Tikrit Journal of Eng. Sciences/Vol.18/No.2/June 2011, (72-87)

Verification and Generation of Safe Straight Paths for a 4-DOF Spherical Manipulator

Rawand Ehsan J.T.
College of Engineering - Kirkuk University

Abstract

Sometimes in a manufacturing environment, a robotic arm is wanted to move in a straight path such as welding, painting and assembling. This straight path causes the manipulator to actuate all or most of its joints in the same time to track the path. Along this path, the manipulator may reach a specific singular configuration in its workspace at which one or more joints are in their limits, or a part of the path lies outside the workspace. These conditions make the arm's movement be unsmooth and may cause damage to the manufacturing process. In this paper, the singularities inside the workspace of a 4-DOF spherical manipulator are indicated and a method is presented for finding the arm configurations (assuming that all joints are actuated at the same time) along a straight path between an initial and a goal configurations. All joint limits are presented and if a part of the path lies outside the workspace, the model processes this condition by introducing a new initial configuration through changing the third joint's (q_3) position only. A smooth straight path is generated between any two configurations using the parametric equations of the line connecting them. Unlike the analytical inverse kinematics, which needs a (4 x 4) homogeneous transformations convention matrix (DH) to find the joint variables, this method needs only the initial configuration, goal configuration, link lengths and the corresponding Cartesian coordinates of the path. It always gives the correct solution for the under taken path.

Keywords: Singularity, Jacobian matrix, rank deficiency, path generation, kinematics, configuration.

توليد مسارات مستقيمة أمنة و التحقق منها لذراع ألي ذو حركة كروية واربع درجات من الحرية

الخلاصة

في البيئة الصناعية, أحيانا يطلب من الذراع الألي التَحَرُّك على مسار مستقيم كما هو الحال في اللحام والطلاء والتجميع. هذا المسار المستقيم قد يجبر الذراع على تحريك جميع أو معظم مفاصله (مشغلاته) في نفس الوقت لكي يتعقب المسار. وعلى طول المسار, قد يصل الذراع إلى نقاط في محيط عمله فيها أحد أو أكثر من مفاصله تكون قد وصلت إلى الحد المسموح به والتي لايمكن به الذراع مواصلة حركته أو أن هذا المسار قد يكون بشكل كلي أو جزئي واقع خارج فضاء عمل الذراع وكل هذا يجعل عمل أو حركة الذراع غير سلس أو قد يسبب الضرر للمهمة الموكلة بها. في هذا البحث, كل نقاط التي عندها الذراع يخسر أحد أو أكثر من حركة مفاصله داخل فضاء عمله يتم التحقق منها. الذراع المستعمل له أربع درجات من الحرية وذات حركة كروية. يتم تقديم طريقة تعتمد على حركة الذراع على مسار مستقيم بين نقطة بداية الحركة ونقطة الهدف ويتم فيها حساب مقدار حركة كل مفصل (على فرض أن كل المفاصل تتحرك في نفس الوقت) لضمان بقاء الذراع على المسار. في حالة كون كل أو جزء من هذا المسار يقع خارج فضاء عمل الذراع فأن الطريق أعلاه سوف يقوم بحساب نقطة بداية جديدة لحركة الذراع

بأعتماد على تغير قيمة المفصل الثالث فقط. يتم أعتماد معادلة الخط المستقيم الثلاثي البعد في إيجاد مسار مستقيم بين أية نقطتين وعلى الخلاف من الحل العكسي للمعادلات الكينماتيكية (التي تحتاج إلى مصفوفة 4x4 في إيجاد مقادير حركة المفاصل) تحتاج هذه الطريقة فقط إلى المعلومات حول نقطتي البداية والهدف وأبعاد الذراع و أحداثيات النقاط المنتخبة على المسار. الطريقة دائما تعطي الحل الصحيح للمسار المعتمد.

الكلمات الدالة

النقاط الانفرادية, مصفوفة الجاكوبي, نقص المرتبة, توليد المسار, الكينماتيك, مجموعة قيم المفاصل.

Notations

- X Cartesian position vector.
- $\Phi(q)$ position vector in terms of joint
- q Vector of generalized coordinates (joint variables).
- n number of DOF.
- P A (3x1) translation vector.
- R A (3x3) rotation matrix.
- T A (4x4) transformation matrix.
- S A set.
- u A subvector of generalized coordinates.
- q_{intial} = $[q_{1i}\ q_{2i}\ q_{3i}\ q_{4i}]$ initial configurations.
- q_{goal} = $[q_{1g}\ q_{2g}\ q_{3g}\ q_{4g}]$ goal configurations.
- $\Phi_q(q)$ Jacobian matrix.
- p_i A singular set of constant generalized coordinates.
- R^n Space of n- coordinates.
- Ψ A bounded parameterized subsurface.
- t A parameter.

Introduction

Many tasks performed by a manipulator arm in a manufacturing environment such as welding, spray-painting and assembling, required that the end-effector follows a straight path trajectory connecting an initial configuration to a goal configuration.

During this process of motion, singular behavior of the manipulator may occur inside its workspace, or a part of the straight path lies outside the workspace. All theses conditions make the path be unsmooth which by itself may cause damages to the manufacturing process. In simple terms, a singularity of a robotic arm occurs where the number of instantaneous degree of freedom (DOF) of its end-effector differs from the expected number based on the DOF of its individual actuated joints. There are mainly three types of manipulator singularities: work-space boundary singularity at which one of the joints reaches its limit, a singularity inside the work-space at which one or more joints reach their limits, and a singularity also inside the workspace at which the manipulator losses one of its DOF without being any joint at its limit.

The significance of singularities in the design and control of robots is well known and there is an extensive literature on the determination and analysis of singularities for a wide variety of serial manipulators-indeed such an analysis is an essential part of manipulator design. Donelan[1], in his study, provides singularity theory methodologies for a deeper analysis with the aim of classifying singularities, providing local models and local and global invariants, and surveys the applications of singularity-theoretic methods in robot kinematics and presents some new results. Investigations of manipulator singularities are reported by Abdel-Malek[2]. He presented algorithms base on the Jacobian matrix ranks deficiency and classified the singularity into three types: type **I**, where no joints reach their limits and types **II** and **III** where some joints reach their limits. According to theses types, a series of generalized constant coordinates subset vectors is generated that can be submitted into the position vector of the end-effector to produce a series of parametric singular surfaces and curves as a function of the remaining generalized non constant coordinate vectors. These singular curves and surfaces can be used also to draw the interior and exterior boundaries to the workspace of the robotic arm; this is shown by the work of Abdel-Malek[3].

One of the main problems in robotics research is the generation of trajectories that a manipulator must follow and the computation of the joint variables required to move the hand to the target positions. A proper motion plan can have advantages with respect to different aspects, for example, obstacle avoidance, work or method simplicity and efficiency, better tracking performance etc. For multi-link robotic systems, the automatic task execution can be divided into three smaller subproblems[4]:

P1 For a given robot and task, plan a path for the end-effector between two specified positions. Such a path optimize a performance index, in the mean time satisfies either equality (for instance, robot's end-tip is required to move on a surface) or inequality (for instance, obstacle avoidance, joint angle limit) constraints.

P2 For a given end-effector path expressed in the task (operational) space (usually coincides with the Cartesian space), find the joint trajectory according to our knowledge about the robot kinematics and kinetics. Similarly, some performance index can be optimized in case of a redundant robot; namely, the robot has more DOFs than necessary to perform the given task.

P3 Design a feedback controller which can track the given reference joint trajectory accurately.

Generation of path trajectory is usually accomplished by the inverse kinematics of the manipulator, which may be hard to derive or may not exist at all. As alternative approaches, neural networks and optimal search methods have been used for inverse kinematics modeling and control in robotics. Rosales, Gan, Hu, and Oyama[5] present a first analytical solution to the inverse kinematics of **Pioneer 2** robotic arm which combined with an optimal search method. On some rare occasions, the inverse model provides completely wrong solution due to the inaccuracy problem in **atan2** function, which is a disadvantage of the analytical inverse model and in order to avoid this problem, they used a hybrid approach. This approach works as follows: given a desired DH convention, the inverse kinematic will provide joint variables. Its corresponding position and orientation will be calculated using the forward kinematics and if this solution meets the correctness criterion, the joint variables will be sent to the arm, otherwise, an optimal search will be conducted to get a satisfactory solution. Qin and Perpinan[6] present a machine learning approach for trajectory inverse kinematics. Given a trajectory in workspace, find a feasible trajectory in angle space (joint space). The method learns offline a conditional density model of the joint variables given the workspace coordinates. This density implicit defines the multivalued inverse kinematics mapping for any workspace point. At run time, the method computes the modes of the conditional density given each of the workspace points, and finds the reconstructed joint variable by minimizing over the set of modes a global, trajectory –wide constraint that penalties discontinuous jumps in joint space or invalid inverse. They demonstrate the approach with a PUMA 560 robot arm. Their approach works well even when the workspace trajectory contains singularities. Calderon, Rosales, and Alfaro[7] presents a comparison between an analytical inverse kinematics based hybrid approach and a resolve motion rate control method (RMRC) for controlling the **Pioneer** arm. In their work, trajectories for arm to follow in the Cartesian space or work space are obtained by image processing via imitation. This implies having a transformation from the visual information of the external model to the execution information of the arm. The transformation process gives the position/orientation of a specific point and the processing of sequential images produces a sequence of target points.

As it can be noted from above, there are many problems in path generation and joint variable calculations. These problems can be summarized into two mainly problems:

1- Singularities,

2- The uncertainties that may result in the solutions of the inverse kinematics model, therefore, the researchers produced many methods and approaches to overcome these problems.

In this paper, a 4-DOF spherical manipulator is presented. All singularities of the manipulator are obtained using the algorithms in the work of Abdel-Malek[2]. A method based on the geometry movement of the spherical manipulator is developed. A straight line connects an initial and final configuration and according to the parametric equation of this line, the end-effector is forced to track the path by computing the joint variables. The method assumes that all joints must be actuated at the same time. The first joint variable (q_1) is calculated depending on the change in the parametric coordinates, the second joint variable (q_2) changes in the interval [$q_{2\ initial}$, $q_{2\ goal}$] with a specified step, and the third and fourth joint variables (q_3, q_4) are computed based on corresponding q_2 and the change in the parametric coordinates. (q_2 & q_3) are updated when (q_4) has a negative value, since $q_4 \in$ [q_4: $0 \rightarrow 400$] and this is by letting $q_4 = 0$ and computing the corresponding q_2 & q_3. If a part of the path lies outside the workspace, the method produces a new initial configuration by changing (q_3) only. The paper is organized as follows: kinematics of the manipulator is given in section 2. In section 3, the singularity algorithms and generation of the joint variables according to the path parametric equation are presented. Finally, some conclusions are given in section 4.

Manipulator Kinematics

1- Forward Kinematics

For a serial manipulator, the forward (direct) kinematics describes the position of the end-effector– parametrised in space by, say, $x_1, \ldots, x_6$ where three parameters correspond to translations, and three to rotations– as a function *f* of the actuated joint variables $q_1, \ldots, q_n$. The joint variables are the angles between the links in the case of revolute joints, and the link extension in the case of prismatic joints. The fixed coordinate systems attached to the 4- link spherical manipulator linkages, which called the word or base frame, are shown in figure (1). Five word frames are used to describe the position and orientation of the end-effector (frame 4) with respect to manipulator base (frame 0). The homogeneous transformations or Denavit-Hartenberg (*DH*) convention is used to simplify the transformation among the attached coordinate frames, combines the operations of rotation (*R*) and translations (*P*) into a single general matrix multiplication, and finds the link parameters. For the manipulator shown in figure (1), the four *DH* convention matrices are:

$$T_1^0 = \begin{bmatrix} \cos q_1 & -\sin q_1 & 0 & 0 \\ \sin q_1 & \cos q_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ..(1)$$

$$T_2^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & q_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ..(2)$$

$$T_3^2 = \begin{bmatrix} -\sin q_3 & 0 & \cos q_3 & 0 \\ \cos q_3 & 0 & \sin q_3 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ..(3)$$

$$T_4^3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q_4 + d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ..(4)$$

q_1 and q_3 are joints 1 & 3 angles, q_2 and q_4 are joints 2 & 4 extensions, and d_1 and d_4 are the link lengths. The general forward kinematics DH transformation can be obtained by $\prod_{i=1}^{4} T^{i}{}_{i-1}$,and is given as below:

$$T_4^0 = \begin{bmatrix} -\cos q_1 \sin q_3 & \sin q_1 & \cos q_1 \cos q_3 & (q_4 + d_4)\cos q_1 \cos q_3 \\ -\sin q_1 \sin q_3 & -\cos q_1 & \sin q_1 \cos q_3 & (q_4 + d_4)\sin q_1 \cos q_3 \\ \cos q_3 & 0 & \sin q_3 & (q_4 + d_4)\sin q_3 + q_2 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ..(5)$$

with link parameter shown in table (1).

This manipulator has joint constraints as follows:

$0 \le q_1 \le 360^0$, $0 \le q_2 \le 400$ *mm*,
$-75^0 \le q_3 \le 180^0$, and $0 \le q_4 \le 400$ *mm*.

2-<u>Inverse Kinematics</u>

The inverse kinematics problem concerned with finding the joints variables in terms of the end-effector position and orientation, and it is, in general, more difficult than forward kinematics problem. The more degrees of freedom that the manipulator may have, the more difficult inverse kinematics solution is. Because the current manipulator has 4-DOF, closed form solution, that based on analytic expressions, can be used[8].

Let:

$$H = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ..(6)$$

be a (4x4) homogenous transformation, here *H* represents the desired position and orientation of the end-effector on the path, and the task is to find the values for joint variables so that $T_4^0 = H$. Therefore:

$$\left.\begin{aligned} r_{12} &= \sin q_1 \\ r_{22} &= -\cos q_1 \end{aligned}\right\} q_1 = \tan^{-1} 2(r_{12}, -r_{22}) \quad ..(7)$$

and

$$\left.\begin{aligned} r_{31} &= \cos q_3 \\ r_{33} &= \sin q_3 \end{aligned}\right\} q_3 = \tan^{-1} 2(r_{31}, -r_{33}) \quad ..(8)$$

and for q_2 & q_4

$$\left.\begin{aligned} p_x &= (q_4 + d_4)\cos q_1 \cos q_3 \\ q_4 &= \frac{p_x - d_4 \cos q_1 \cos q_3}{\cos q_1 \cos q_3} \end{aligned}\right\} \quad ..(9)$$

or

$$\left.\begin{aligned} p_y &= (q_4 + d_4)\sin q_1 \cos q_3 \\ q_4 &= \frac{p_y - d_4 \sin q_1 \cos q_3}{\sin q_1 \cos q_3} \end{aligned}\right\} \quad ..(10)$$

and

$$\left.\begin{aligned} p_z &= (q_4 + d_4)\sin q_3 + q_2 + d_1 \\ q_2 &= p_z - (q_4 + d_4)\sin q_3 - d_1 \end{aligned}\right\} \quad ..(11)$$

Equations (7) through (11) are the general solutions of inverse kinematics.

Path Trajectory

1- Verification

It must be noted that the singularity algorithms listed in this section are presented in the work of Abdel-Malek[2,3]. The position vector of a point on the end-effector of a serial manipulator can be written in the terms of joint coordinates as:

$$X = \Phi\,(q) \quad ..(12)$$

where $q \in R^n$ and $\Phi\,(q)$ can be obtained from the forward kinematics *DH* conversion which can be written as:

$$T_n^0 = \begin{bmatrix} R_n^0 & \Phi(q) \\ 0 & 1 \end{bmatrix} \quad ..(13)$$

End-effector velocities can be determined by deriving eq.(12) w.r.t. time:

$$\dot{X} = \Phi_q\, \dot{q} \quad ..(14)$$

where $\Phi_q = \partial\Phi_i / \partial q_j$, (i, j: 0 $\rightarrow$ n). Define a subvector p_i of q as a set of constant generalized coordinates $p_i \in R^m$ where $m \leq n$-1, and $q = u \cup p_i$. Singular sets p_i can be obtained from studying the rank-deficiency of the Jacobian matrix. Three singularity types are identified:

1- Jacobian singularities: this is obtained when no joints reach their limits and they satisfy the following eq.:

$S^{(1)}$ = {$p_i \in R^m$: Rank[Φ_q] < 3, for some constant subset of q} ..(15)

2- Singularity sets characterized by the null space criterion imposed on the reduced-order manipulator i.e. some joints reach their limits. These sets satisfy the eq.:

$S^{(2)}$ = { $p_i \in R^m$: dim[null($\Phi_{q^*}^T (q^*)$)] $\geq$ 1, for some constant subset of q} ..(16)

Φ_{q^*} denotes the Jacobian after reducing the order of the manipulator (substituting a joint limit). and,

3- Singularity sets defined by a combination of all constant generalized coordinates:

$S^{(3)}$ = { $p_i \in R^{n\text{-}2}$: [$q_i^{\text{limit}}, q_j^{\text{limit}}$], for i, j : $1 \rightarrow n$; $i \neq j$} ..(17)

Substituting these singular sets into the position vector given by eq.(12) yields singular surfaces and curves parameterized by $\Psi(u)$ such that:

$$\Phi(u, p_i) = \Psi(u) \quad ..(18)$$

The position vector of a point on the end-effector of the spatial manipulator shown in figure (1) is:

$$\Phi(q) = \begin{bmatrix} (q_4 + d_4)\cos q_1 \cos q_3 \\ (q_4 + d_4)\sin q_1 \cos q_3 \\ (q_4 + d_4)\sin q_3 + q_2 + d_1 \end{bmatrix} \quad ..(19)$$

where $q = [q_1 \; q_2 \; q_3 \; q_4]^T$, and the Jacobian is derived as:

$$\Phi_q = \begin{bmatrix} -(q_4+d_4)s_1c_3 & 0 & -(q_4+d_4)c_1s_3 & c_1c_3 \\ (q_4+d_4)c_1c_3 & 0 & -(q_4+d_4)s_1s_3 & s_1c_3 \\ 0 & 1 & (q_4+d_4)c_3 & s_3 \end{bmatrix} \quad ..(20)$$

where $s_{1,3}$ & $c_{1,3}$ denote $\sin(q_{1,3})$ and $\cos(q_{1,3})$ respectively.

The Jacobian rank-deficiency is studied under the conditions of the singularity sets and the following results are found:

There are no singular sets due to Jacobian singularities because the results obtained from eq.(15) do not satisfy the joint constraints. Therefore, $S^{(1)}$ = null. Singularity sets defined by fixing one joint at its limit and solving eq.(16) are given by $S^{(2)} = \{p_1, p_2\}$ where $p_1 = (q_3 = 90^0, q_4 = 0)$ and $p_2 = (q_3 = 90^0, q_4 = 400)$. And finally, singularity sets resulting from the combinations of any two joints reaching their limits are $S^{(3)} = \{p_i, i = 3 \rightarrow 14\}$ where $p_3 = (q_2 = 0, q_3 = -75^0)$, $p_4 = (q_2 = 0, q_3 = 180^0)$, $p_5 = (q_2 = 0, q_4 = 0)$,
$p_6 = (q_2 = 0, q_4 = 400)$, $p_7 = (q_2 = 400, q_3 = -75^0)$, $p_8 = (q_2 = 400, q_3 = 180^0)$,
$p_9 = (q_2 = 400, q_4 = 0)$, $p_{10} = (q_2 = 400, q_4 = 400)$, $p_{11} = (q_3 = -75^0, q_4 = 0)$,
$p_{12} = (q_3 = -75^0, q_4 = 400)$, $p_{13} = (q_3 = 180^0, q_4 = 0)$, and $p_{14} = (q_3 = 180^0, q_4 = 400)$. Substituting each set of p_i, (i: 1 → 14) into eq.(19) yields singular surfaces in R^3 (Ψi) part of which are shown in figure (2) and the manipulator workspace which is shown in figure (3).

2- Generation (finding joint variables)

The aim of this work is finding the manipulator configurations (joint variables) along a straight path connecting an initial configuration to a goal one without using the inverse kinematic model, which may give uncertain solution or no solution. In the current implementation, the straight path between $[q_{intial}]$ & $[q_{goal}]$ is simulated as a line in R^3 space with parametric equations given by:

$$[X] = [X_{inital}] + [\Delta X] * t, \quad 0 \le t \le 1 \quad ..(21)$$

where $[\Delta X] = [X_{goal}] - [X_{intial}]$ ...(22)

$[X_{intial}]$ & $[X_{goal}]$ are the initial and goal Cartesian coordinate vectors defined by substituting $[q_{intial}]$ & $[q_{goal}]$ into eq.(19). By choosing a specific increment n_d, the path can be divided into n_d subintervals with end points ∈ line parametric equations. At each point, the manipulator's configuration can be determined as the following algorithm:
(k: 1 → n_d), $t = (k\text{-}1)/n_d$, and point coordinate vector is found from eq.(21).
According to point coordinate vector and figure (4), the first joint variable q_1 can be computed as:

$$(q_1)_k = \begin{cases} \tan^{-1}\left|\dfrac{y_k}{x_k}\right|, if\,(+x_k \;\&+y_k) \\ \pi - \tan^{-1}\left|\dfrac{y_k}{x_k}\right|, if\,(-x_k \;\&+y_k) \\ \pi + \tan^{-1}\left|\dfrac{y_k}{x_k}\right|, if\,(-x_k \;\&-y_k) \\ 2\pi - \tan^{-1}\left|\dfrac{y_k}{x_k}\right|, if\,(+x_k \;\&-y_k) \end{cases} \quad ..(23)$$

The second joint variable q_2 varies uniformly with the assumed increment:

$$(q_2)_k = q_{2intial} + (k\text{-}1) \cdot n_2 \quad ..(24)$$

where $n_2 = (\Delta q_2)/n_d$. Third joint variable q_3 is determined depending on the coordinate vector and q_2 as shown in figure (5):

$$(q_3)_k = \begin{cases} \tan^{-1}\left|\dfrac{(z)_k - ((q_2)_k + d_1)}{(d)_k}\right| \\ \pi - \tan^{-1}\left|\dfrac{(z)_k - ((q_2)_k + d_1)}{(d)_k}\right|, if\,(q_3)_1 \neq (q_{3\,intial}) \\ -\tan^{-1}\left|\dfrac{((q_2)_k + d_1) - (z)_k}{(d)_k}\right|, if\,(z)_k < (q_2)_k + d_1 \end{cases} \quad ..(25)$$

where $(d)_k = \sqrt{(x)^2{}_k + (y)^2{}_k}$. And finally, the fourth joint variable q_4 is computed using the following equation: (figure (6))

$$(q_4)_k = \sqrt{((z)_k - ((q_2)_k + d_1))^2 + (d)^2{}_k} - d_4 \quad ..(26)$$

The joint variables q_1, q_2, and q_3 have values that $\in$ joint constraints, but q_4 may go out the minimum joint constraint and to avoid this state, it is assumed that q_4 = 0 and new values of q_2 and q_3 are evaluated:

For $(q_3)_k < 0$, $q_4 = 0$,

$$(q_2)_k = (z)_k - \sqrt{(d_4)^2 - d_k^2} - d_1 \quad ..(27)$$

80

and reuse of eq.(25). Now all joint variables are known, but q_1 and q_3 must be updated according to the initial configuration q_{intial}. For q_1, if $(q_3)_k > 90^0$:

$$(q_1)_k = (q_1)_k - \pi \quad ..(28)$$

and for q_3, if $(q_3)_k < (q_3)_{k+1}$ & $\Delta q_3 < 0$, then :

$$(q_3)_{k+1} = \tan^{-1}\left|((z)_{k+1} - ((q_2)_{k+1} + d_1))/(d)_{k+1}\right| \quad ..(29)$$

as shown in figure (7).

If the straight path (line) that connects $[q_{intial}]$ & $[q_{goal}]$ $\in$ manipulator's work space, then equations (23) through (29) give the required joint variables that can make the end-effector follows this straight path and ensure that all joint variables $\in$ joint constraints. But if all or a part of it $\notin$ manipulator's workspace, then the initial configuration q_i must be changed so the path can be tracked. In this work, to produce a new q_i, the following technique is presented: q_{3i} is changed to a new one so that the straight path between the new $[q_{intial}]$ & $[q_{goal}]$ be tangent to the semicircle that is a part of the manipulator boundary workspace, generated when $q_2 = q_4 = 0$ and $q_3 \in$ (q_3: $-75^0 \rightarrow 180^0$), in the two configurations plane. This technique gives two values of q_{3i}. Figure (8) shows the four probabilities that all or a part of the path lies out the manipulator's workspace. From figure (8), the following calculations can be made to find out if the path $\notin$ workspace and produce a new $[q_{intial}]$ based on the above technique:

If $\{(q_{3i} > 90^0$ & $\Delta q_3 > 0$ & $(z)_k < d_1\}$ or $\{(q_3)_k < -75^0\}$ or $\{x_k^2 + y_k^2 - (z_k - d_1)^2 < d_4^2\}$, then some or all determined joint variables may $\notin$ joint constraints (i.e. all or a part of the path $\notin$ workspace), therefore, a new q_{3i} is generated as listed below:

79

$$\left.\begin{aligned} D_p &= \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} \\ D_i &= d_4 + q_{4i} \\ D_g &= d_4 + q_{4g} \\ D_{ci} &= \sqrt{D_i^2 - d_4^2} \\ D_{cg} &= \sqrt{D_g^2 - d_4^2} \end{aligned}\right\} \quad ...(30)$$

since the angles α_1 and α_2 are always > 90^0, it can be calculated as:

$$\left.\begin{aligned}\alpha_1 &= \sin^{-1}(d_4/D_g)\\ \alpha_2 &= \cos^{-1}((D_p^{\,2}+D_g^2-D_i^2)/2\cdot D_p\cdot D_g)\end{aligned}\right\} \quad \ldots(31)$$

based on the values of α_1 and α_2, the lengths of AD are computed:

$$\left.\begin{aligned}AD_B^2 &= (D_p^{\,2}+(D_{ci}+D_{cg})^2-2\cdot D_p\cdot(D_{ci}+D_{cg})\cdot\cos(\alpha_1+\alpha_2))\\ AD_S^2 &= (D_p^{\,2}+(D_{ci}+D_{cg})^2-2\cdot D_p\cdot(D_{ci}+D_{cg})\cdot\cos(\alpha_1-\alpha_2))\end{aligned}\right\} \quad .(32)$$

also the new values of q_3 are always less than 90^0, therefore,

$$\left.\begin{aligned}q_{3B} &= \cos^{-1}((2\cdot D_i^2-AD_B^2)/2\cdot D_i^2)\\ q_{3S} &= \cos^{-1}((2\cdot D_i^2-AD_S^2)/2\cdot D_i^2)\end{aligned}\right\} \quad ..(33)$$

now, the new values of [q_{intial}] can be generated by editing the values of q_{3i} :

$$\left.\begin{aligned}q_{3iB} &= q_{3i}+\left(\frac{|\Delta q_3|}{\Delta q_3}\right)\cdot q_{3B}\\ q_{3iS} &= q_{3i}+\left(\frac{|\Delta q_3|}{\Delta q_3}\right)\cdot q_{3S}\end{aligned}\right\} \quad ..(34)$$

the form of eq.(34) ensures that the new calculated values of q_{3i} are edited corresponding to the sign of Δq_3. The choice of q_{3i} (q_{3i}: q_{3iB} or q_{3iS}), that satisfy the above technique, is made by introducing a parameter called I_{test}. When the two values of q_{3i} are generated, the method uses q_{3iS} first to produce [q_{intial}], then all joint variables are calculated if any value of [q] $\not\in$ joint constraints, which means that the path $\not\in$ workspace, then q_{3iB} is submitted to determine [q_{intial}].

Figure (9) shows a model of the spherical manipulator that was manufactured to help in building and applying the presented method. Four different sets of [q_{intial}] & [q_{goal}] are used as inputs to the method for testing and table (2) shows the results. The method flowchart is shown in figure (10).

Conclusions

In this paper, a method is built for determining the joint variables of a spherical manipulator with 4-DOF end-effector to track a straight path between two given configurations. In this method, when all or a part of the path lies out the workspace, a new initial configuration is generated. All singularity surfaces of the manipulator workspace are also determined. The presented method always gives a suitable unique solution. The exist of singular surfaces in the manipulator workspaces does not affect the solution because the method computations depend on dividing the path between [q_{intial}] & [q_{goal}] into subintervals at which all joint variables are calculated. The method can be improved to make the current manipulator tracks any known paths. For same calculations, the number of inputs in this method is less than general inverse kinematics since the last one needs the DH matrix at each point for the same path.

References

1- Donelan P. S. “Singularity-theoretic methods in robot kinematics”, School of mathematics, Statistics and computer Science, University of Wellington, New Zealand, 2006.
2- Abdel-Malek, K., Yeh H. J, and Khairallah N. “Workspace, void, and volume determination of the general 5DOF manipulator”, Mechanics of Structures and Machines, vol. 27, No. 1, pp. 91-117, 1999.
3- Abdel-Malek, K., Yeh H. J, and Othman S. “Interior and exterior boundaries to the workspace of mechanical manipulators”,

Robotics and Computer-Integrated Manufacturing, in press.

4- Yueshi S. and Kunt H. "Optimal joint trajectory planning for manipulator robot performing constrained motion tasks", Department of Information Engineering and National ICT Australia Limited, Australian National University, Australia, 2002.

5- Rosales E. M., Gan J. Q., and Oyama E., "A Hybrid Approach to Inverse Kinematics Modeling and Control of Pioneer 2 Robotic Arms", Technical report, University of Essex, UK, 2003.

6- Qin C. and Miguel A., "Trajectory Inverse Kinematics by Conditional Density Modes", EECS, School of Engineering, University of California, USA, 2007.

7- Acosta C., A. and Hu H., "Trajectory Generation and Tracking of a 5-DOF Robotic Arm", Department of Computer Science, University of Essex, UK, 2004.

8- Spong, M. and Vidyasagar, M., "Robot dynamics and control", 2-nd edition, Prentice Hall, USA, 343p, 2004.

9- Abdel-Malek, K., Adkins F., and Haug E., "On The Determination of Boundaries to Manipulator Workspaces", Robotics and Computer-Integrated Manufacturing, Vol. 13, No. 1, pp. 1-15, 1997.

Table (1): DH parameters for the 4-link spherical manipulator

Link	a_i (mm)	α_i	d_i (mm)	θ_i
1	0	0^0	d_1=30	q_1
2	0	90^0	q_2	0^0
3	0	90^0	0	$q_3 + 90^0$
4	0	0^0	$q_4 + d_4$	0
q_i = joint variable, d_4 = 30				

Table (2): The results of four sets of different [$q_{initial}$] & [q_{goal}] used as inputs to the method, n_d = 11.

Point coordinates generated from the parametric equation (eq. (21)).			The corresponding generated joint variables.				The tracking point coordinates computed by substituting the corresponding generated joint variables into the forward kinematics (eq.(19)).		
x_p (mm)	y_p (mm)	z_p (mm)	q_1 (degree)	q_2 (mm)	q_3 (degree)	q_4 (mm)	x_t (mm)	y_t (mm)	z_t (mm)
٨,٧٥٠٠	-١٥,١٥٥٤	٧٠,٣١٠٩	120	١٠,٠٠٠٠	١٢٠,٠٠٠٠	٥,٠٠٠٠	٨,٧٥٠٠	-١٥,١٥٥٤	٧٠,٣١٠٩
١,٨٩٧٨	-٣,٢٨٧١	٧٤,٨٢٥٧	120	١٥,٠٦٦٨	٩٧,٢٦٨٧	٠	١,٨٩٧٨	-٣,٢٨٧١	٧٤,٨٢٥٧
-٤,٩٥٤٣	٨,٥٨١٢	٧٩,٣٤٠٤	120	٢١,٠٢٤١	٧٠,٧١٣٧	0	-٤,٩٥٤٣	٨,٥٨١٢	٧٩,٣٤٠٤
-١١,٨٠٦٥	٢٠,٤٤٩٥	٨٣,٨٥٥٢	120	٢٩,٥٠٠٠	٤٥,٨٨٦٥	٣,٩٢٢٧	-١١,٨٠٦٥	٢٠,٤٤٩٥	٨٣,٨٥٥٢
-١٨,٦٥٨٧	٣٢,٣١٧٨	٨٨,٣٧٠٠	120	٣٦,٠٠٠٠	٣٠,٩٤٠٧	١٣,٥٠٨٦	-١٨,٦٥٨٧	٣٢,٣١٧٨	٨٨,٣٧٠٠
-٢٥,٥١٠٨	٤٤,١٨٦١	٩٢,٨٨٤٨	120	٤٢,٥٠٠٠	٢١,٧٧٨٣	٢٤,٩٤٣٢	-٢٥,٥١٠٨	٤٤,١٨٦١	٩٢,٨٨٤٨
-٣٢,٣٦٣٠	٥٦,٠٥٤٤	٩٧,٣٩٩٦	120	٤٩,٠٠٠٠	١٥,٨٦٨٨	٥٠,١٢٩٦	-٣٢,٣٦٣٠	٥٦,٠٥٤٤	٩٧,٣٩٩٦
-٣٩,٢١٥٢	٦٧,٩٢٢٧	١٠١,٩١٤٣	120	٥٥,٥٠٠٠	١١,٨٢٠٦	٣٧,٢٩٠٤	-٣٩,٢١٥٢	٦٧,٩٢٢٧	١٠١,٩١٤٣
-٤٦,٠٦٧٣	٧٩,٧٩١٠	١٠٦,٤٢٩١	120	٦٢,٠٠٠٠	٨,٩٠٠٧	٦٣,٢٥٧٧	-٤٦,٠٦٧٣	٧٩,٧٩١٠	١٠٦,٤٢٩١
-٥٢,٩١٩٥	٩١,٦٥٩٣	١١٠,٩٤٣٩	120	٦٨,٥٠٠٠	٦,٧٠٥٧	٧٦,٥٦٨١	-٥٢,٩١٩٥	٩١,٦٥٩٣	١١٠,٩٤٣٩
-٥٩,٧٧١٧	١٠٣,٥٢٧٦	١١٥,٤٥٨٧	120	٧٥,٠٠٠٠	٥,٠٠٠٠	٩٠,٠٠٠٠	-٥٩,٧٧١٧	١٠٣,٥٢٧٦	١١٥,٤٥٨٧
[$q_{initial}$] = [120^0 10 120^0 5], [q_{goal}] = [120^0 75 5^0 90], the straight path that connects [$q_{initial}$] & [q_{goal}], lies inside the workspace. $q_3)_{new} = 0^0$.									

x_p (mm)	y_p (mm)	z_p (mm)	q_1 (degree)	q_2 (mm)	q_3 (degree)	q_4 (mm)	x_t (mm)	y_t (mm)	z_t (mm)
٣,٥٦٣٥	-٢٠,٢٠٩٤	٢٠٦,٣٨١٦	١٠٠,٠٠٠٠	١٢٠	١١٠,٠٠٠٠	٣٠,٠٠٠٠	٣,٥٦٣٥	-٢٠,٢٠٩٤	٢٠٦,٣٨١٦
١٧,٤٧٥٩	-١٥,٦٧٢٥	١٨٨,٨٦١١	٣١٨,١١٤٠	١١٢	٦٣,٣٩٢٣	٢٢,٤١١٨	١٧,٤٧٥٩	-١٥,٦٧٢٥	١٨٨,٨٦١١
٣١,٣٨٨٣	-١١,١٣٥٦	١٧١,٣٤٠٧	٣٤٠,٤٦٦٩	١٠٤	٤٨,٢٦٩٤	٢٠,٠٣٥٥	٣١,٣٨٨٣	-١١,١٣٥٦	١٧١,٣٤٠٧
٤٥,٣٠٠٧	-٦,٥٩٨٧	١٥٣,٨٢٠٢	٣٥١,٧١٢٣	٩٦	٣١,٢٨٧٥	٢٣,٥٦٩٣	٤٥,٣٠٠٧	-٦,٥٩٨٧	١٥٣,٨٢٠٢
٥٩,٢١٣٢	-٢,٠٦١٨	١٣٦,٢٩٩٨	٣٥٨,٠٠٥٨	٨٨	١٧,١٦٣٩	٣٢,٠١٠٧	٥٩,٢١٣٢	-٢,٠٦١٨	١٣٦,٢٩٩٨
٧٣,١٢٥٦	٢,٤٧٥١	١١٨,٧٧٩٤	١,٩٣٨٦	٨٠	٦,٨٤٢٢	٤٣,٦٩٢٣	٧٣,١٢٥٦	٢,٤٧٥١	١١٨,٧٧٩٤
٨٧,٠٣٨٠	٧,٠١٢٠	١٠١,٢٥٨٩	٤,٦٠٦٠	٧٢	-٠,٤٨٦٣	٥٧,٣٢٣١	٨٧,٠٣٨٠	٧,٠١٢٠	١٠١,٢٥٨٩
١٠٠,٩٥٠٤	١١,٥٤٨٩	٨٣,٧٣٨٥	٦,٥٢٦٤	٦٤	-٥,٧٦٦٨	٧٢,١٢٥٧	١٠٠,٩٥٠٤	١١,٥٤٨٩	٨٣,٧٣٨٥
١١٤,٨٦٢٨	١٦,٠٨٥٩	٦٦,٢١٨٠	٧,٩٧٢١	٥٦	-٩,٦٧٩١	٨٧,٦٥٨٦	١١٤,٨٦٢٨	١٦,٠٨٥٩	٦٦,٢١٨٠
١٢٨,٧٧٥٣	٢٠,٦٢٢٨	٤٨,٦٩٧٦	٩,٠٩٨٤	٤٨	-١٢,٦٦٣١	١٠٣,٦٦٧٥	١٢٨,٧٧٥٣	٢٠,٦٢٢٨	٤٨,٦٩٧٦
١٤٢,٦٨٧٧	٢٥,١٥٩٧	٣١,١٧٧١	١٠,٠٠٠٠	٤٠	-١٥,٠٠٠٠	١٢٠,٠٠٠٠	١٤٢,٦٨٧٧	٢٥,١٥٩٧	٣١,١٧٧١
[$q_{initial}$] = [100^0 120 100^0 30], [q_{goal}] = [10^0 40 -15^0 120], the straight path that connects [$q_{initial}$] & [q_{goal}], lies inside the workspace. $q_3)_{new} = 0^0$.									

x_p (mm)	y_p (mm)	z_p (mm)	q_1 (degree)	q_2 (mm)	q_3 (degree)	q_4 (mm)	x_t (mm)	y_t (mm)	z_t (mm)
			350	٥,٠٠٠٠	١٢٠,٠٠٠٠	١٥,٠٠٠٠			
			350	٨,٦٥٢٣	٦٢,٧٢٠٤	0			
			350	-٢,٩٢٠١	٨٠,٤٠٥٩	0			
			350	-١١,٧٥٩٩	٨٢,٨١٩٢	0			
			350	-١٧,٩٢٢٤	٦٥,٣٧٥٧	٠			
			350	٣٧,٥٠٠٠	-٦٠,٠٠٠٠	١٢,٥٠٠٠			
			350	٤٤,٠٠٠٠	-٦٠,٠٠٠٠	٣٠,٠٠٠٠			
			350	٥٠,٥٠٠٠	-٦٠,٠٠٠٠	٤٧,٥٠٠٠			
			350	٥٧,٠٠٠٠	-٦٠,٠٠٠٠	٦٥,٠٠٠٠			
			350	٦٣,٥٠٠٠	-٦٠,٠٠٠٠	٨٢,٥٠٠٠			
			350	٧٠,٠٠٠٠	-٦٠,٠٠٠٠	١٠٠,٠٠٠٠			

[$q_{initial}$] = [350^0 5 120^0 15], [q_{goal}] = [350^0 70 -60^0 100], a part of the straight path that connects [$q_{initial}$] & [q_{goal}], lies outside the workspace, therefore, $q_3)_{new} = 53.68^0$ is calculated and add to the q_{3i} to form a new [$q_{initial}$] = [350^0 5 ٦٦,٣٢^0 15]

x_p (mm)	y_p (mm)	z_p (mm)	q_1 (degree)	q_2 (mm)	q_3 (degree)	q_4 (mm)	x_t (mm)	y_t (mm)	z_t (mm)
-٩,٠٣٦٦	١٥,٦٥١٩	٧٦,٢١١١	350	٥,٠٠٠٠	٦٦,٣٢٠٠	١٥,٠٠٠٠	-٩,٠٣٦٦	١٥,٦٥١٩	٧٦,٢١١١
-١١,٣٨٣0	١٩,٧١٥٩	٦٧,٣٣١٧	350	١١,٥٠٠٠	٤٨,٦٠٩٧	٤,٤٣٢٠	-١١,٣٨٣0	١٩,٧١٥٩	٦٧,٣٣١٧
-١٣,٧٢٩٣	٢٣,٧٧٩٩	٥٨,٤٥٢٢	350	١٦,٣٦٨٢	٢٣,٧٥٣٥	٠	-١٣,٧٢٩٣	٢٣,٧٧٩٩	٥٨,٤٥٢٢
-١٦,٠٧٥٦	٢٧,٨٤٣٨	٤٩,٥٧٢٨	350	٢٤,٥٠٠٠	-٨,٧١٢٨	٢,٥٢٦٦	-١٦,٠٧٥٦	٢٧,٨٤٣٨	٤٩,٥٧٢٨
-١٨,٤٢٢٠	٣١,٩٠٧٨	٤٠,٦٩٣٤	350	٣١,٠٠٠٠	-٢٨,٨٦١٥	١٢,٠٦٩٤	-١٨,٤٢٢٠	٣١,٩٠٧٨	٤٠,٦٩٣٤
-٢٠,٧٦٨٣	٣٥,٩٧١٨	٣١,٨١٣٩	350	٣٧,٥٠٠٠	-٤٠,٦٦٧٤	٢٤,٧٦١٢	-٢٠,٧٦٨٣	٣٥,٩٧١٨	٣١,٨١٣٩
-٢٣,١١٤٧	٤٠,٠٣٥٨	٢٢,٩٣٤٥	350	٤٤,٠٠٠٠	-٤٧,٨٤٥٧	٣٨,٨٨٢٨	-٢٣,١١٤٧	٤٠,٠٣٥٨	٢٢,٩٣٤٥
-٢٥,٤٦١٠	٤٤,٠٩٩٧	١٤,٠٥٥٠	350	٥٠,٥٠٠٠	-٥٢,٥٣٤٢	٥٣,٧١٣٧	-٢٥,٤٦١٠	٤٤,٠٩٩٧	١٤,٠٥٥٠
-٢٧,٨٠٧٣	٤٨,١٦٣٧	٥,١٧٥٦	350	٥٧,٠٠٠٠	-٥٥,٧٩٦٧	٦٨,٩٣٥٥	-٢٧,٨٠٧٣	٤٨,١٦٣٧	٥,١٧٥٦
-٣٠,١٥٣٧	٥٢,٢٢٧٧	-٣,٧٠٣٩	350	٦٣,٥٠٠٠	-٥٨,١٨٣٧	٨٤,٣٩٢١	-٣٠,١٥٣٧	٥٢,٢٢٧٧	-٣,٧٠٣٩
-٣٢,٥٠٠٠	٥٦,٢٩١٧	-١٢,٥٨٣٣	350	٧٠,٠٠٠٠	-٦٠,٠٠٠٠	١٠٠,٠٠٠	-٣٢,٥٠٠٠	٥٦,٢٩١٧	-١٢,٥٨٣٣

x_p (mm)	y_p (mm)	z_p (mm)	q_1 (degree)	q_2 (mm)	q_3 (degree)	q_4 (mm)	x_t (mm)	y_t (mm)	z_t (mm)
			٣٠٠,٠٠٠٠	٠	٨٠,٠٠٠٠	٣٠,٠٠٠٠			
			٣٢٠,٩٧٤٧	٤٠,٠٠٠٠	٧٧,٥٢٢٧	١١,٦٠٥٢			
			٣٤٥,٣٠٢٥	٧٣,٥٥٥٣	٧٢,٤٣٥٩	0			
			٥,٧٨٠٩	٩٥,٦٢٠٠	٦٩,٣٤٠٣	0			
			١٩,٧٩٧٢	١١٨,٢٢٥٢	٦٤,١٥٦٨	0			
			٢٨,٩٦٤٤	٢٠٠,٠٠٠٠	- ٦٤,١٧٠0	٦,٩٣٠٧			
			٣٥,١٣٤٨	٢٤٠,٠٠٠٠	-٦٩,٤٥٠٣	٢٥,٢٢٠٦			
			٣٩,٤٧٨٥	٢٨٠,٠٠٠٠	-٧١,٩٧٤٤	٤٣,٧٩٤٧			
			٤٢,٦٦٨٣	٣٢٠,٠٠٠٠	- ٧٣,٤٢٤٣	٦٢,٤٨١٨			
			٤٥,٠٩٦٣	٣٦٠,٠٠٠٠	-٧٤,٣٥٥٥	٨١,٢٢٥٠			
			٤٧,٠٠٠٠	٤٠٠,٠٠٠٠	-٧٥,٠٠٠٠	١٠٠,٠٠٠٠			

[$q_{initial}$] = [300^0 0 80^0 30], [q_{goal}] = [47^0 400 -75^0 100], a part of the straight path that connects [$q_{initial}$] & [q_{goal}], lies outside the workspace, therefore, $q_3)_{new}$ = ٤٣,٣٤٢٤^0 is calculated and add to the q_{3i} to form a new [$q_{initial}$] = [300^0 0 ٣٦,٦٥٧٦^0 30]

x_p (mm)	y_p (mm)	z_p (mm)	q_1 (degree)	q_2 (mm)	q_3 (degree)	q_4 (mm)	x_t (mm)	y_t (mm)	z_t (mm)
٢٤,٠٦٦٥	-٤١,٦٨٤٥	٦٥,٨٢١٩	٣٠٠,٠٠٠	0	٣٦,٦٥٧٦	٣٠,٠٠٠٠	٢٤,٠٦٦٥	-٤١,٦٨٤٥	٦٥,٨٢١٩
٢٣,٩٥٤٦	-٣٥,٠٠٥٣	٨٩,٦٨٢٧	١٢٤,٣٤٦٢	٤٠	١٥,١٢٨٦	١٦,٧٩٨٥	٢٣,٩٥٤٦	-٣٥,٠٠٥٣	٨٩,٦٨٢٧
٢٣,٨٤٢٦	-٢٨,٤٢٦١	١١٣,٥٤٣٤	٣٠٩,٩٨٨٥	٨٠	٥,٤٥٥٦	٧,٢٧٠٢	٢٣,٨٤٢٦	-٢٨,٤٢٦١	١١٣,٥٤٣٤
٢٣,٧٣٠٦	-٢١,٧٩٦٩	١٣٧,٤٠٤٢	٣١٧,٤٣٢١	١٢٠	-٢١,٣٥٠٩	٤,٥٩٦٢	٢٣,٧٣٠٦	-٢١,٧٩٦٩	١٣٧,٤٠٤٢
٢٣,٦١٨٧	-١٥,١٦٧٧	١٦١,٢٦٥٠	٣٢٧,٢٩١٨	١٦٠	-٤٥,٦٧١٢	١٠,١٦٩٦	٢٣,٦١٨٧	-١٥,١٦٧٧	١٦١,٢٦٥٠
٢٣,٥٠٦٧	-٨,٥٣٨٥	١٨٥,١٢٥٨	٣٤٠,٠٣٧١	٢٠٠	-٦٠,٨٦٨١	٢١,٣٧٢٨	٢٣,٥٠٦٧	-٨,٥٣٨٥	١٨٥,١٢٥٨
٢٣,٣٩٤٧	-١,٩٠٩٣	٢٠٨,٩٨٦٥	٣٥٥,٣٣٤٣	٢٤٠	-٦٨,٩٥٧٨	٣٥,٣٧٢٨	٢٣,٣٩٤٧	-١,٩٠٩٣	٢٠٨,٩٨٦٥
٢٣,٢٨٢٧	٤,٧١٩٩	٢٣٢,٨٤٧٣	١١,٤٥٩٧	٢٨٠	-٧٢,٨٨٥٧	٥٠,٧٢٧٣	٢٣,٢٨٢٧	٤,٧١٩٩	٢٣٢,٨٤٧٣
٢٣,١٧٠٨	١١,٣٤٩١	٢٥٦,٧٠٨١	٢٦,٠٩٥٧	٣٢٠	-٧٤,٥٤٠٦	٦٦,٧٩٣٩	٢٣,١٧٠٨	١١,٣٤٩١	٢٥٦,٧٠٨١
٢٣,٠٥٨٨	١٧,٩٧٨٣	٢٨٠,٥٦٨٩	٣٧,٩٤٢٥	٣٦٠	-٧٥,٠٤٠٥	٨٣,٢٧٠٠	٢٣,٠٥٨٨	١٧,٩٧٨٣	٢٨٠,٥٦٨٩
٢٢,٩٤٦٨	٢٤,٦٠٧٥	٣٠٤,٤٢٩٦	٤٧,٠٠٠٠	٤٠٠	-٧٥,٠٠٠٠	١٠٠,٠٠٠٠	٢٢,٩٤٦٨	٢٤,٦٠٧٥	٣٠٤,٤٢٩٦

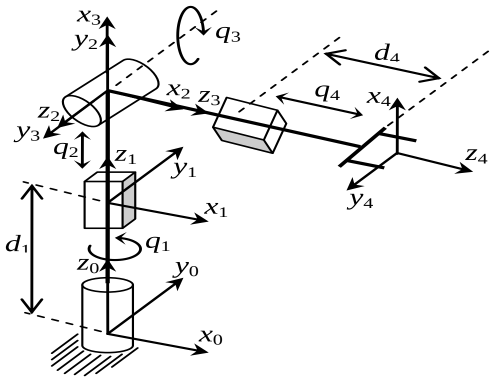
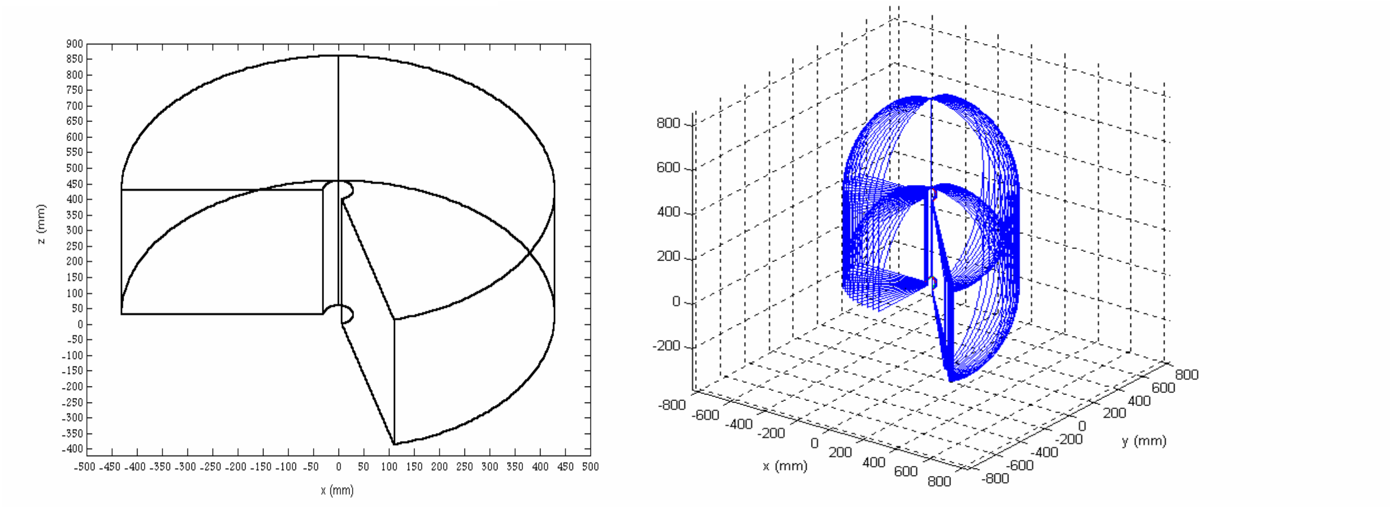
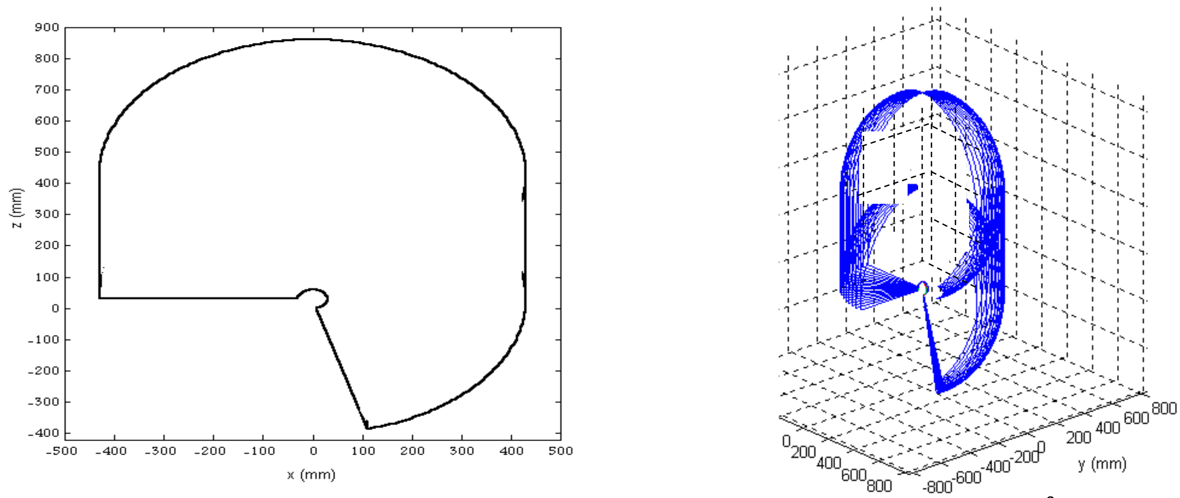


Figure (1): The spherical manipulator with frame assignments



a- In *x*-*z* plane ($\Delta q_1 = 0$) **b- In 3D (q_1: 0 → 90^0)**

Figure (2): Singularity surfaces



a- In *x*-*z* plane ($\Delta q_1 = 0$) **b- In 3D (q_1: 0 → 90^0)**

Figure (3): Manipulator's work space

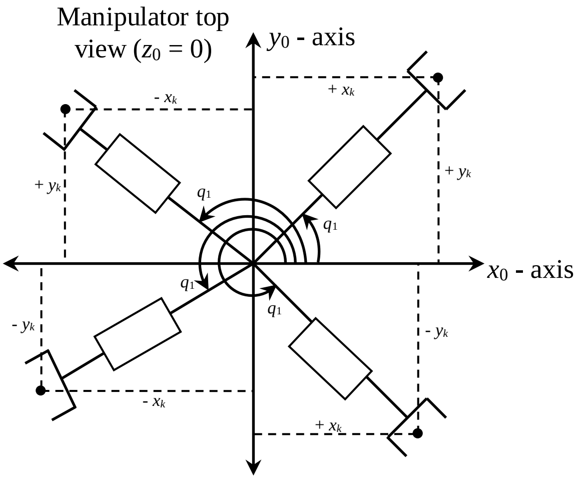


Figure (4): Calculations of q_1

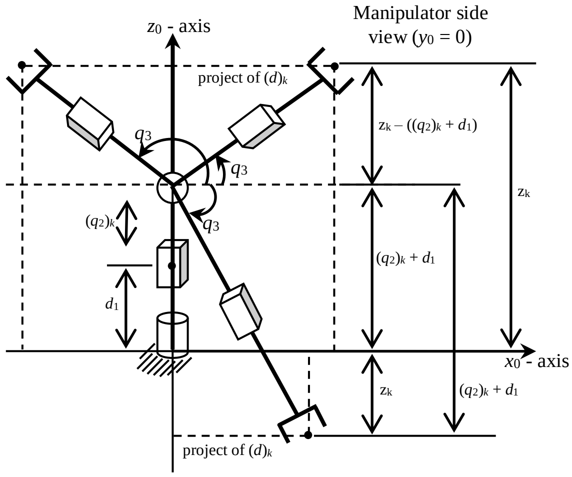


Figure (5): Calculations of q_3

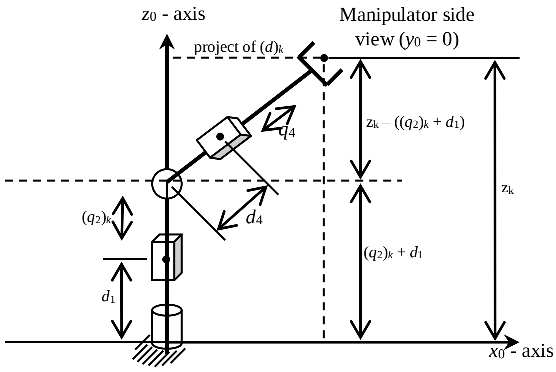


Figure (6): Calculations of q_4

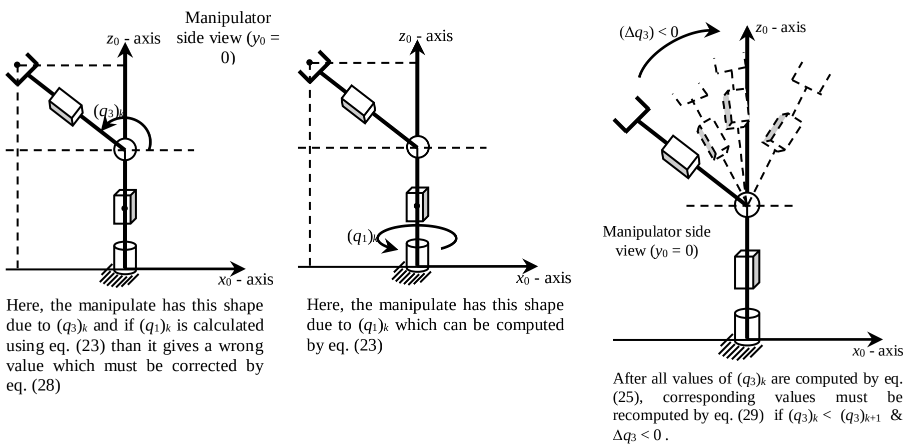


Figure (7): Recalculations of q_1 & q_3

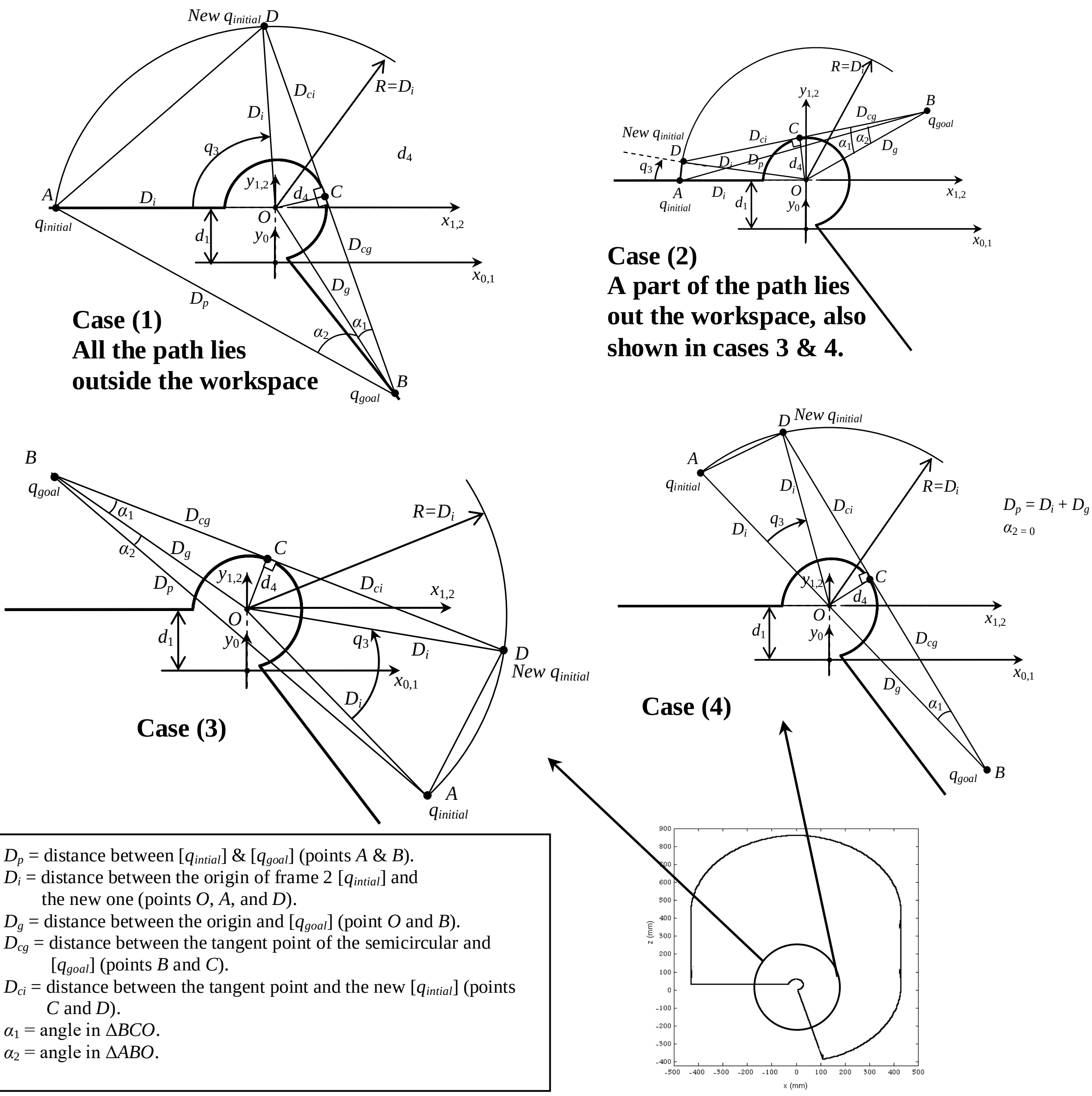


Figure (8): The four probabilities of the straight path

Zero reference point | $q_2 = 0$, $q_3 = 180^0$, $q_4 = 400$ | $q_2 = 0$, $q_3 = -75^0$, $q_4 = 400$ | $q_1 = 10^0$ $q_2 = 400$, $q_3 = 75^0$, $q_4 = 400$

Figure (9): The model of the spherical manipulator in different configurations

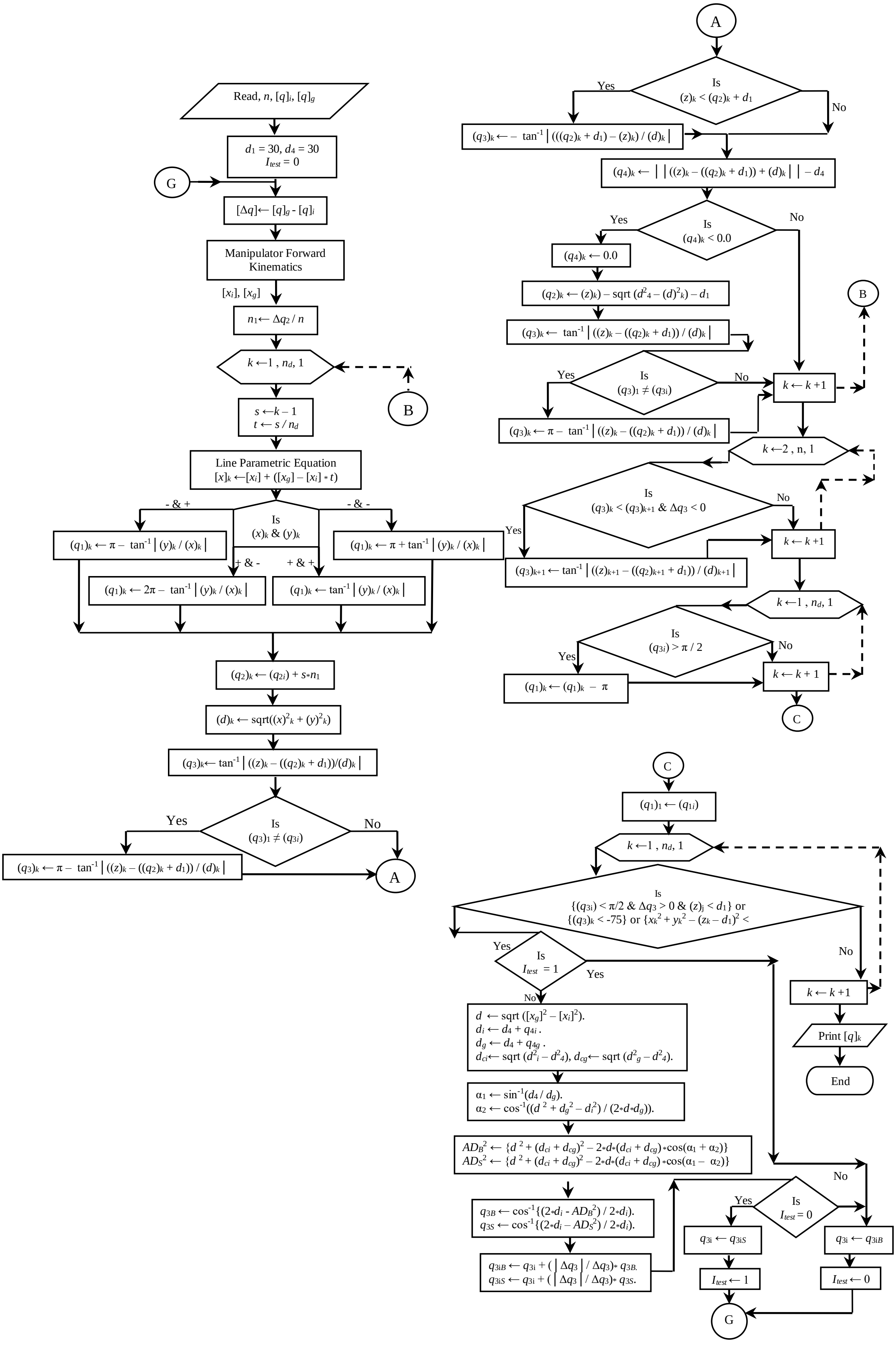


Figure (10): The method overall flow Chart