms which are usually attached to a patient-side tower structure and consists of two to three arms that control the operative instruments and a separate arm that controls the video endoscope as shown in the Figure 1.



Figure 1. Zeus and Da Vinci robotic arms

Both the Zeus and Da Vinci systems enhance dexterity in several ways. Internal software filters out the natural tremor of a surgeon's hand, which becomes particularly evident under high magnification and problematic when attempting fine maneuvers in very small fields. In addition, the system can scale movements such that large movements of the control grips can be transformed into smaller movements inside the patient. Finally the system group has 7-DOF. The Surgery arm has 6-DOF plus 1-DOF for the tool actuation. Arrangement, the DOF "a" and "b" are respectively the last DOF at the carrier and the 6-DOF robot joints has Roll-Yaw-Pitch motions as shown in the Figures .2, 3 and 4.

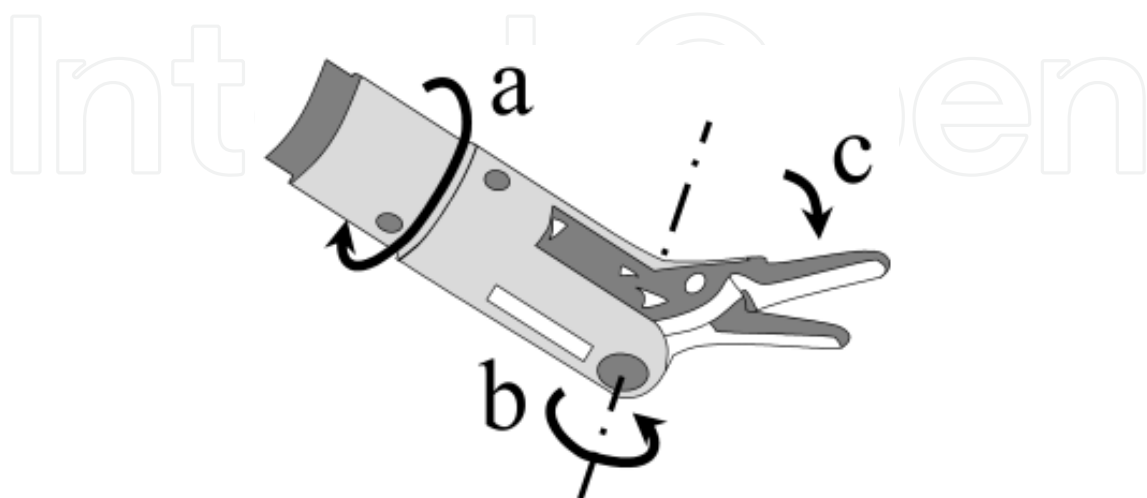


Figure 2. The ZEUS® surgery tools

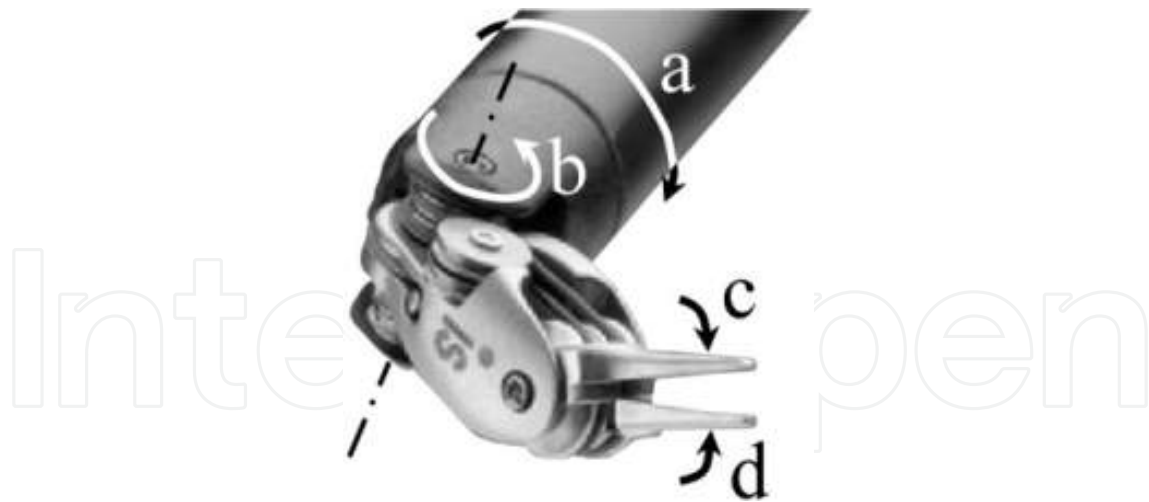


Figure 3. Da Vinci® surgery tools



Figure 4. Roll-Yaw-Pitch motions

2. The kinematic model

2.1. The forward kinematics

The forward kinematics problem is concerned with the relationship between the individual joints of the robot manipulator and the position and orientation of the tool or end effector. Stated more formally, the forward kinematics problem is to determine the position and orientation of the end effector, given the values for the joint variables of the robot. The joint variables are the angles between the links in the case of revolute or rotational joints, and the link extension in the case of prismatic or sliding joints.

The solution is always unique: one given joint position vector always corresponds to only one single end effector pose. The FK problem is not difficult to solve, even for a completely arbitrary kinematic structure.

There are different methods for a forward kinematic analysis: like using straightforward geometry and using transformation matrices.

2.1.1. The Denavit-Hartenberg convention

The D-H modeling rules:

1. Find and identify all joint axes: Z_0 to Z_{n-1} .
2. Establish the base frame. Set base origin anywhere on the Z_0 axis. Choose X_0 and Y_0 conveniently and to form a right hand frame.
3. Locate the origin O_i where the common normal to Z_{i-1} and Z_i intersects Z_i .
If Z_i intersects Z_{i-1} locate O_i at this intersection.
If Z_{i-1} and Z_i are parallel, locate O_i at Joint $(i+1)$.
4. Establish X_i along the common normal between Z_{i-1} and Z_i through O_i , or in the direction normal to the plane $Z_{i-1} - Z_i$ if these axes intersect. See Figure 5.
5. Establish Y_i to form a right hand system
6. Repeat Steps 3 to 5 for $i = 1 : n-1$.
7. Establish the end effector (n) frame: $O_n X_n Y_n Z_n$. Assuming the n^{th} joint is revolute.
Set $K_n = a$ along the direction Z_{n-1} and establish the origin O_n conveniently along Z_n , at center of tool tip. Set $j_n = 0$ in the direction of tool closure (opening) and set $i_n = n$, such that $n = oxa$.
If the tool is not a simple gripper, set X_n and Y_n conveniently to form a right hand frame.
8. Create a table of "Link" parameters: See Figure 5.
Joint Angle θ_i : angle between X_{i-1} and X_i about Z_i .
Link Offset d_i : distance from X_{i-1} and X_i along Z_i .
Link Twist α_i : angle between Z_i and Z_{i+1} about X_i .
Link length a_i : distance between Z_i and Z_{i+1} along X_i .
9. Form HTM matrices $A_1, A_2 \dots A_n$ from the information contained in each row of the LP table by substituting θ, d, α and a into the general model.
10. Build forward kinematic solution: $T_1^n = A_1 * A_2 * \dots * A_n$.

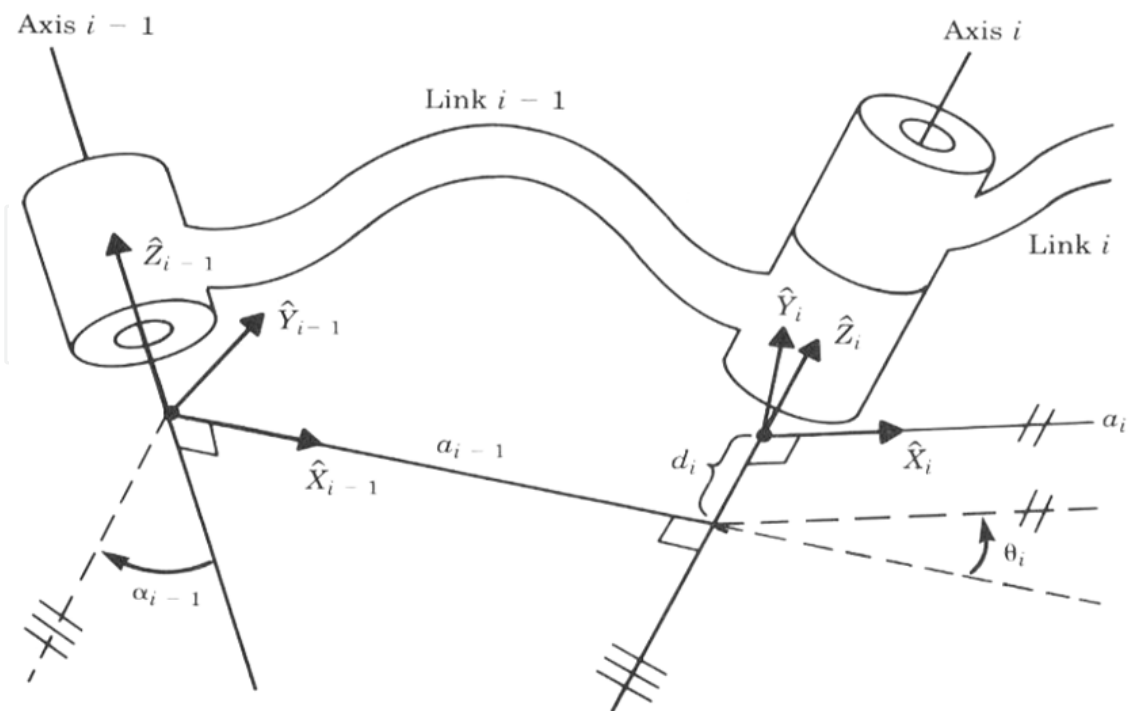


Figure 5. Construction of the link frame

2.2. The kinematics of Zeus and Da Vinci robotic arms

The geometrical model of the surgical robot in this study has 6 degrees of freedom (DOF) and an extra one for tool action. The end-effector has 3 rotations (Roll, Pitch and Yaw) as shown in Figure 6 and the frame assignment of 6-DOF surgical manipulator is represented in Figure 7.

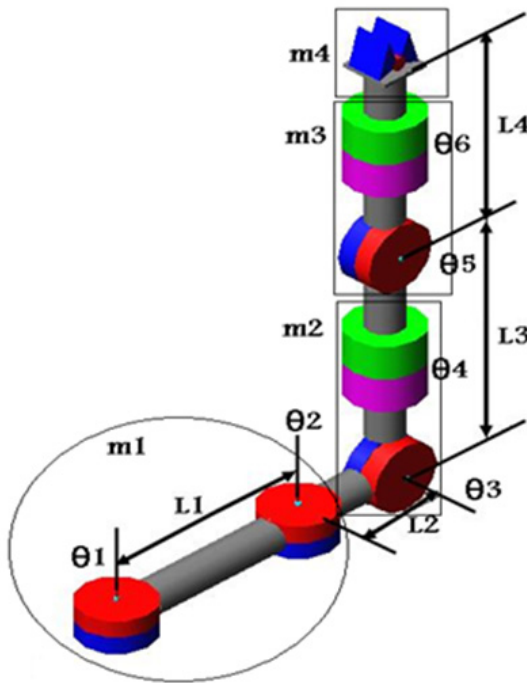


Figure 6. The geometrical model of surgical manipulator

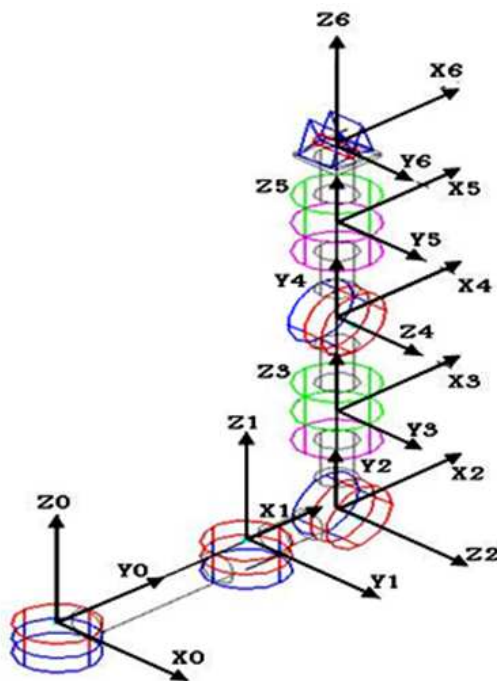


Figure 7. The frame assignment of 6-DOF surgical manipulator

The 6-DOF manipulator kinematic parameters are derived using Denavit Hartenberg formulation shown in Table 2.

Link #	θ_j	d_j	a_j	α_j	T
1	θ_1	0	L_1	0	0T_1
2	θ_2	0	L_2	90	1T_2
3	θ_3	0	0	-90	2T_3
4	θ_4	L_3	0	90	3T_4
5	θ_5	0	0	-90	4T_5
6	θ_6	L_4	0	0	5T_6

Table 2. Full mobility robotic tool: geometry and link parameters

The flowchart representing the sequence of generating the MATLAB CODE of the link transformations matrix is shown in Figure 8.

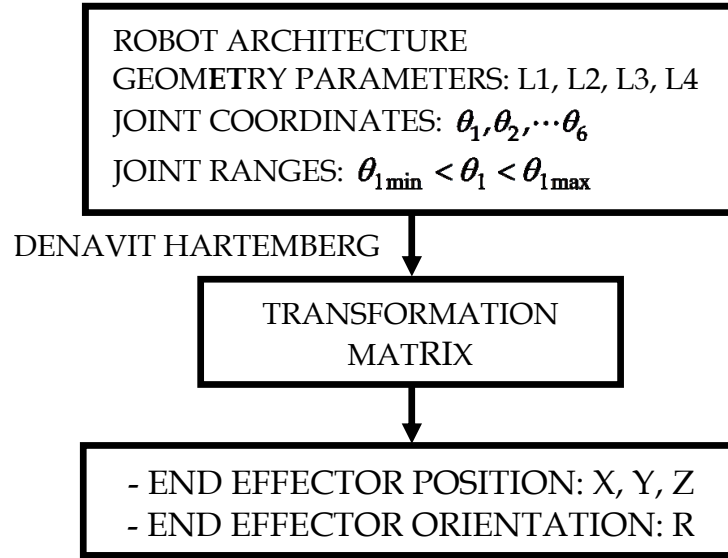


Figure 8. The flowchart of the link transformations matrix

The general form of the Homogeneous Transformation Matrix is:

$${}^{j-1}T_j = \begin{bmatrix} \cos \theta_j & -\sin \theta_j \cos \alpha_j & \sin \theta_j \sin \alpha_j & a_j \cos \theta_j \\ \sin \theta_j & \cos \theta_j \cos \alpha_j & -\cos \theta_j \sin \alpha_j & a_j \sin \theta_j \\ 0 & \sin \alpha_j & \cos \alpha_j & d_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

The link transformations matrix can be given as:

$${}^0T_1 = \begin{bmatrix} C_1 & -S_1 & 0 & L_1 C_1 \\ S_1 & C_1 & 0 & L_1 S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^1T_2 = \begin{bmatrix} C_2 & 0 & S_2 & L_2 C_2 \\ S_2 & 0 & -C_2 & L_2 S_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^2T_3 = \begin{bmatrix} C_3 & 0 & -S_3 & 0 \\ S_3 & 0 & C_3 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3T_4 = \begin{bmatrix} C_4 & 0 & -S_4 & 0 \\ S_4 & 0 & C_4 & 0 \\ 0 & -1 & 0 & L_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^4T_5 = \begin{bmatrix} C_5 & 0 & -S_5 & 0 \\ S_5 & 0 & C_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^5T_6 = \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & L_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The kinematics equations of the end effectors are manipulated using MATLAB symbolic Toolbox were as follows:

$$\begin{bmatrix} P_X \\ P_Y \\ P_Z \\ Z \end{bmatrix} = \begin{bmatrix} ((-(C_1*C_2-S_1*S_2)*C_3*C_4-(C_1*S_2-S_1*C_2)*S_4)*S_5+ \\ (C_1*C_2-S_1*S_2)*S_3*C_5)*L_4-(C_1*C_2-S_1*S_2)*S_3*L_3+ \\ C_1*L_2*C_2-S_1*L_2*S_2+L_1*C_1 \\ \\ ((-(C_1*S_2+S_1*C_2)*C_3*C_4-(C_1*C_2-S_1*S_2)*S_4)*S_5+ \\ (C_1*S_2+S_1*C_2)*S_3*C_5)*L_4-(C_1*S_2+S_1*C_2)*S_3*L_3+ \\ S_1*L_2*C_2+C_1*L_2*S_2+L_1*S_1 \\ \\ (-S_3*C_4*S_5-C_3*C_5)*L_4+C_3*L_3 \\ \\ 1 \end{bmatrix}$$

2.3. The robot workspace

The workspace of a robot can be defined as the set of points that are reachable by the manipulator (with fixed base). Roughly speaking the workspace is the volume of space which the end effector of the robot can reach. Both shape and total volume are important. Workspace is also called work volume or work envelope.

The workspace depends on the characteristics of the manipulator; physical configurations, size, number of axes, the robot mounted position (overhead gantry, wall-mounted, floor mounted, on tracks, etc), limits of arm and joint configurations. The addition of an end effector can move or offset the entire work volume.

The kinematics design of a manipulator can tailor the workspace to some extent to the operational requirements of the robot.

Some robots will have unusable spaces such as dead zones, singular poses, and wrist-wrap poses inside of the boundaries of their reach. Elbow manipulators tend to have a wider volume of workspace.

1. Dexterous workspace: This is the volume of space which the end-effector of the manipulator can reach with all orientations.
2. Reachable workspace: This is the volume of space which the end-effector of the manipulator can reach with at least one orientation.

The dexterous workspace is obviously a subset of the reachable workspace.

2.3.1. The workspace calculation

The workspace may be found mathematically by writing equations that define the robot's links and joints and including their limitations, such as ranges of motions for each joint. Alternatively, the workspace may be found empirically, by moving each joint through each range of motion and combining all the space in can reach and subtracting what it cannot reach.

The workspace of the surgical manipulator can be represented by solving the inverse kinematic equations and taking into consideration all the physical limits of the joints. Figure 9 represents the flowchart showing the sequence of generating the three dimensional workspace of the robot and end-effector to be manipulated using the MATLAB symbolic Toolbox.

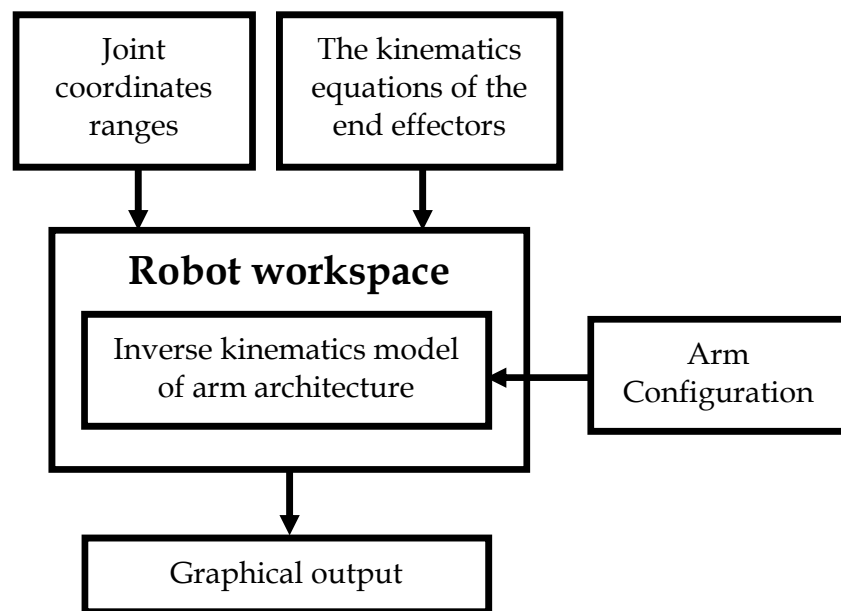


Figure 9. The flowchart of The Robot workspace

Table 3 represent the physical limits of the six joints while Figure10 represent the workspace of end-effector respectively.

Link #	1	2	3	4	5	6
θ_i (degree)	-180°	-90°	0	-180°	-90°	-180°
θ_f (degree)	180°	90°	180°	180°	90°	180°
L (mm)	0	50	15	36	0	39.5

Table 3. Joint coordinates ranges

The workspace of the end-effector depends on the physical limits of the six joints angle, i.e. if the angle range of the robot joints is changes the workspace of the robot changes. So it is important to take into account the accuracy in determining the angle range of the robot joints to get the required workspace which covers the work area.

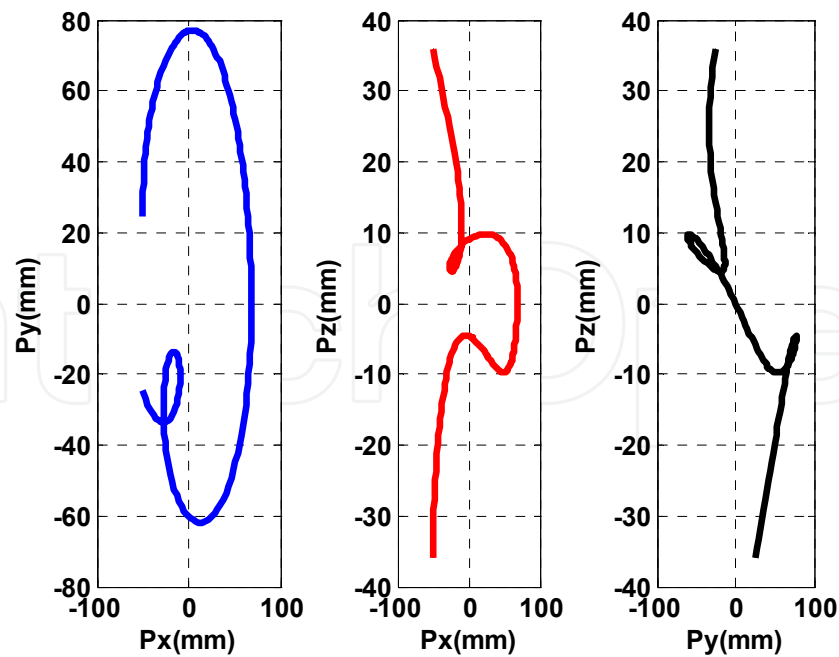


Figure 10. The work space of end effector

2.4. The robot Jacobian

The Jacobian is a representation of the geometry of the elements of mechanism in time. It allows the conversion of differential motions or velocities of individual joints to differential motions or velocities of points of interest. It also relates the individual joint motion to overall mechanism motions. Jacobian is time related; since the values of θ_i vary in time, and the magnitude of the elements of Jacobian vary in time as well.

2.4.1. The differential motions and velocities equations

Differential motions are small movements of robot parts that can be used to derive velocity relationships between different parts of the robot. To find these relations the following steps, are to be considered:

1. Frames relative to a fixed frame.
2. Robot joint relative to a fixed frame.
3. Jacobian matrix.
4. Robot velocity relationship.

2.4.2. The Jacobian equations

Suppose we have a set of equations Y_i in terms of variables X_j

$$Y_i = f_i(X_1, X_2, X_3, \dots, X_j) \quad (2)$$

The differential change in Y_i for a differential change in X_j is:

$$\begin{aligned}
\delta Y_1 &= \frac{\partial f_1}{\partial X_1} \delta X_1 + \frac{\partial f_1}{\partial X_2} \delta X_2 + \dots + \frac{\partial f_1}{\partial X_i} \delta X_i \\
\delta Y_2 &= \frac{\partial f_2}{\partial X_1} \delta X_1 + \frac{\partial f_2}{\partial X_2} \delta X_2 + \dots + \frac{\partial f_2}{\partial X_i} \delta X_i \\
&\vdots \\
&\vdots \\
\delta Y_i &= \frac{\partial f_i}{\partial X_1} \delta X_1 + \frac{\partial f_i}{\partial X_2} \delta X_2 + \dots + \frac{\partial f_i}{\partial X_i} \delta X_i
\end{aligned} \tag{3}$$

2.4.3. The Jacobian matrix

$$[\delta Y_i] = \left[\frac{\partial f_i}{\partial X_i} \right] [\delta X_j] \equiv [D] = [J][D_\theta] \tag{4}$$

2.4.4. The Jacobian relations

The Kinematics equations of the end effectors which are used to calculate the Jacobian matrix of the robot are:

$$P_X = f_1 = ((-C_1 C_2 - S_1 S_2) C_3 C_4 - (-C_1 S_2 - S_1 C_2) S_4) S_5 + (C_1 C_2 - S_1 S_2) S_3 C_5 L_4 - (C_1 C_2 - S_1 S_2) S_3 L_3 + C_1 L_2 C_2 - S_1 L_2 S_2 + L_1 C_1$$

$$P_Y = f_2 = ((-C_1 S_2 + S_1 C_2) C_3 C_4 - (C_1 C_2 + S_1 S_2) S_4) S_5 + (C_1 S_2 + S_1 C_2) S_3 C_5 L_4 - (C_1 S_2 + S_1 C_2) S_3 L_3 + S_1 L_2 C_2 + C_1 L_2 S_2 + L_1 S_1$$

$$P_Z = f_3 = (-S_3 C_4 S_5 - C_3 C_5) L_4 + C_3 L_3$$

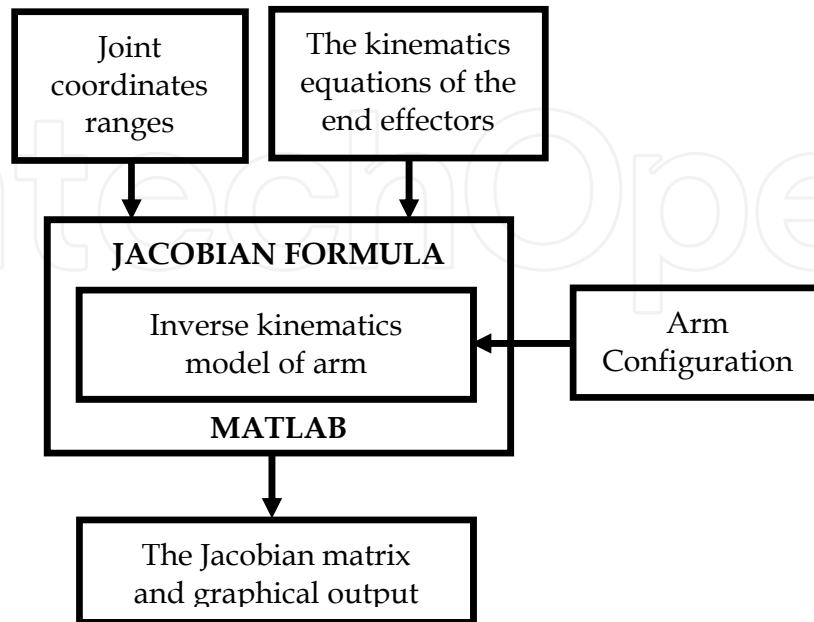


Figure 11. The flowchart of the Jacobian matrix

The Jacobian matrix of the robot can be calculated by MATLAB symbolic Toolbox using the kinematics equations as shown in the Figure 11.

The differential equations of motion of the end effector for the surgical robot are represented by:

$$dP_x = V_x = (((C_1 * S_2 + S_1 * C_2) * C_3 * C_4 - (-C_1 * C_2 + S_1 * S_2) * S_4) * S_5 + (-C_1 * S_2 - S_1 * C_2) * S_3 * C_5 * L_4 - (-C_1 * S_2 - S_1 * C_2) * S_3 * L_3 - S_1 * L_2 * C_2 - C_1 * L_2 * S_2 - L_1 * S_1) * d\theta_1 + (((C_1 * S_2 + S_1 * C_2) * C_3 * C_4 - (-C_1 * C_2 + S_1 * S_2) * S_4) * S_5 + (-C_1 * S_2 - S_1 * C_2) * S_3 * C_5) * L_4 - (-C_1 * S_2 - S_1 * C_2) * S_3 * L_3 - S_1 * L_2 * C_2 - C_1 * L_2 * S_2) * d\theta_2 + (((C_1 * C_2 - S_1 * S_2) * S_3 * C_4 * S_5 + (C_1 * C_2 - S_1 * S_2) * C_3 * C_5) * L_4 - (C_1 * C_2 - S_1 * S_2) * C_3 * L_3) * d\theta_3 + ((C_1 * C_2 - S_1 * S_2) * C_3 * S_4 - (-C_1 * S_2 - S_1 * C_2) * C_4) * S_5 * L_4 * d\theta_4 + (((-C_1 * C_2 + S_1 * S_2) * C_3 * C_4 - (-C_1 * S_2 - S_1 * C_2) * S_4) * C_5 - (C_1 * C_2 - S_1 * S_2) * S_3 * S_5) * L_4 * d\theta_5$$

$$dP_y = V_y = (((-C_1 * C_2 + S_1 * S_2) * C_3 * C_4 - (-C_1 * S_2 - S_1 * C_2) * S_4) * S_5 + (C_1 * C_2 - S_1 * S_2) * S_3 * C_5) * L_4 - (C_1 * C_2 - S_1 * S_2) * S_3 * L_3 + C_1 * L_2 * C_2 - S_1 * L_2 * S_2 + L_1 * C_1) * d\theta_1 + (((-C_1 * C_2 + S_1 * S_2) * C_3 * C_4 - (-C_1 * S_2 - S_1 * C_2) * S_4) * S_5 + (C_1 * C_2 - S_1 * S_2) * S_3 * C_5) * L_4 - (C_1 * C_2 - S_1 * S_2) * S_3 * L_3 + C_1 * L_2 * C_2 - S_1 * L_2 * S_2) * d\theta_2 + (((C_1 * S_2 + S_1 * C_2) * S_3 * C_4 * S_5 + (C_1 * S_2 + S_1 * C_2) * C_3 * C_5) * L_4 - (C_1 * S_2 + S_1 * C_2) * C_3 * L_3) * d\theta_3 + ((C_1 * S_2 + S_1 * C_2) * C_3 * S_4 - (C_1 * C_2 - S_1 * S_2) * C_4) * S_5 * L_4 * d\theta_4 + (((-C_1 * S_2 - S_1 * C_2) * C_3 * C_4 - (C_1 * C_2 - S_1 * S_2) * S_4) * C_5 - (C_1 * S_2 + S_1 * C_2) * S_3 * S_5) * L_4 * d\theta_5$$

$$dP_z = V_z = ((-C_3 * C_4 * S_5 + S_3 * C_5) * L_4 - S_3 * L_3) * d\theta_3 + S_3 * S_4 * S_5 * L_4 * d\theta_4 + (-S_3 * C_4 * C_5 + C_3 * S_5) * L_4 * d\theta_5$$

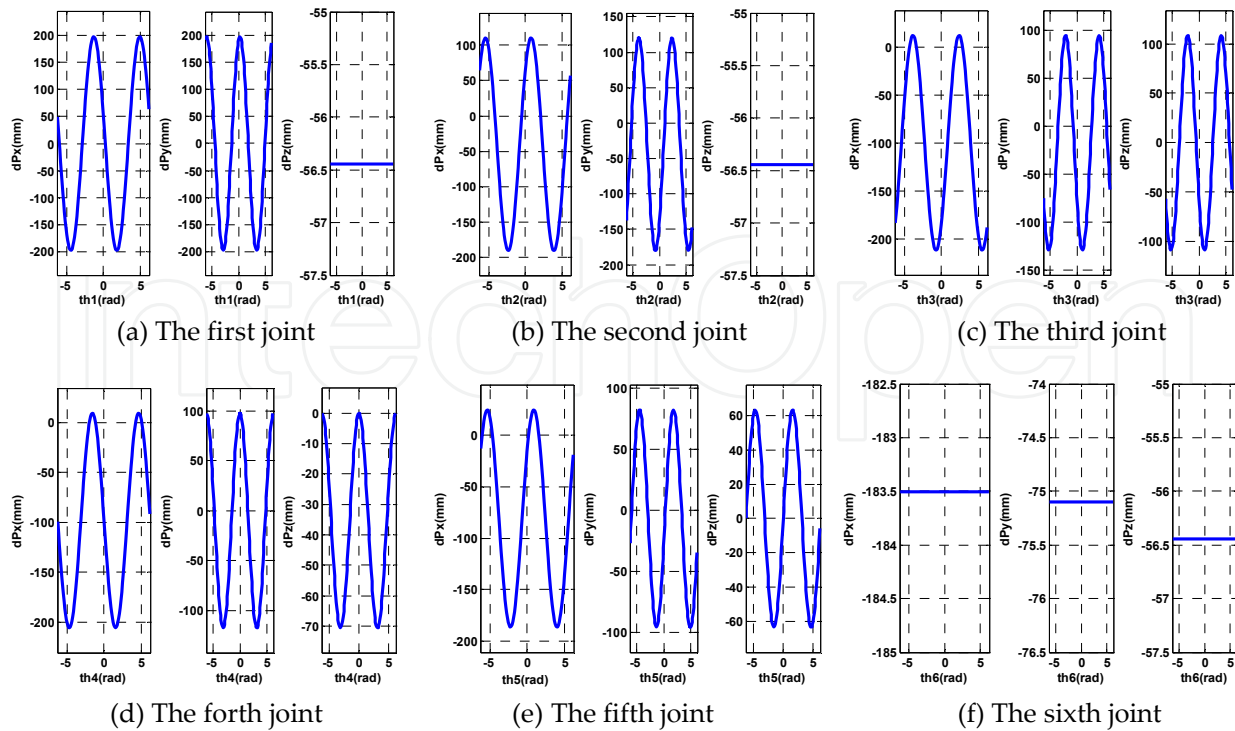


Figure 12. The relation between angle of joints and the end effector differential translation

The previous differential equations of motion of the end effector represent the relation between the magnitude of the elements of Jacobian (the elements of end effector motion) and the joints angle. Figures 12 a, b, c, d, e and f represents the relation between angle of joints and the end effector differential translation. This shows six figures is (a, b, c, d, e and f) for six joints. It is to be noted that in Figures 12-a and 12-b for the first and second joints respectively the relation between angle of joint and the end effector differential translation in the Z direction is constant i.e. the motion of the end effector in the Z direction is not affected by the change in angle of the first and second joints. In the same time the motion of the end effector in all directions (X, Y and Z) is not affected by the change in angle of the six joints because the frame of the end effector is in the same directions of the frame of the joint number six. This can clearly be inferred from the geometrical model and the frame of the robot given in Figures 6 and 7 respectively.

The relations between the magnitude of the elements of Jacobian (the elements of end effector motion at all directions) i.e. dP_x , dP_y and dP_z give of the path of the robot joints as shown in Figures 13 a, b, c, d, e and f.

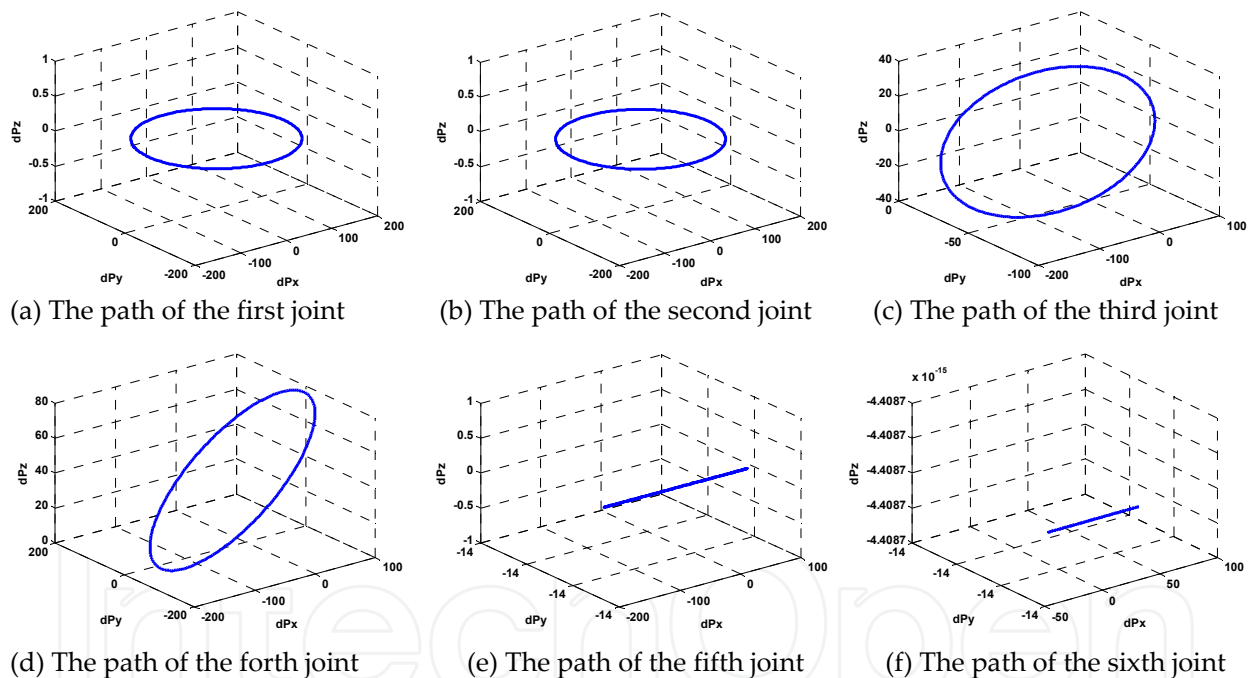


Figure 13. The path of the robot joints

3. The trajectory planning

3.1. Introduction to the trajectory planning

Robot study is divided into two parts; they are the kinematics and dynamics. This means that using the equations of motion of the robot, its position can be determined if the joint variables are known. Path and trajectory planning relates the way a robot is moved from one location to another in controlled manner. In this chapter, a study of the sequence of

movements is to be made to create a controlled movement between motion segments, in straight-line motion, or in sequential motions. Path and trajectory planning requires the use of both kinematics and dynamics of robots. In practice, precise motion requirements are so intensive that approximations are always necessary [23].

What is the different between the path and the trajectory planning?

A Path is defined as a sequence of robot configurations in particular order without regard to timing of these configurations as shown in Figure 14. A trajectory is concerned about when each part of the path must be obtained thus specifying timing (Velocity and Acceleration). Or a trajectory is a spatial position/time curve that usually represents a desired manipulation motion in either link or Cartesian space as shown in Figure 15.

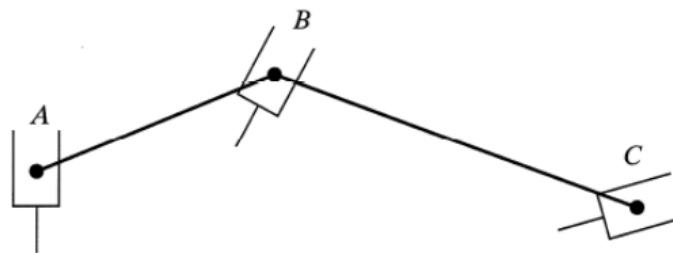


Figure 14. Sequential robot movements in a path [23]

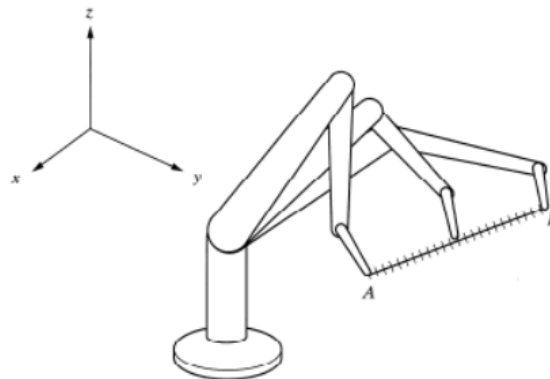


Figure 15. Sequential motions of a robot to follow a straight line [23]

3.2. The methods to calculate trajectory planning

There are four different methods which have been derived to calculate the trajectory planning of the robot using MATLAB code. The trajectory planning in the four methods needs two parameters, namely the joint angle range and the final time required to complete the process. The first one was selected from the workspace needed to complete the process. In any case, this is dependent on the type of surgical applications and can easily be identified. The final time required is a very important parameter to derive the trajectory planning of the robot. As all the results that have been inferred from the orientation, velocity, acceleration and torque of the robot are based on this parameter. If the time increases the velocity required decreases and acceleration i.e. the inertia of the

robot link consequently decreases accidentally irrespective of the methods used. The orientation must be seen carefully examined and the behaviour should be increasing gradually with the time from the initial angle to the final angle. The final time required to complete the process for each method should be optimized and this is called optimal planning. In this work the trajectory planning of the four times is derived and then, comparison of the results at specific joint is made. The proper time which gives satisfactory results is obtained.

3.2.1. Third-Order polynomial trajectory planning

The most common techniques for trajectory planning for industrial robots are the use of polynomial of different orders, such as Cubic and B-splines, linear segments with parabolic blends and the soft motion trajectory [22]. The Linear Segments with Parabolic blends trajectories are faster and more suitable for industrial applications. On the other hand, the higher order polynomials trajectory as well as the soft motion trajectory [24] are easy to design and control especially for the jerk. They are accurate, precise and suitable for medical applications. In this work, the trajectory planning for each joint is designed using third order polynomial as a rest-to-rest manoeuvring where the link starts from rest, accelerates and decelerates at the end of the trajectory. The trajectory is given by [23]:

$$\theta(t) = C_0 + C_1 t + C_2 t^2 + C_3 t^3 \quad (5)$$

In which C_0, C_1, C_2 and C_3 are coefficients to be determined from the initial conditions as follows:

$$\dot{\theta}(t) = C_1 + 2 C_2 t + 3 C_3 t^2 \quad (6)$$

$$\dot{\theta}(t_i) = 0 \text{ and } \dot{\theta}(t_f) = 0 \quad (\text{Rest to Rest manoeuvring})$$

By substituting in to equation (6):

$$\begin{aligned} \dot{\theta}(t_i) &= C_1 = 0 \\ \dot{\theta}(t_f) &= C_1 + 2 C_2 t_f + 3 C_3 t_f^2 = 0 \end{aligned} \quad (7)$$

The initial and final location and orientation of robot are known from:

$$\theta(t_i) = \theta_i \text{ and } \theta(t_f) = \theta_f$$

By substituting in to equation (5):

$$\begin{aligned} \theta(t_i) &= C_0 = \theta_i \\ \theta(t_f) &= C_0 + C_1 t_f + C_2 t_f^2 + C_3 t_f^3 = \theta_f \end{aligned} \quad (8)$$

By solving equations (7), (8) simultaneously:

$$C_2 = 3(\theta_f - \theta_i) / t_f^2 \text{ and } C_3 = 2(\theta_f - \theta_i) / t_f^3$$

3.2.2. Fifth-Order polynomial trajectory planning

The third order trajectory only takes into account starting and end velocities. The equations of the fifth order polynomial takes into account starting and end accelerations. In this case, the total number of boundary conditions are six, allowing a fifth.

The initial and final velocities are zero. (Rest to Rest manoeuvring) and the trajectory is given by [23]:

$$\theta(t) = C_0 + C_1 t + C_2 t^2 + C_3 t^3 + C_4 t^4 + C_5 t^5 \quad (9)$$

In which C_0, C_1, C_2, C_3, C_4 and C_5 are coefficients to be determined from the initial and final conditions as follows:

$$\dot{\theta}(t) = C_1 + 2C_2 t + 3C_3 t^2 + 4C_4 t^3 + 5C_5 t^4 \quad (10)$$

$$\ddot{\theta}(t) = 2C_2 + 6C_3 t + 12C_4 t^2 + 20C_5 t^3 \quad (11)$$

$$\dot{\theta}(t_i) = 0 \text{ and } \dot{\theta}(t_f) = 0 \quad (\text{Rest to Rest manoeuvring})$$

$$\ddot{\theta}(t_i) = \ddot{\theta}_i \text{ and } \ddot{\theta}(t_f) = \ddot{\theta}_f$$

$$C_0 = \theta_i, C_1 = \dot{\theta}_i, C_2 = \frac{\ddot{\theta}_i}{2}, C_3 = \frac{20\theta_f - 20\theta_i - (8\dot{\theta}_f + 12\dot{\theta}_i) t_f - (3\ddot{\theta}_i - \ddot{\theta}_f) t_f^2}{2t_f^3},$$

$$C_4 = \frac{30\theta_f - 30\theta_i + (14\dot{\theta}_f + 16\dot{\theta}_i) t_f + (3\ddot{\theta}_i - 2\ddot{\theta}_f) t_f^2}{2t_f^4} \text{ and } C_5 = \frac{12\theta_f - 12\theta_i - (6\dot{\theta}_f + 6\dot{\theta}_i) t_f + (\ddot{\theta}_i - \ddot{\theta}_f) t_f^2}{2t_f^5}$$

3.2.3. Linear segments with parabolic blends

Linear segments can be blended with parabolic sections at the beginning and the end of the motion segment, creating continuous position and velocity. Acceleration is constant for the parabolic sections, yielding a linear velocity at the common points A and B as shown in the Figure 16.

3.2.3.1. First parabolic blends (t_0 to t_a)

$$\theta(t) = C_0 + C_1 t + \frac{1}{2} C_2 t^2 \quad (12)$$

$$\dot{\theta}(t) = C_1 + C_2 t \text{ and } \ddot{\theta}(t) = C_2$$

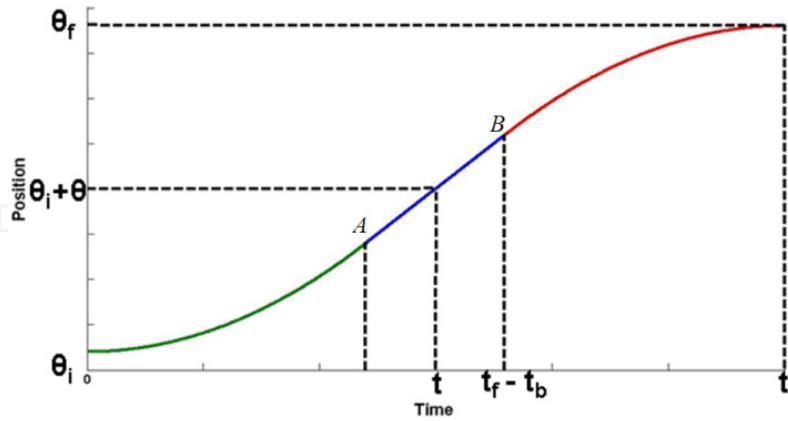


Figure 16. Scheme for linear segments with parabolic blends

The position of the robot at time t_0 is known and using the inverse Kinematic equations of the robot, the joint angles at via points and at the end of the motion can be found.

To blend the motion segments together, the boundary conditions at each point are used to calculate the coefficients of the parabolic segments.

3.2.3.2. Straight line (t_a to t_b)

$$\theta_A = \theta_B + wt, \quad \dot{\theta}_A = \dot{\theta}_B + w \text{ and } \ddot{\theta} = 0$$

3.2.3.3. Second parabolic blends ($t_f - t_b$ to t_f)

$$\theta(t) = C_f + \frac{w}{2t_b}(t_f - t)^2 \quad (13)$$

$$\dot{\theta}(t) = \frac{w}{t_b}(t_f - t) \text{ and } \ddot{\theta}(t) = -\frac{w}{t_b}$$

$$t_b = \frac{(\theta_i - \theta_f + wt_f)}{w}, \quad C_2 = \frac{w}{t_b} \text{ and } w_{\max} = \frac{2(\theta_f - \theta_i)}{t_f}$$

3.2.4. Soft motion trajectory planning

In this method we consider the trajectory planning of points generated by a motion planning technique. The motion planner calculates the trajectory which the end effector must follow in space. However, the temporal characteristics of this movement are independent. One important difference between industrial robotic manipulators and service robot applications is the human interaction, which introduce safety and comfort constraints. In this work, we define soft motions conditions to facilitate this cohabitation. An on-line trajectory planner is proposed here. It generates the necessary references to produce soft

motion and a control loop that guarantees the end effector's motion characteristics (jerk, acceleration, velocity and position) in the Cartesian space, by using quaternion feedback. Two visual feedback control loops are proposed: a visual servoing control loop in a shared position - vision schema and a visual guided loop (which is the general case of soft motion trajectory) are given by:

3.2.4.1. *The motion with a maximum jerk (J_{max})*

$$\theta(t) = \theta_0 + \dot{\theta}_0 t + \frac{1}{2} \ddot{\theta} t^2 + \frac{1}{6} J_{max} t^3 \quad (14)$$

$$\dot{\theta}(t) = \dot{\theta}_0 + \ddot{\theta} t + \frac{1}{2} J_{max} t^2, \quad \ddot{\theta}(t) = \ddot{\theta}_0 + J_{max} \quad \text{and} \quad J(t) = J_{max}$$

3.2.4.2. *The motion with a maximum acceleration (A_{max})*

$$\theta(t) = \theta_0 + \dot{\theta}_0 t + \frac{1}{2} \ddot{\theta}_{max} t^2 \quad (15)$$

$$\dot{\theta}(t) = \dot{\theta}_0 + \ddot{\theta}_{max} t, \quad \ddot{\theta}(t) = \ddot{\theta}_{max} \quad \text{and} \quad J(t) = 0$$

3.2.4.3. *Finally, the equations for the motion with a maximum velocity (V_{max})*

$$\theta(t) = \theta_0 + \dot{\theta}_{max} t \quad (16)$$

$$\dot{\theta}(t) = \dot{\theta}_{max}, \quad \ddot{\theta}(t) = 0 \quad \text{and} \quad J(t) = 0$$

The initial conditions are:

$$\ddot{\theta}(0) = 0, \quad \dot{\theta}(0) = 0, \quad \theta(0) = \theta_0 \quad \text{and} \quad D = \theta_D - \theta_0$$

Where: D is general traversed angular movement.

The final conditions are:

$$\ddot{\theta}(t_f) = 0, \quad \dot{\theta}(t_f) = 0 \quad \text{and} \quad \theta(t_f) = \theta_D$$

We have two limit conditions to obtain the traversed angular movement:

- **Condition (1):**

Where $\dot{\theta}_{max}$ is reached. It means, $\ddot{\theta}_{max}$ is reached too. Then we have to find the traversed angular (Dthr1). The limit times used to calculate (Dthr1) are:

$$T_j = T_{j_{max}}, \quad T_a = T_{a_{max}} \quad \text{and} \quad T_v = 0$$

The angular movement (Dthr1) becomes:

$$Dthr1 = \frac{\ddot{\theta}_{\max} \dot{\theta}_{\max}}{J_{\max}} + \frac{\dot{\theta}_{\max}^2}{\ddot{\theta}_{\max}}$$

- **Condition (2):**

Where only $\ddot{\theta}_{\max}$ is reached. Then we have to find the traversed angular (Dthr2). The limit times used to calculate (Dthr2) are:

$$T_j = T_{j_{\max}}, T_a = 0 \text{ and } T_v = 0$$

The angular movement (Dthr2) becomes:

$$Dthr2 = 2 \frac{\ddot{\theta}_{\max}^3}{J_{\max}^2}, T_{j_{\max}} = \frac{\ddot{\theta}_{\max}}{J_{\max}} \text{ and } T_{a_{\max}} = \left(\frac{\dot{\theta}_{\max}}{\ddot{\theta}_{\max}} \right) - \left(\frac{\ddot{\theta}_{\max}}{J_{\max}} \right)$$

The soft motion trajectory planning of 6-DOF surgical robot is divided in to three cases depending on the maximum Jerk algorithm as:

- **Case (1) (General case):** If $D \geq Dthr1$

$$T_j = T_{j_{\max}}, T_a = T_{a_{\max}} \text{ and } T_v = \frac{D - Dthr1}{\dot{\theta}_{\max}}$$

- **Case (2):** If $Dthr2 > D \geq Dthr2$

$$T_v = 0, T_j = T_{j_{\max}} \text{ and } T_a = \sqrt{\frac{\ddot{\theta}_{\max}^2}{4J_{\max}} + \frac{D}{\ddot{\theta}_{\max}}} - \frac{3\ddot{\theta}_{\max}}{2J_{\max}}$$

- **Case (3):** If $D < Dthr2$

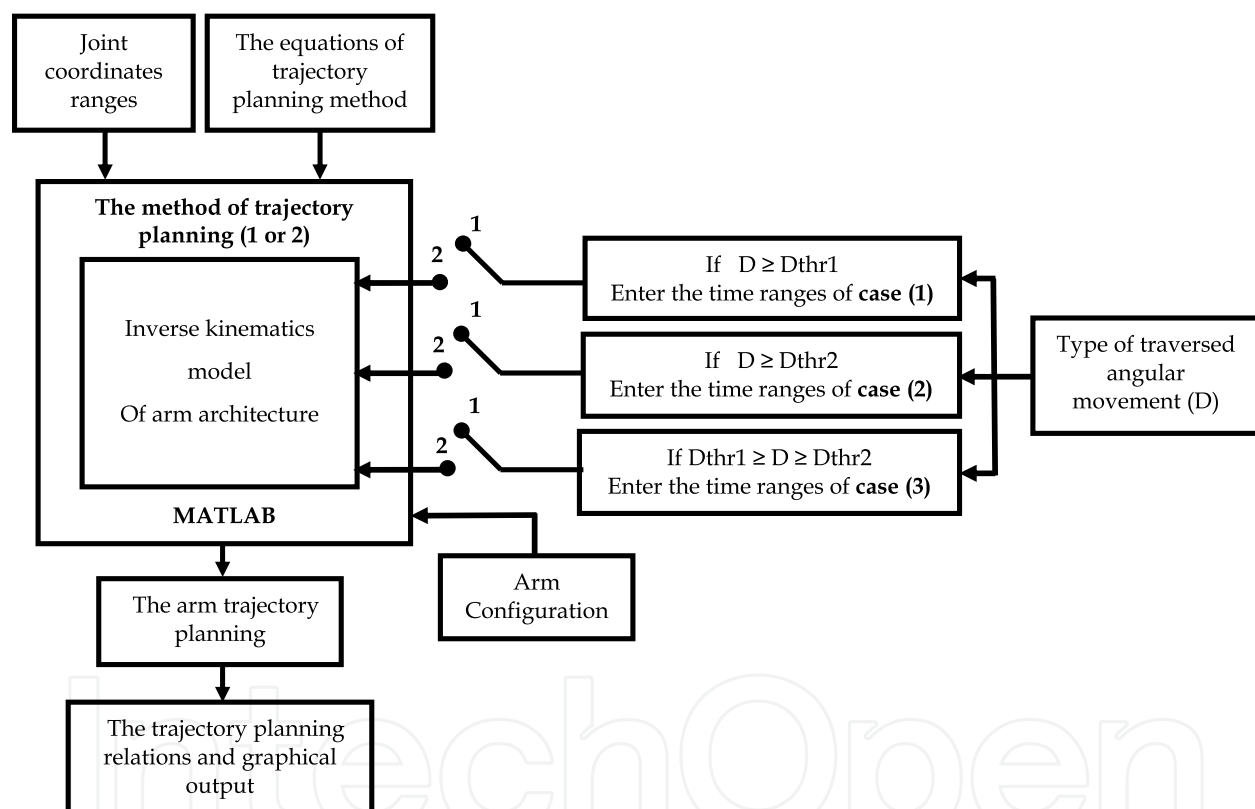
$$T_v = 0, T_a = 0 \text{ and } T_j = \sqrt[3]{\frac{D}{2J_{\max}}}$$

3.3. The robot trajectory planning parameters

In this work, four different methods are applied here to design the joints trajectories third order polynomial, fifth order polynomial, linear segments with parabolic blends and soft motion trajectory. The trajectories have the same initial and final angles and different four duration times (5, 10, 20 and 60 s) are applied to choose the best duration time give the correct dynamic response. As well but they differ in the acceleration and the jerk. After designing the joints trajectories, the hub torques of the robot actuators can be simulated using MATLAB. The parameters of six joints were obtained using inverse kinematics analysis as shown in Table 4.

The flowchart representing the sequence of generating the trajectory planning is shown in the Figure 17.

Link#	1	2	3	4	5	6
θ_i (degree)	-180°	-90°	0	-180°	-90°	-180°
θ_f (degree)	180°	90°	180°	180°	90°	180°
L (mm)	0	50	15	36	0	39.5
$\ddot{\theta}_i$ (degree /s ²)	5	5	5	5	5	5
$\ddot{\theta}_f$ (degree /s ²)	-5	-5	-5	-5	-5	-5
$\dot{\theta}_{\max}$ (degree /s)	135	70	70	135	70	135
$\ddot{\theta}_{\max}$ (degree /s ²)	80	40	40	80	40	80
J_{\max} (degree /s ³)	160	80	80	160	80	160
t_i (s)	0	0	0	0	0	0
t_f (s)	5	5	5	5	5	5

Table 4. Full robotic mobility information used in trajectory planning**Figure 17.** The flowchart of the trajectory planning

Where: Position.1 for first methods of trajectory planning and Position .2 for Soft motion trajectory planning.

4. The dynamic model

4.1. Introduction to the dynamic model of robot

Manipulator dynamics is concerned with the equations of motion, the way in which the manipulator moves in response to torques applied by the actuators, and external forces. The

history and mathematics of the dynamics of serial-link manipulators are well covered in the literature. The equations of motion for an n-axis manipulator are given by [25]:

$$\mathbf{Q} = \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) \quad (17)$$

These equations of motion can be derived by six methods, namely the Newton's second law method, D'Alembert method, Lagrange method, Hemilton method, Lagrange-Euler method and Newton-Euler method. But not all previous methods can be used to derive the equations of motion for the robot subject of this work. That is not all methods can easily derived the equations of motion for a robot having multi-degree of freedom.

4.2. Dynamic equations for multiple-degree of freedom robots

As we have stated, the dynamic equation for two-degree of freedom system is much more complicated than a one-degree of freedom system. Similarly, these equations for multiple-degree of freedom robot are cumbersome and complicated, but can be found by calculating the kinetic and potential energies of the links and joints [25]. For the robot considered in this work the robot has 6-DOF and the most appropriate method to derive the equations of motion is likely to be Lagrange-Euler technique.

4.2.1. Derivation of the equations of motion by Lagrange-Euler technique

The Lagrange-Euler technique is utilized here to calculate the kinetic energy, potential energy and to derive the dynamic equations in symbolic form using the MATLAB symbolic toolbox for the six-degree of freedom robot. The equations of motion are given in a concise form similar to that given in [25].

4.2.1.1. The total kinetic energy

The total kinetic energy of multiple-degree of freedom robot is given in a concise form as:

$$K_i = \frac{1}{2} \sum_{i=1}^n \sum_{p=1}^i \sum_{r=1}^i \text{Trace}(U_{ip} J_i U_{ir} T) \dot{q}_p \dot{q}_r + \frac{1}{2} \sum_{i=1}^n I_{iact} \dot{q}_i^2 \quad (18)$$

$$U_{ip} = \frac{\partial {}^0T_i}{\partial \theta_p} = Q_p {}^0T_i, \quad U_{ir} = \frac{\partial {}^0T_i}{\partial \theta_r} = Q_r {}^0T_i, \quad U_{ir} = U_{ri}, U_{ip} = U_{pi} \text{ and}$$

$$Q_i(\text{revolute}) = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad Q_i(\text{prismatic}) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

But in our case all joints in the robot arm used in surgical applications are considered revolute joints where $Q_p = Q_r = Q$.

4.2.1.2. The potential energy

The potential energy of multiple-degree of freedom robot may be given in a concise form as:

$$P = \sum_{j=1}^n P_i = \sum_{j=1}^n [-m_i g^T ({}^0T_j \bar{X}r_j)] \quad (19)$$

$$g^T = \begin{bmatrix} g_x & g_y & g_z & 0 \end{bmatrix}$$

4.2.1.3. Robot equations of motion

The equations of motion of multiple-degree of freedom robot be given in a compact form as:

$$T_i = \sum_{j=1}^n D_{ij} \ddot{q}_j + I_{i(act)} \ddot{q}_i + \sum_{j=1}^n \sum_{k=1}^n D_{ijk} \dot{q}_j \dot{q}_k + D_i \quad (20)$$

$$D_{ij} = \sum_{p=\max(i,j)}^n \text{Trace}(U_{pj} J_p U_{pi}^T), \quad D_{ijk} = \sum_{p=\max(i,j,k)}^n \text{Trace}(U_{pj} J_p U_{pi}^T),$$

$$D_i = \sum_{p=1}^n -m_p g^T U_{pj} \bar{r}_p, \quad U_{pik} = \frac{\partial U_{pi}}{\partial \theta_k} = U_{pi} Q_k {}^0T_k$$

$$J_i = \begin{bmatrix} \frac{(-I_{XX} + I_{YY} + I_{ZZ})}{2} & I_{XY} & I_{XZ} & m_i \bar{X}_i \\ I_{XY} & \frac{(I_{XX} - I_{YY} + I_{ZZ})}{2} & I_{YZ} & m_i \bar{Y}_i \\ I_{XZ} & I_{YZ} & \frac{(I_{XX} + I_{YY} - I_{ZZ})}{2} & m_i \bar{Z}_i \\ m_i \bar{X}_i & m_i \bar{Y}_i & m_i \bar{Z}_i & m_i \end{bmatrix}$$

Link #	1	2	3	4	5	6
I _{act} (Nmm)	10000	10000	10000	10000	10000	10000
m (kg)	6.055	6.055	6.055	6.055	6.055	6.055
L (mm)	0	50	15	36	0	39.5
g (m/s ²)	9.81	9.81	9.81	9.81	9.81	9.81
R (mm)	75	75	75	75	75	75
I _{xx}	(1/12)*m _i *(3*(R ₂)+(L _i /4)2)					
I _{yy}	(1/12)*m _i *(3*(R ₂)+(L _i /4)2)					
I _{zz}	m _i * R ₂					
I _{xy}	0					
I _{xz}	0					
I _{yz}	0					
X	L _i /2					
Y	0					
Z	0					

Table 5. Full robotic mobility parameters

For the MATLAB simulation, the parameters of the robotic arm are given in table 5. The flowchart for the algorithm employed to calculate the torque history for each actuator based on the derived equations of motion (Equation 20) is shown in the Figure 18. The simulated results for the actuators torques change depending on the method of trajectory planning used to calculate the torque. For example the simulated results for the actuators torques calculated by the Third-Order Polynomial trajectory planning are presented in section 3. It can be seen that the torque history over the selected period of time (5, 10, 20 and 60 s) has considerable fluctuations.

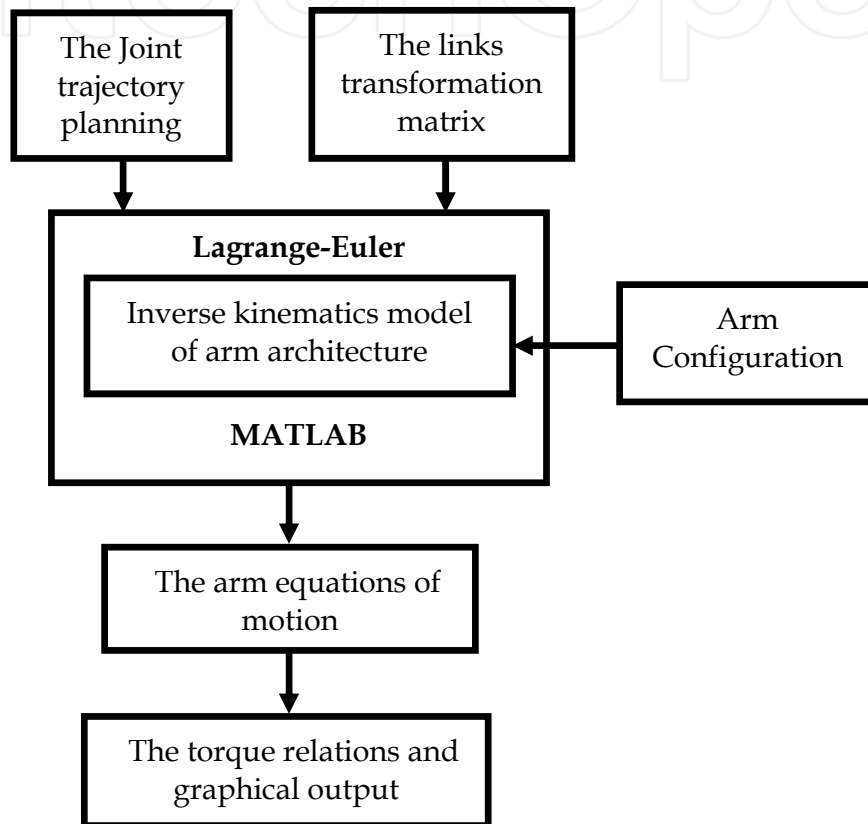


Figure 18. The flowchart of the dynamic model

5. The dynamic response

5.1. Introduction to the dynamic response and dynamic response analysis

The results presented in this chapter are those of chapters IV and V that is the trajectory planning and dynamic modeling respectively as they are very much related to each other. The results are divided into two sections. The first one is for the trajectory planning of the surgical robot which is divided into four parts, each represents a method from four chosen different methods for trajectory planning. As mentioned in chapter IV the joints limits for some of the joints are similar as the joints (1, 4, and 6) and (2, 5). Only the trajectory planning of similar joints will be represented. It should be noted that the results of trajectory planning for all methods were derived using four different times (5, 10, 20 and 60 s) in order to select

the best time which gives the best and smooth orientation, velocity, acceleration (i.e. the inertia of the robot link) and torque. The dynamic modeling results are dependent on the trajectory planning results. This requires the results of the dynamic modeling to be represented after the trajectory planning results i.e. after comparison of the results of the trajectory planning and selecting the best time to be used.

Finally, a comparison of the results is held to choose the best method that gives the smooth set trajectory planning and best performance of the robot under investigation. The simulation results were obtained using MATLAB.

5.2. The trajectory planning analysis

As previously stated that trajectory planning was derived using four methods of the trajectory planning to select the best method that gives the smooth set trajectory planning and best performance of the robot under investigation.

The trajectory planning results is presented in two ways. The first one shows the trajectory planning of the first joint of robot derived using four different times and the second one shows the torque history for the first joint and a comparison between the four times of the first joint of robot in terms of the orientation, velocity, acceleration and the torque history to select the best time that can be used to derive the trajectory planning and torque history for all joints of robot model.

5.2.1. Third order Polynomial trajectory planning

Figures 19 and 20 show comparisons between the orientation, velocity, acceleration, torque and four different time ranges (5, 10, 20 and 60 s) of the first joint of the surgical robot. It is clearly shown in Figure 19 that the orientation behaviour increases gradually with the time from the initial angle to the final angle and as the time increases the velocity required decreases and also the acceleration i.e. the inertia of the robot link decreases. It is also clearly shown in Figure 20 that the original torque history has considerable fluctuations. It is clear that the highest hub torque is for joint one while actuator torque of joint 6 is the lowest.

From Figures 19 and 20 the optimum final time required to complete the process for the robot can be selected. By inspection of Figure 19 for the trajectory planning of the robot we find that for all times the same behaviours i.e. the velocity and acceleration are inversely proportional with the times and the orientation behaviour increases gradually with the time from the initial angle to the final angle. The largest time i.e. 60s to complete the results of the surgical robot can be selected. Since this time has the lowest acceleration i.e. the lowest inertia. In Figure 20 for the torque history in the time 60s we find that the torque decreases with time and the over shooting decreases gradually near the steady state time 60s.

It is seen from Table 4 that the joints angle ranges are very large. So, it is assured that the robot accelerations will not exceed the maximum permissible limits for robot's capabilities if the robot is satisfying the trajectory in one segment.

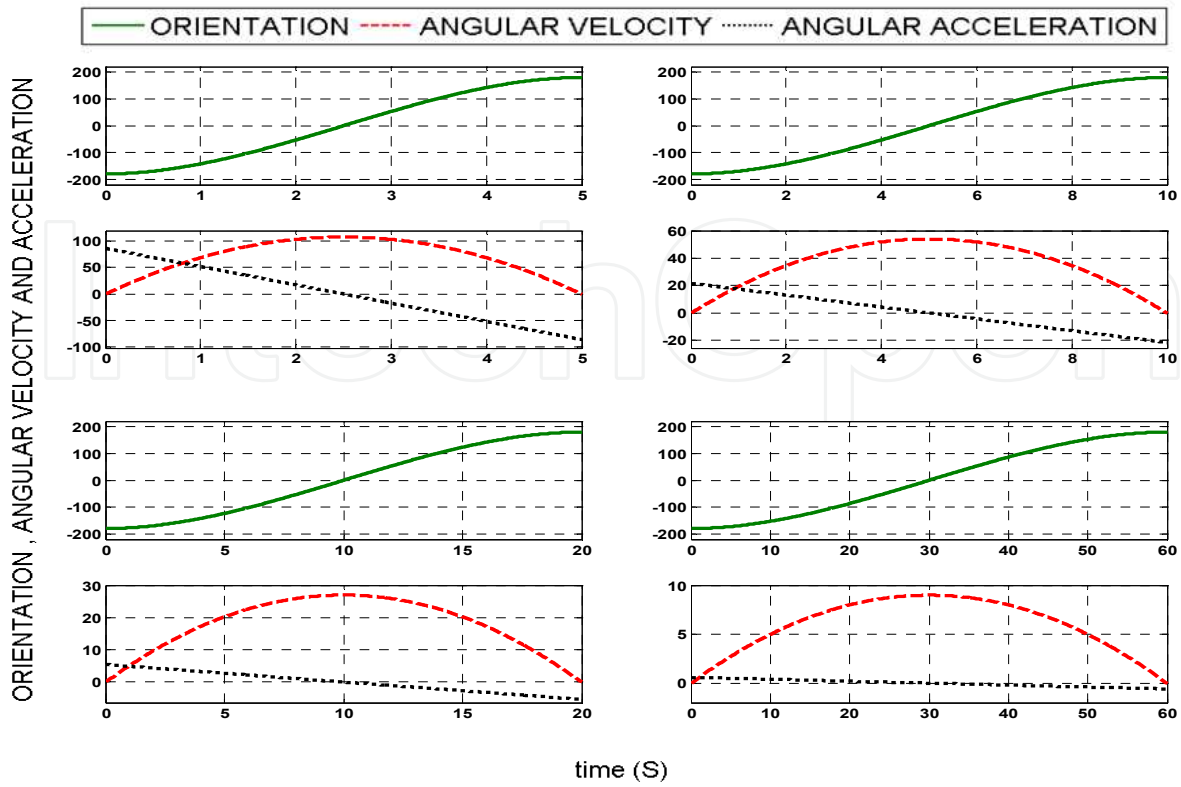


Figure 19. The time comparison of third order trajectory planning of the first joint

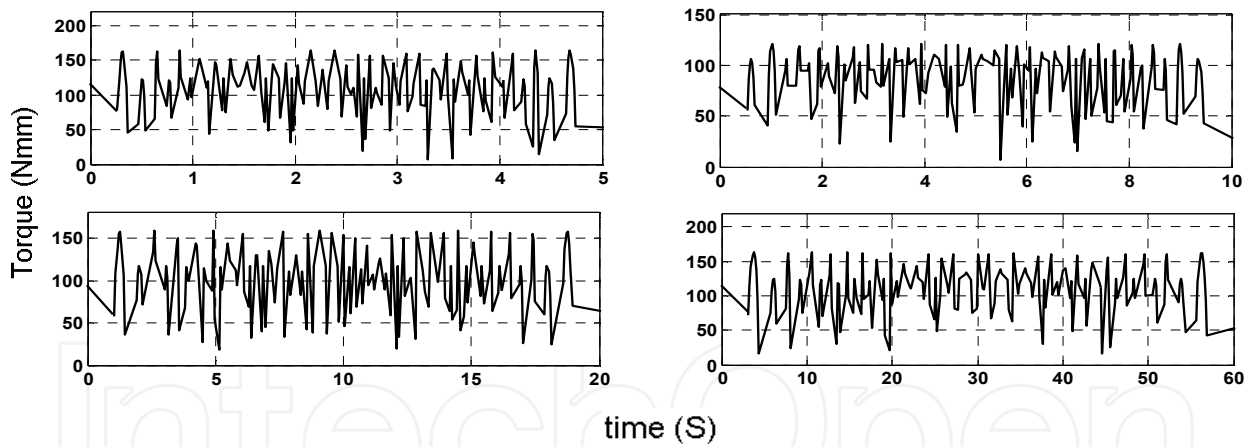
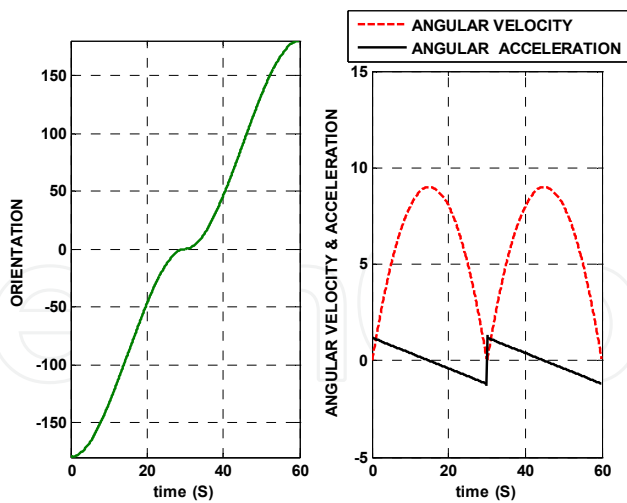


Figure 20. The comparison between the time and torque of the first joint using third order trajectory planning

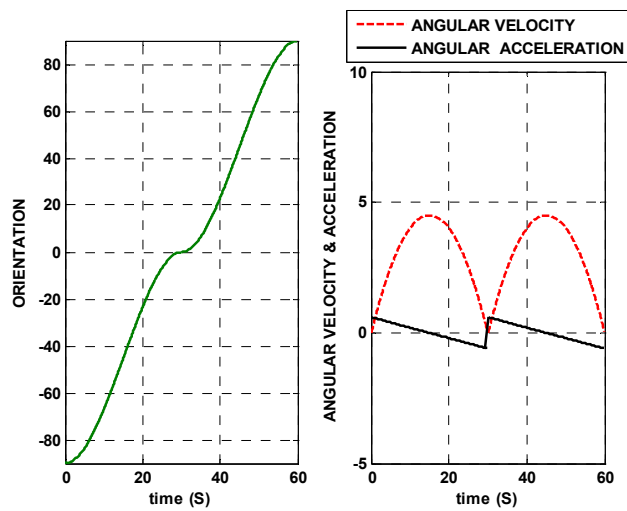
For Third-order Polynomial trajectory planning the maximum accelerations for robot's capabilities is given by:

$$\ddot{\theta}_{\max} = \left| \frac{6(\theta_f - \theta_i)}{(t_f - t_i)^2} \right| \quad (21)$$

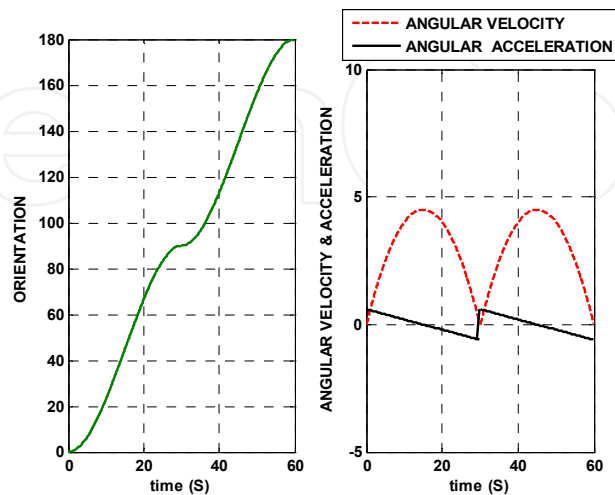
From Figure 19 the robot acceleration needed at the beginning of the motion is 0.62 (degree/s²) as well as -0.62 (degree/s²) deceleration at the end of the motion.



(a) The first, fourth and sixth joints



(b) The second and fifth joints



(c) The third joint

Figure 21. The orientation, angular velocity and angular acceleration for six joint micro-robot using Third order Polynomial trajectory planning

But from the equation 6.1 the maximum permissible accelerations for this robot joints is 0.6 (degree/s²). So to be ensured that robot joint accelerations will not exceed the maximum accelerations for robot's capabilities the robot should satisfy the trajectory planning in two or more segments.

Figures 21a, b and c show the modified trajectories for the six joints, after dividing the trajectory in to two segments.

The joints angle ranges of the initial and final angles are large compared to the values of the velocity and accelerations. So it is more suitable to represent the angle-Time relation and velocity and acceleration – Time relations in two separate figures.

5.2.2. Fifth order Polynomial trajectory planning

Figures 22 and 23 show comparisons between the orientation, velocity, acceleration, and torque for four different time ranges (5, 10, 20 and 60 s) of the first joint of the robot. As is clearly shown in Figure 22 the orientation behaviour increases gradually with time from the initial angle to the final angle for the times (5, 10 and 20 s) only. But for the time 60s the orientation increases gradually from the initial angle to 40 degree then decreases to -40 degree and then increases reaching the final angle. And if the time increases the velocity required decreases. This is similar as to what is shown in Figure 23 which presents the original torque history that has considerable fluctuations. It is clear that the highest hub torque is for joint one while actuator torque of joint 6 is the lowest.

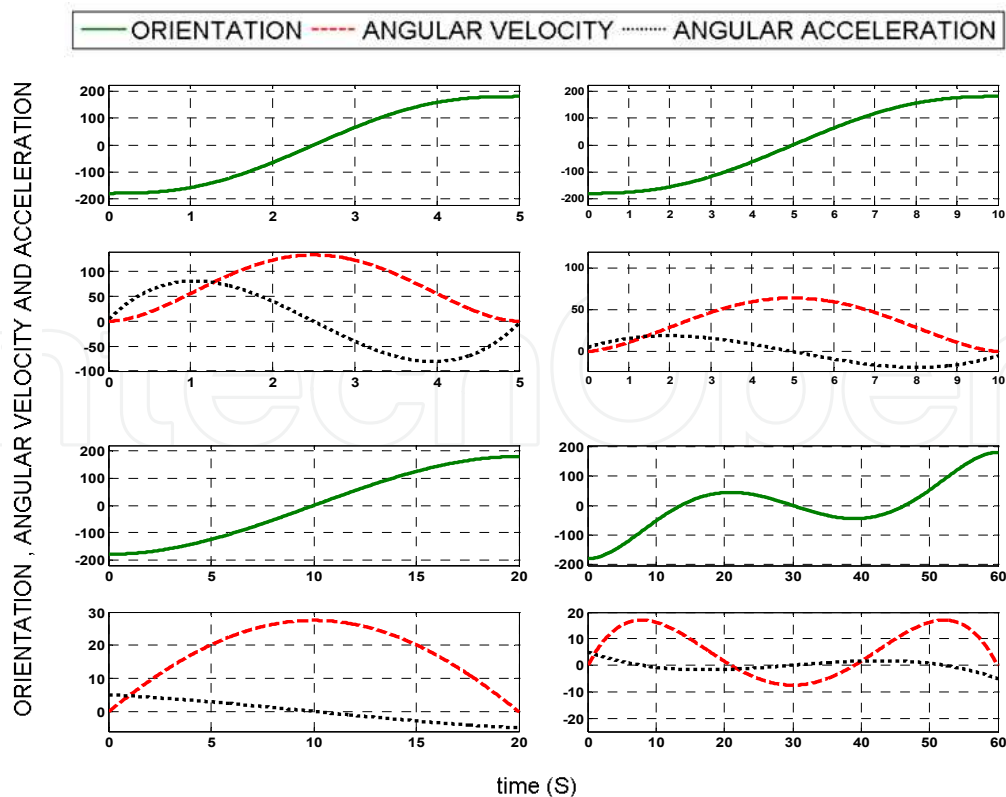


Figure 22. The time comparison of fifth order trajectory planning of the first joint

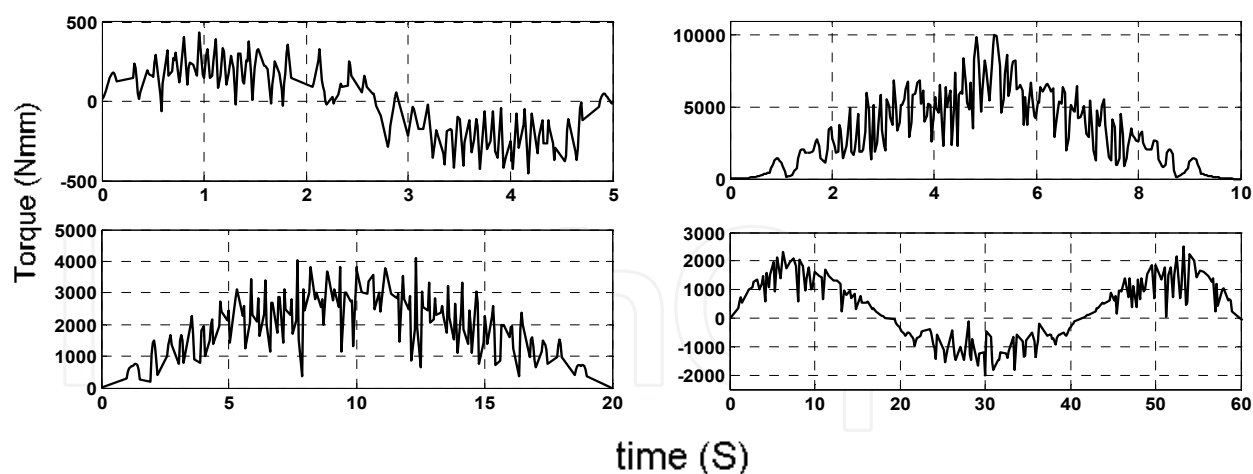


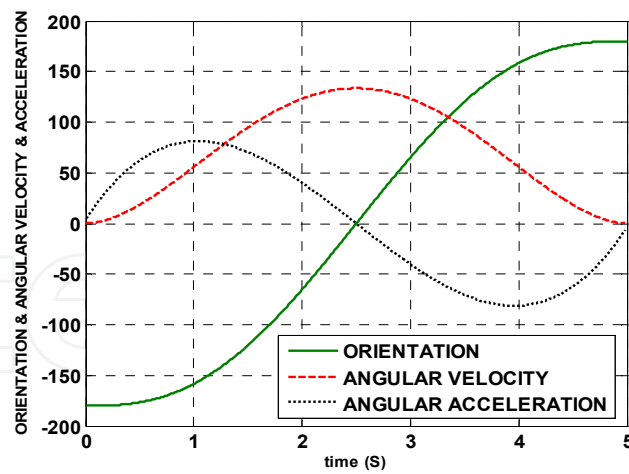
Figure 23. The comparison between the time and torque of the first joint using fifth order trajectory planning

From Figures 22 and 23 the optimum final time required to complete the process for the robot can be selected. It is seen from figures that the trajectory planning of the robot for the times (5, 10 and 20 s) has the same properties i.e. the velocity is inversely proportional with time. The orientation behaviour increased gradually with the time from the initial angle to the final angle. Where the time 60s is omitted from selected. For figure 23 for the torque history in the time (5, 10, 20 and 60 s) it is found for $t_f = 5$ s, the dominant part in the torque history is the inertia matrix. Increasing the final time to 10, 20 and 60s shifts the dominant term from inertia matrix to Centrifugal and Coriolis matrices. This is due to the vanishing of the acceleration at most of the joint trajectory. Another consequence of increasing the final time is the dramatic increase in the peak value of the joint torque which requires big actuator size for the same task (i.e. the same joint parameters). The time 5s to complete the results of the surgical robot can be selected. Since this time has orientation behaviour was increased gradually with the time from the initial angle to the final angle and the torque history curves were affected by the inertia of the link of robot.

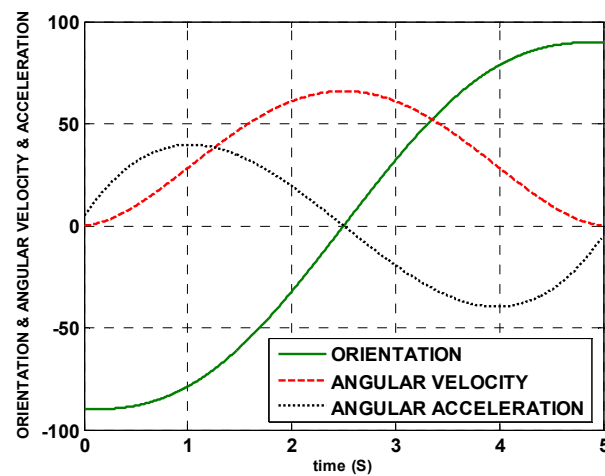
Figures 24a, b and c show the trajectories for the six joints of the robot using Fifth order Polynomial trajectory planning.

5.2.3. Linear segments with parabolic blends

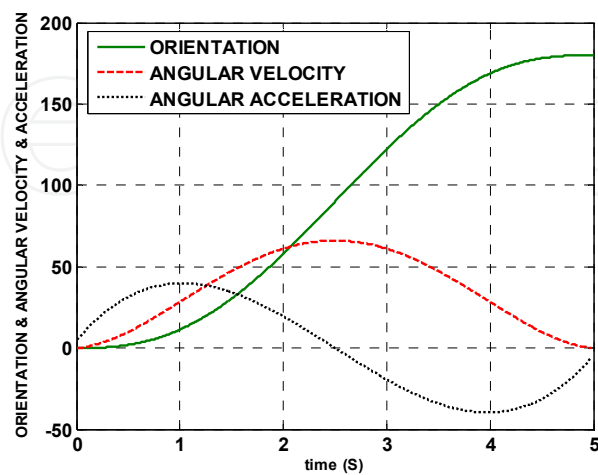
Figures 25 and 26 show comparisons between the orientation, velocity, acceleration, torque and four different time ranges (5, 10, 20 and 60 s) of the first joint of the surgical robot. It is clearly shown in the Figure 25 that the orientation behaviour increases gradually with the time from the initial angle to the final angle and as the time increases the velocity required decreases and also the acceleration i.e. the inertia of the robot link decreases. It is also clearly shown in the Figure 26 that the original torque history has considerable fluctuations. It is clear that the highest hub torque is for joint one while actuator torque of joint 6 is the lowest.



(a) The first, fourth and sixth joints



(b) The second and fifth joints



(c) The the third joint

Figure 24. The orientation, angular velocity and angular acceleration for six joint micro-robot using Fifth order Polynomial trajectory planning

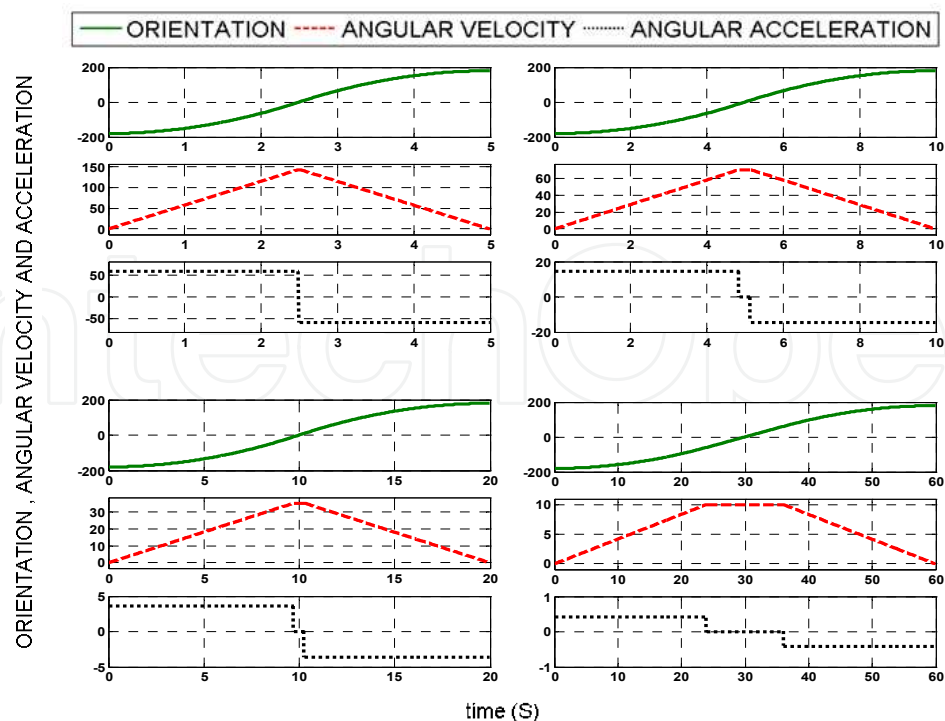


Figure 25. The time comparison of liner segments with parabolic blends trajectory of the first joint

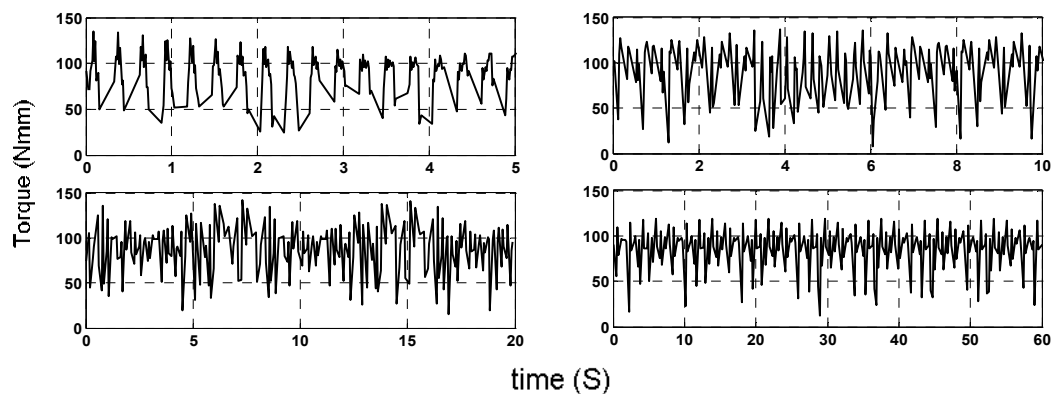
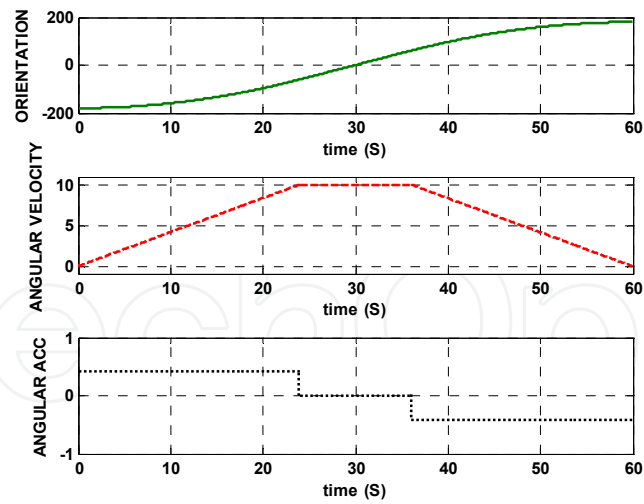


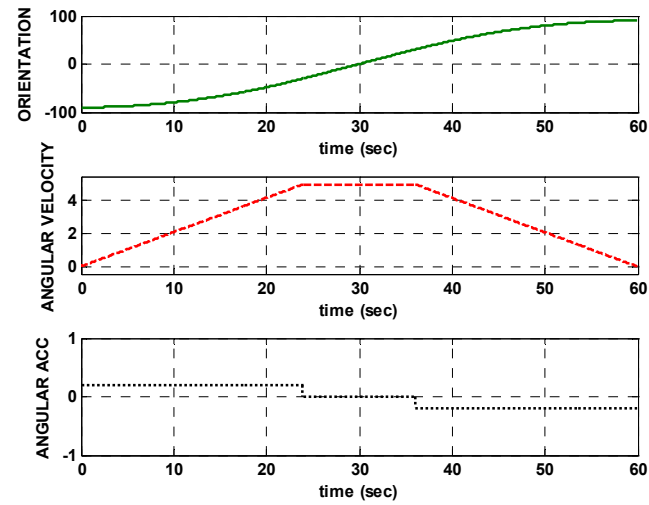
Figure 26. The comparison between the time and torque of the first joint using liner segments with parabolic blends trajectory planning

From Figures 25 and 26 the optimum final time required to complete the process for the robot can be selected. By inspection of Figure 25 the trajectory planning has three segments they are First parabolic blends, Straight line and Second parabolic blends. The straight line segment was very important segment because the velocity in this segment is constant and the acceleration was zero i.e. no inertia of the link of the robot in this segment of time. It is therefore better to have a larger period for this segment. The largest time i.e. 60s to complete the results of the robot can be selected. Since this time has the largest period of straight line segment. By inspection of Figure 26 for the torque history in the time 60s we find that the torque decreases with the time and the shooting decreases gradually near the steady state time 60s.

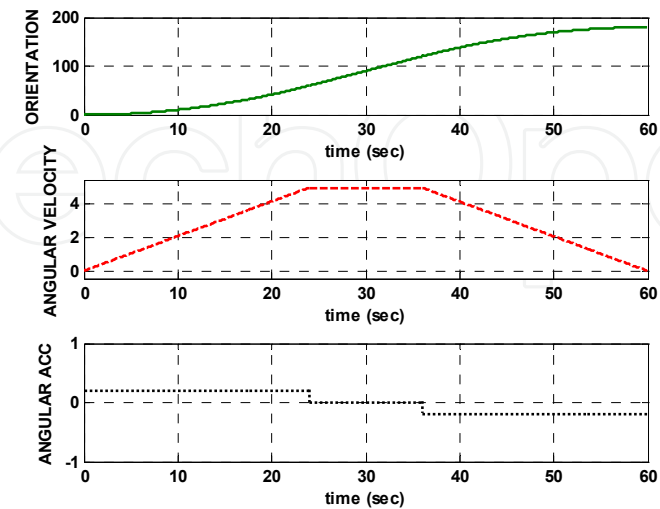
Figures 27a, b and c show the trajectories for the six joints of the robot using linear segments with parabolic blends.



(a) The first, fourth and sixth joints



(b) The second and fifth joints



(c) The third joint

Figure 27. The orientation, angular velocity and angular acceleration for six joint micro-robot using parabolic blends trajectory planning

5.2.4. Soft motion trajectory planning

Figures 28 and 29 show comparisons between the orientation, velocity, acceleration, jerk, torque and four different time ranges (5, 10, 20 and 60 s) of the first joint of the surgical robot. It is clearly shown in the Figure 28 that the orientation behaviour increases gradually with the time from the initial angle to the final angle and as the time increases the velocity required decreases and also the acceleration i.e. the inertia of the robot link decreases. It is also clearly shown in Figure 29 that the original torque history has considerable fluctuations. It is clear that the highest hub torque is for joint one while actuator torque of joint 6 is the lowest.

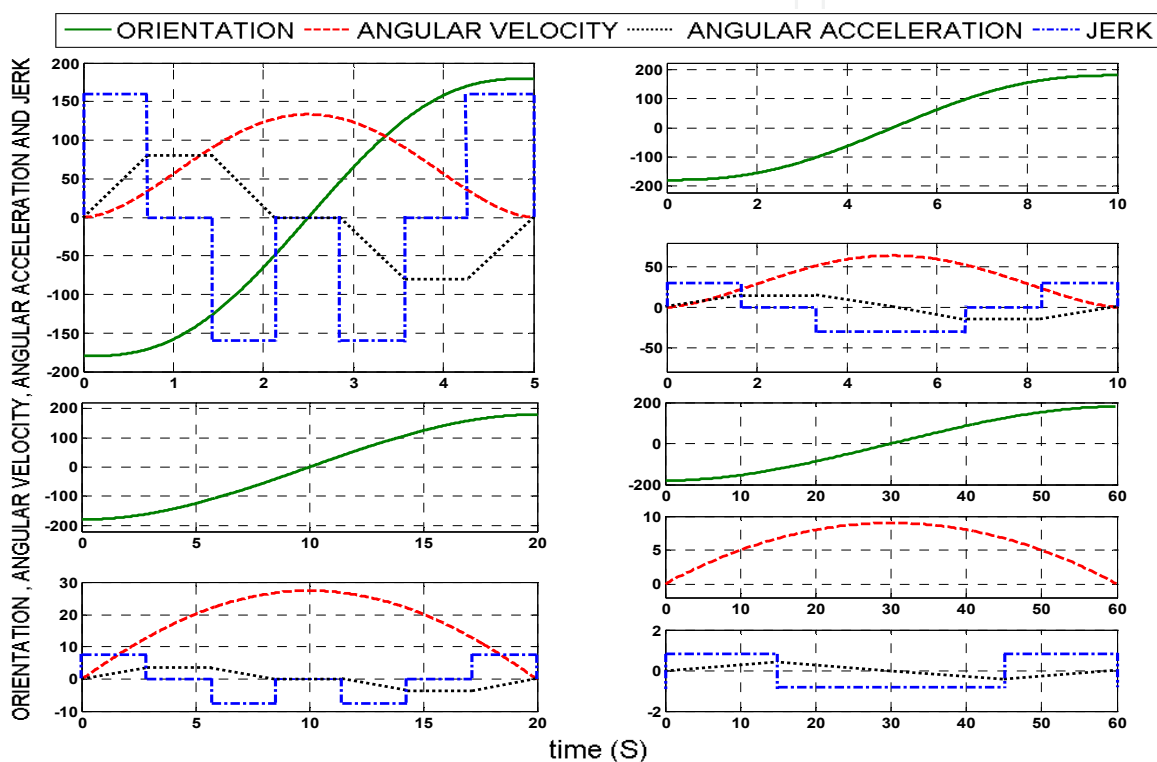


Figure 28. The time comparison of Soft motion trajectory planning of the first joint

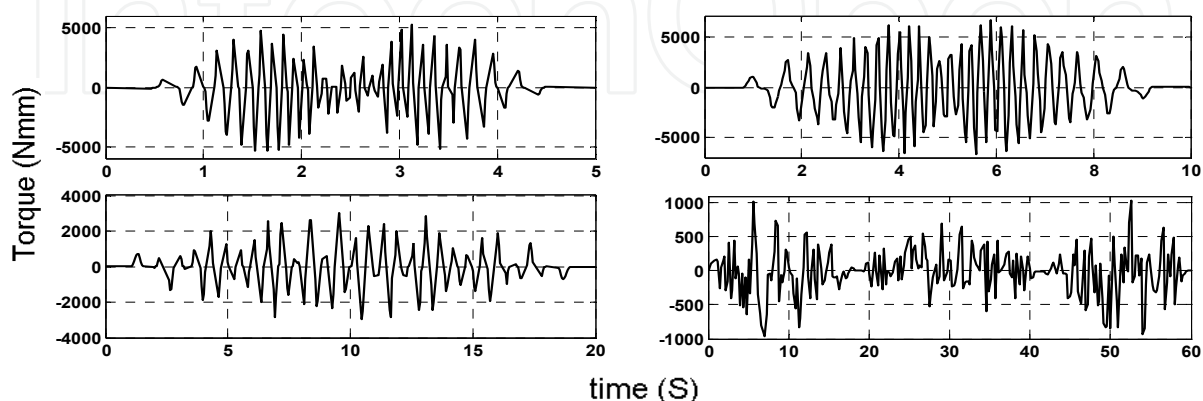
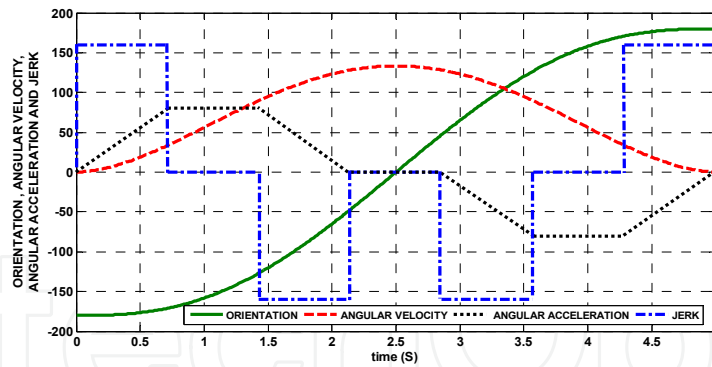
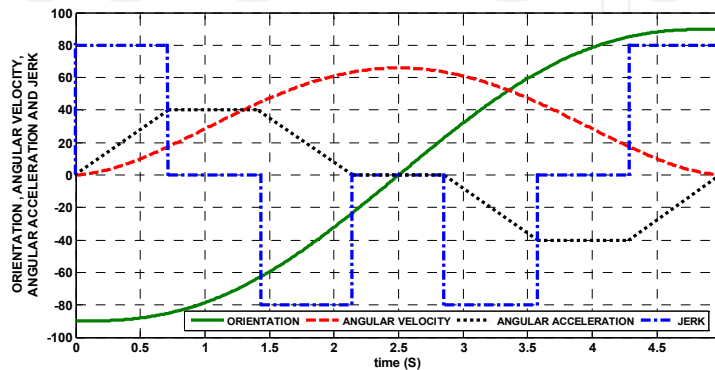


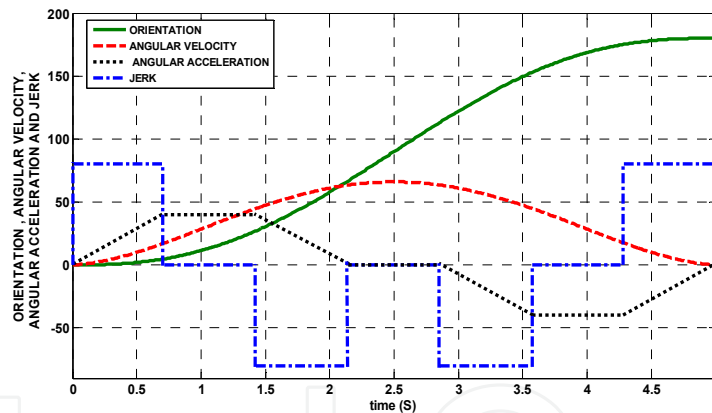
Figure 29. The comparison between the time and torque of the first joint using soft motion trajectory planning



(a) The first, fourth and sixth joints



(b) The second and fifth joints



(c) The third joint

Figure 30. The orientation, angular velocity and angular acceleration for six joint micro-robot using soft motion trajectory planning

From Figures 28 and 29 we can select the optimum final time required to complete the process for the robot. By inspection of Figure 28 we find the trajectory planning three segments they are maximum jerk, maximum acceleration and maximum velocity. The maximum acceleration and maximum velocity segments were very important segments because the velocity in the maximum velocity segment is constant and the acceleration was zero i.e. no inertia of the link of the robot in this segment of time and the acceleration in the maximum acceleration segment is constant and the jerk of the link of the robot was zero. It is therefore better to have a larger period for this segment. It is seen from figures that the segments were released from time 60s but the the maximum velocity segment only was released from time 10s. Where the times (10

and 60 s) is omitted from selected i.e. the times (10 and 60 s) have properties were not satisfactory for trajectory of robot. And by inspection of Figure 29 for the torque history in the times (5 and 20 s) we find for $t_f = 5$ s, the dominant part in the torque history is the inertia matrix. Increasing the final time to 20s, shift the dominant term from inertia matrix to Centrifugal and Coriolis matrices since the effect of angular velocity will be obviously high. This is due to the vanishing of the acceleration at most of the joint trajectory. Another consequence is of increasing the final time is the dramatic change in the peak value of the joint torque which requires big actuator size for the same task. So we can select the time 5s to complete the results of the robot because the torque history curve was affected by the inertia and it has the important segment i.e. maximum acceleration and maximum velocity segments.

Figures 30a, b and c show the trajectories for the six joints of the robot using soft motion trajectory planning.

5.3. The dynamic response analysis

As previously stated that the dynamic analysis of the surgical robot was derived using Lagrange-Euler technique and the results of the dynamic analysis were depended on the method of trajectory planning. In the trajectory planning results were derived using four different methods. The dynamic analysis results of surgical robot are divided into four parts each part in for each special method of the four different methods which have been derived from the trajectory planning.

Figures 31 to 34 show the original torque history which clearly shown considerable fluctuations. It is clear that the highest hub torque is experienced at joint one while actuator torque of joint 6 is the lowest.

5.3.1. The dynamic analysis results using Third order Polynomial trajectory planning

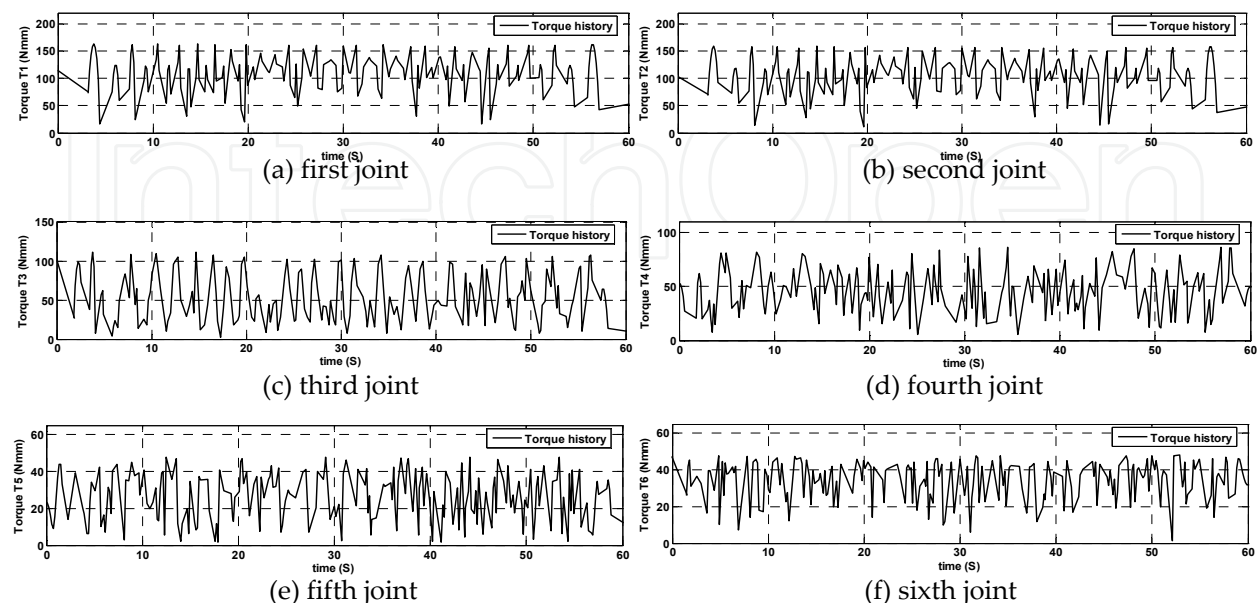


Figure 31. Torque history for Third order Polynomial trajectory planning

5.3.2. The dynamic analysis results using Fifth order Polynomial trajectory planning

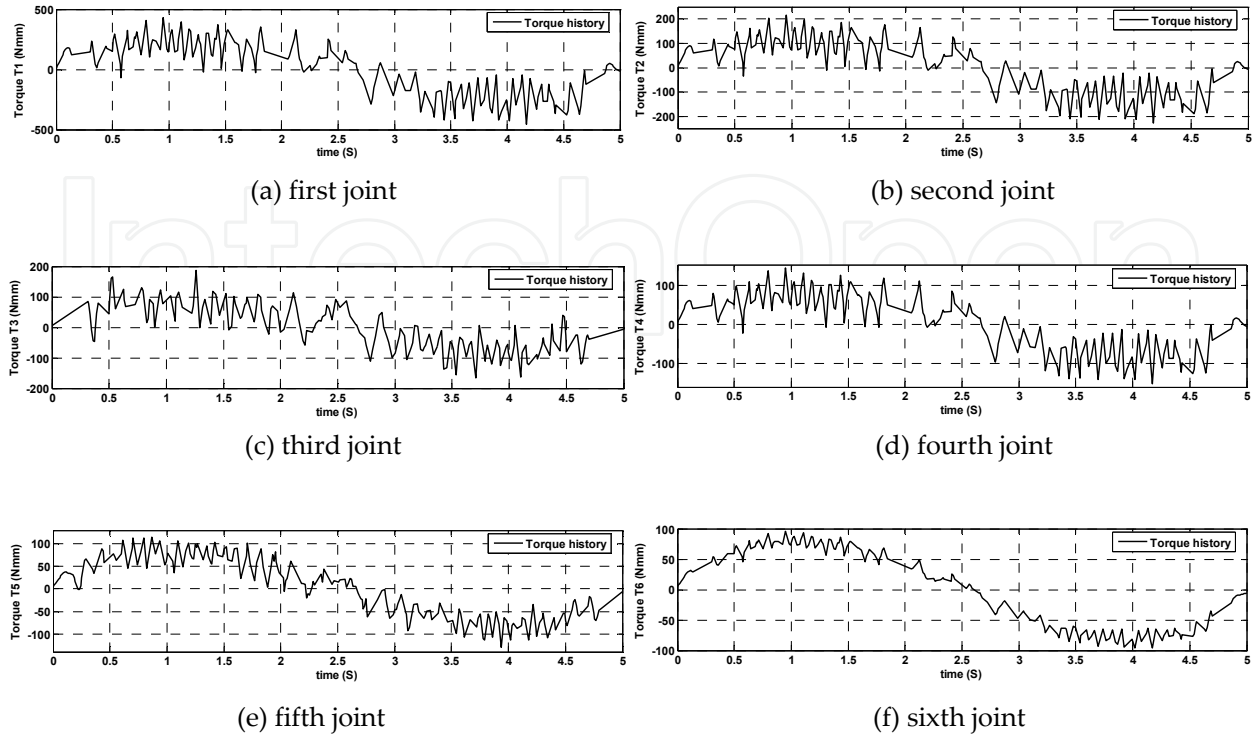


Figure 32. Torque history for Fifth order Polynomial trajectory planning

5.3.3. The dynamic analysis results using Linear segments with parabolic blends trajectory planning

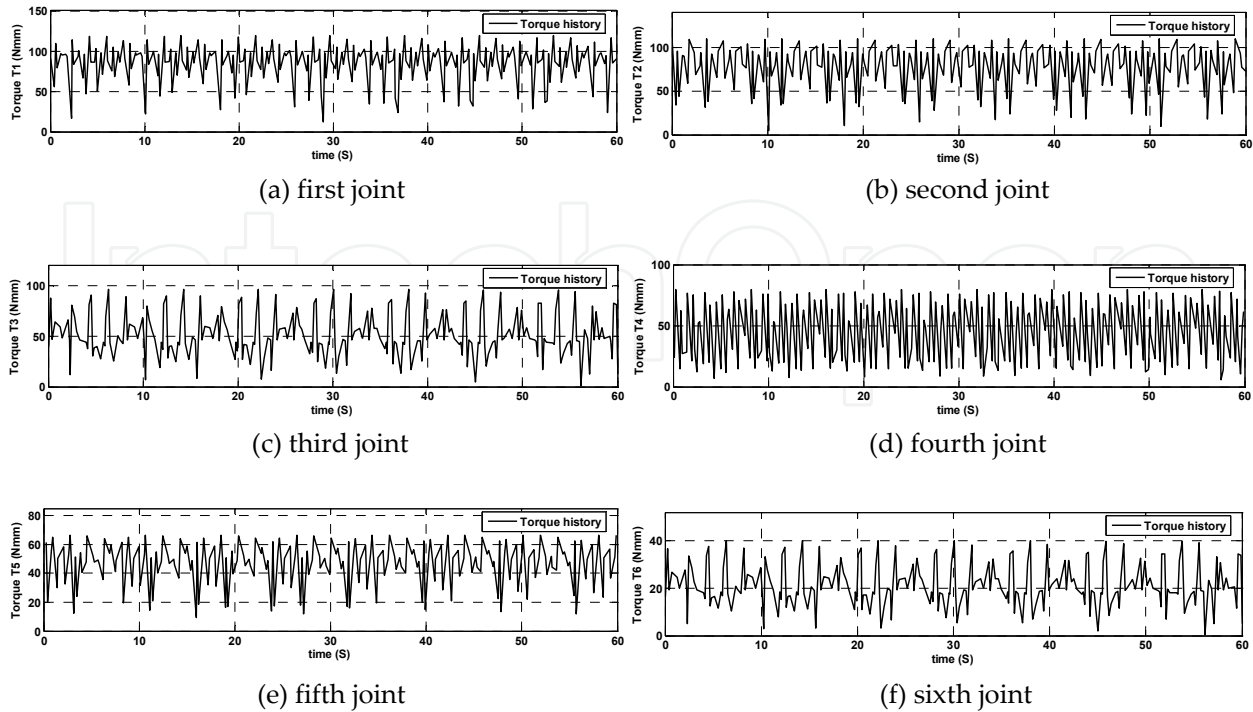


Figure 33. Torque history for Linear segments with parabolic blends trajectory planning

5.3.4. The dynamic analysis results using Soft motion trajectory planning

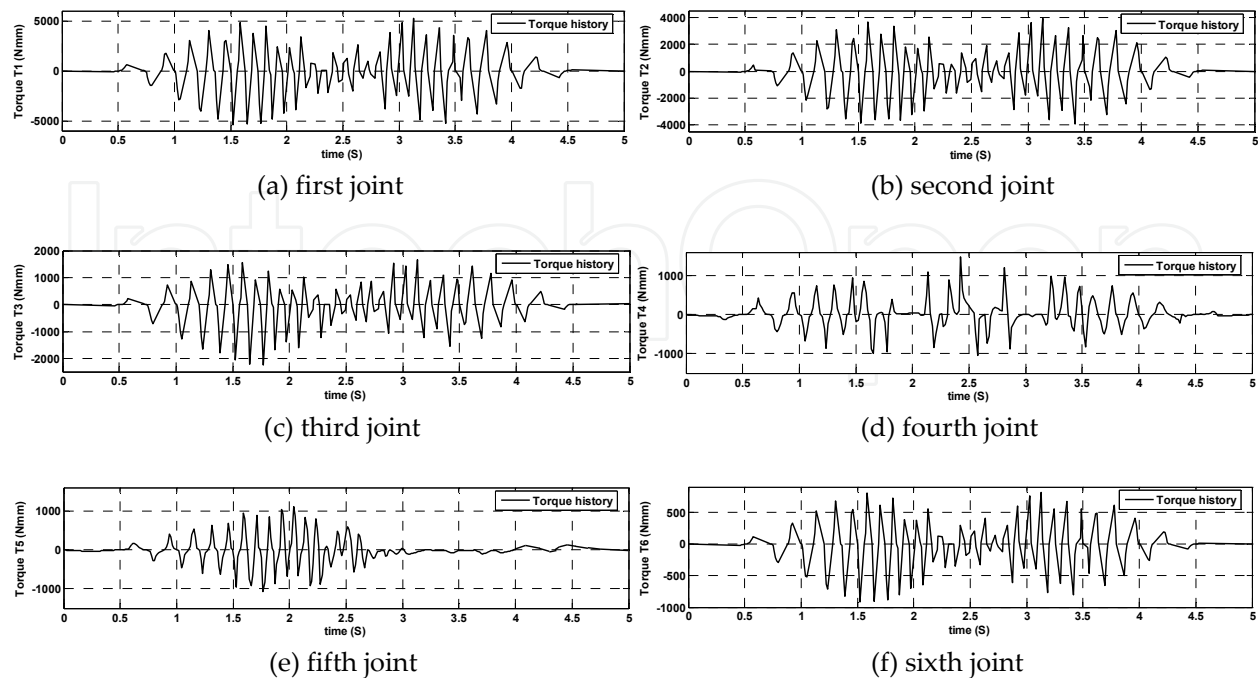


Figure 34. Torque history for Soft motion trajectory planning

6. Conclusions

A kinematic and dynamic analysis for a six-degree-of-freedom surgical robot were presented in this work. The kinematic model is based on Denavit-Hartenberg representation and the workspace of the end-effector is defined by solving the inverse kinematics problem. Four different methods were used to derive the trajectory planning for the six joints and were designed and employed to calculate the torque history for the six actuators. The dynamic equations of motion in symbolic form were derived using the Lagrange-Euler technique and the torque history was obtained using MATLAB for each joint. The proposed algorithm is flexible and can be extended to any robot configuration provided that the Denavit-Hartenberg presentation was available and the physical limits of joints are defined. The original torque history has considerable fluctuations. It was shown that the highest hub torque was of joint 1 while actuator torque of joint 6 was the lowest. It should be also noted that changing the final time for the joint trajectory changes the torque history considerably. The final time required to complete the process was selected depending on the method used to derive the trajectory planning as previously stated.

It was clearly shown in this work that the best method of trajectory planning that gives the smooth set trajectory planning and best performance of the robot under investigation was the soft motion trajectory planning because the most important reason for this selection was the torque history that has the lowest number ever of shooting and the shooting was distributed regularly over the period of time unlike the other methods which have a long number of shootings and were distributed randomly. Also the reason for selecting this

method was the disappearance of the shooting quite before the final time of trajectory i.e. the steady state time. T

Author details

Wael A. Al-Tabey

*Department of Mechanical Engineering Faculty of Engineering,
Alexandria University, Alexandria, Egypt*

Appendix

Appendix (A): Notation

C_j	$\cos \theta_j$
S_j	$\sin \theta_j$
j	the total number of joints
i	the total number of coordinates
n	the total number of links
k and p	coefficient (1, 2, 3... n)
x, y, z	local joint coordinates
X, Y, Z	global joint coordinates
Y_i	the Kinematics equations of the end effectors
X_j	the joint variables ($\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$)
D	the end effector differential translation matrix
D_θ	the joint differential motion matrix
J	the robot Jacobian matrix ($i \times j$)
θ_i	the initial values of each joint angle
θ_f	the final values of each joint angle
t_f	the time duration
$\theta(t), \dot{\theta}(t), \ddot{\theta}(t), J(t)$	the orientation, angular velocity, angular acceleration and angular jerk respectively
$\ddot{\theta}_0, \dot{\theta}_0, \theta_0$	the initial conditions
T_{jpa}	Jerk positive initial time
T_{aca}	Acceleration constant initial time
T_{jna}	Jerk negative initial time
T_{vc}	Velocity constant time
T_{jnb}	Jerk negative final time
T_{acb}	Acceleration constant final time
T_{jpb}	Jerk positive final time
q	the vector of generalized joint coordinates
\dot{q}	the vector of joint velocities
\ddot{q}	the vector of joint accelerations
M	the inertia matrix

C	the Coriolis and centrifugal matrix
G	the gravity matrix
Q	the vector of generalized force associated
g^T	the gravity matrix (1x4)
g_x, g_y and g_z	the gravity components in x, y and z direction respectively
\bar{r}_j	the location of the center of mass of link relative to the frame representing the link
T_j	the torque on joint (j)
J_i	the inertia matrix
I_{XX}, I_{YY} and I_{ZZ}	the principal moments of inertia for the link
I_{XY}, I_{XZ} and I_{YZ}	the parallel moments of inertia for the link
m_i	the mass of the link
\bar{X}_i, \bar{Y}_i and \bar{Z}_i	the distance between the X, Y and Z axis to the center of the link mass respectively

Appendix (B): MATLAB Code

```
%*****
%
%                               Micro-Robot Management
%                               Complete MATLAB Code for kinematic
%*****
syms th1 th2 th3 th4 th5 th6 L1 L2 L3 L4
%*****THE INPUTS OF ROBOT*****
L1=input('please enter The length of Link NO(1) (mm)=');
L2=input('please enter The length of Link NO(2) (mm)=');
L3=input('please enter The length of Link NO(3) (mm)=');
L4=input('please enter The length of Link NO(4) (mm)=');
th1min=input('please enter minth1 (degree)=');
th1max=input('please enter maxth1 (degree)=');
th2min=input('please enter minth2 (degree)=');
th2max=input('please enter maxth2 (degree)=');
th3min=input('please enter minth3 (degree)=');
th3max=input('please enter maxth3 (degree)=');
th4min=input('please enter minth4 (degree)=');
th4max=input('please enter maxth4 (degree)=');
th5min=input('please enter minth5 (degree)=');
th5max=input('please enter maxth5 (degree)=');
th6min=input('please enter minth6 (degree)=');
th6max=input('please enter maxth6 (degree)=');
%*****
th1=th1min:2:th1max;th2=th2min:th2max;th3=th3min:th3max;
th4=th4min:2:th4max;th5=th5min:th5max;th6=th6min:2:th6max;
%*****
```

```

c1=cos(th1);c2=cos(th2);c3=cos(th3);c4=cos(th4);c5=cos(th5);c6=cos(th6);
s1=sin(th1);s2=sin(th2);s3=sin(th3);s4=sin(th4);s5=sin(th5);s6=sin(th6);
%*****D.H matrix*****
A1=[c1,-s1,0,L1*c1;s1,c1,0,L1*s1;0,0,1,0;0,0,0,1];A2=[c2,0,s2,L2*c2;s2,0,-c2,L2*s2;0,1,0,0;0,0,0,1];
A3=[c3,0,-s3,0;s3,0,c3,0;0,-1,0,0;0,0,0,1];A4=[c4,0,-s4,0;s4,0,c4,0;0,-1,0,L3;0,0,0,1];
A5=[c5,0,-s5,0;s5,0,c5,0;0,-1,0,0;0,0,0,1];A6=[c6,-s6,0,0;s6,c6,0,0;0,0,1,L4;0,0,0,1];
A01=A1;A02=A1*A2;A03=A1*A2*A3;A04=A1*A2*A3*A4;A05=A1*A2*A3*A4*A5;
A06=A1*A2*A3*A4*A5*A6;
%*****The Kinematics equations of the end effectors*****
px= A06(4,1),py= A06(4,2),pz=A06(4,3)
subplot(1,3,1);plot(xa,ya);
xlabel('Px');ylabel('Py');grid on
subplot(1,3,2);plot(xa,z);
xlabel('Px');ylabel('Pz');grid on
subplot(1,3,3);plot(ya,z);
xlabel('Py');ylabel('Pz');grid on
%*****THE end*****
%*****CREATED BY DR/WAEL A. AL-TABEY*****

```

7. References

- [1] Frumento, C. (2005). Development of micro tools for surgical applications. A Ph. D. Co-Tutorship Thesis, University' Degli Studi De Genova/ Uiversite Piere et Marie Currie Paris.
- [2] Venita, C. Sanjeev, D. and Craig, T. (2006). Surgical robotics and image guided therapy in pediatric surgery: Emerging and converging minimal access technologies, Seminars in Pediatric Surgery, Volume 14, pp. 267– 275.
- [3] Tsai, T. and Hsu, Y. (2004). Development of a parallel surgical robot with automatic bone drilling carriage for stereotactic neurosurgery, IEEE SMC, International Conference on Systems, Man and Cybernetics, Hague, Netherlands, October 10-13, 2004.
- [4] Miller, A. and Christensen, H. (2003). Implementation of Multi-rigid-body Dynamics within a Robotic Grasping Simulator, Proceedings of International Conference, Volume 2, pp. 2262, 14-19 Sept. 2003.
- [5] Featherstone, R. and Orin, D. (2000). Robot Dynamics: Equations and Algorithms, Proc. IEEE Int. Conf. on Robotics and Automation, pp. 826-834.
- [6] Wang, S. (2008). Conceptual design and dimensional synthesis of Micro-Hand, Mechanism and Machine Theory, Volume 43, Issue 9, pp. 1186-1197.
- [7] Alici, G. and Shirinzadeh, B. (2004). Loci of singular configurations of 3-DOF spherical parallel manipulator, Robotics and Autonomous Systems, Volume 48, pp. 77–91.
- [8] Ben-Horin, R. (1998). kinematics, dynamics and construction of a planarly actuated parallel robot, Robotics and computer –integrated Manufacturing, Volume 14, pp. 163–172.
- [9] Bonnifait, P. and Garcia, G. (1999). 6-DOF dynamic localization of an outdoor mobile robot, Control Engineering Practice, Volume 7, pp. 383–390.

- [10] Abdellatif, H. and Heimann, B. (2008). Computational efficient inverse dynamics of 6-DOF fully parallel manipulators by using the Lagrangian formalism, *Mechanism and Machine Theory*, Volume 7, pp. 383–390.
- [11] Zhu, Z. (2005). Kinematic and dynamic modelling for real-time control of Tau parallel robot, *Mechanism and Machine Theory*, Volume 40, pp. 1051–1067.
- [12] Gouliaev, V. and Zavrazhina, T. (2001). Dynamics of a flexible multi-link cosmic robot-manipulator, *Journal of Sound and vibration*, Volume 243(4), pp. 641–65.
- [13] Carrera, E. and Serna, M. (1996). Inverse dynamics of flexible robots, *Mathematics and Computers in Simulation*, Volume 41, pp. 485–508.
- [14] Martins, F. (2008). An adaptive dynamic controller for autonomous mobile robot trajectory tracking, *Control Engineering Practice*, Volume 16, pp. 1354– 1363.
- [15] Valero, F. (2006). Trajectory planning in workspaces with obstacles taking into account the dynamic robot behaviour, *Mechanism and Machine Theory*, Volume 41, pp. 525– 536.
- [16] Geng, T. (2005). Dynamics and trajectory planning of a planar flipping robot, *Mechanics Research Communications*, Volume 32, pp. 636–644.
- [17] Alessandro, G and Vanni, Z. (2008). A technique for time-jerk optimal planning of robot trajectories, *Robotics and Computer-Integrated Manufacturing*, Volume 24, pp. 415– 426.
- [18] Pires, S. (2007). Manipulator trajectory planning using a MOEA, *Applied Soft Computing*, Volume 7, pp. 659– 667.
- [19] Chettibi, T. (2004). Minimum cost trajectory planning for industrial robots, *European Journal of Mechanics A/Solids*, Volume 23, pp. 703–715.
- [20] Alessandro, G. and Vanni, Z. (2007). A new method for smooth trajectory planning of robot manipulators, *Mechanism and Machine Theory*, Volume 42, pp. 455– 471, 2007.
- [21] Corke, P. (1996). A Robotics Toolbox for MATLAB, *IEEE Robotics and Automation Magazine*, 3(1):pp.24–32.
- [22] Ata, A. (2007). Optimal Trajectory Planning for Manipulators: A Review, *Journal of Engineering Science and Technology*, Volume 2, No. 1, pp. 32–54.
- [23] Niku, S. (2001). *Introduction to Robotics Analysis, Systems, Applications*, London, International (UK) Limited, pp. 147–159.
- [24] Herrera, I. and Sidobre, D. (2006). Soft Motion and Visual Control of Service Robot, *The Fifth International Symposium in Robotics and Automation*, August 2006.
- [25] Niku, S. (2001). *Introduction to Robotics Analysis, Systems, Applications*, London, International (UK) Limited, pp. 128–138.
- [26] Al-Tabey, W. (2010). Effect of Trajectory Planning on Dynamic Response of Micro-Robot for Surgical Application, *9th WSEAS International Conference on System Science and Simulation in Engineering (ICOSSSE '10)*, ISBN: 978-960-474-230-1, Iwate Prefectural University, Japan, October 4–6, 2010.
- [27] Tawfik, K. Ata, A. Al-Tabey, W. (2009). Kinematics and dynamics analysis of micro-robot for surgical applications, *World Journal of Modelling and Simulation*, Vol. 5, No. 1, pp. 22–29, ISSN 1 746-7233.