

Cylindrical Manipulator Path Planning Among Static Obstacles Using Artificial Potential Fields

Rawand Ehsan Jalal

(Assistant Lecturer)

College of Engineering – Kirkuk University

Abstract

In this paper, path planning for cylindrical manipulator of 4-DOF is studied. Another view point is presented for using so called 'artificial potential fields' which is used as the base of searching for new and safe points in the manipulator's workspace among static obstacles. Three vectors are used for safe manipulator navigation. The first vector is determined between the end-effector and the goal points and it is used to attract the end-effector to the goal point. While the second and third vectors are computed from points defined on the obstacle and the end-effector. These two vectors are used to repel the end-effector and the arm from the boundaries of the obstacles. In this work, the obstacles are suggested to have a cylindrical shape with different sizes. Displacement detections between the manipulator (its end-effector and arm) and the obstacles are used as sensors for collisions impending. A random movement is suggested for joint two to avoid contacting between the arm and obstacles. At the off – line path mode, all path points are determined by the presented method and some of them are updated, if an arm collision is detected, then joint variables are calculated at each point. In real mode, these joint variables are fed to a simple real control system to make the manipulator tracks the found path. The method gives a safe path for undertaken manipulator. An experimental work is also presented.

Keywords Path planning, cylindrical manipulator, static obstacles, artificial potential fields, configurations, collision detections, line parametric equations, control system.

تخطيط المسار لمعالج الإسطواني الحركة خلال عقبات ساكنة بإستعمال الحقول الكامنة الإصطناعية

روند إحسان جلال

(مدرس مساعد)

كلية الهندسة – جامعة كركوك

الملخص

خلال هذا البحث، تم دراسة تخطيط المسار لمعالج إسطواني الحركة ذات اربع درجات من الحرية. تم تقديم وجهة نظر اخرى لأستعمال ما يسمى ب" الحقول الإصطناعية الكامنة" التي تستعمل كأساس في عملية البحث عن نقاط جديدة وأمنة في فضاء عمل المعالج خلال العقبات الساكنة. أستعملت ثلاث متجهات لجعل ملاحه المعالج آمنة. المتجه الأول مُحَدَّدُ بين الطرف الفعال ونقطة الهدف وهو يُستعمل لجذب الطرف الفعال إلى نقطة الهدف بينما المتجهان الثانية والثالثة يتم حسابهما من نقاط على العارض و الطرف الفعال. هذين المتجهتين تُستعملان لمنع اصطدام الطرف الفعال والذراع مع العقبات. في هذا العمل، اقترح أن تكون للعقبات أشكال إسطوانية بأبعاد مختلفة. حساب المسافات بين المعالج (الطرف الفعال وذراع) والعقبات تم استخدامها كمتحسس لتجنب الإصطدامات. اقترحت الحركة العشوائية للمفصل الثاني لتجنب الاصطدام بين الذراع والعقبات. خلال نمط الحساب، كل نقاط المسار يتم حسابها بالطريقة المُقدَّمة والبعض من هذه النقاط يتم تجديدها وذلك في حالة اكتشاف اصطدام الذراع، ثم في كل نقطة يتم حساب قيم كل المفاصل. وفي النمط التطبيق الحقيقي، قيم كل المفاصل تصبح كإشارة دخول لنظام السيطرة الحقيقي لجعل المعالج يتعقب المسار المكتشف. الطريقة تُعطي مسار آمن للمعالج خلال العوارض المقترحة. تم إجراء تجارب عملية لتطبيق الطريقة المقترحة.

1- Introduction:

A robotic manipulator needs to be able to operate safely in cluttered, 3D environments. In order to achieve a manipulation task, a manipulator must be able to plan a path through its workspace to a goal location, while avoiding obstacles. Therefore, the robot path planning problem can be defined as the problem of finding a collision-free path between two specified configurations among obstacles. Motion or path planning has received much attention for the past two decades from robotics in order to:

- 1- Automatically generate the movements of mobile robots and their arms;
- 2- Automatically plan and program the motions of manufacturing robots and mechanical parts in assembling products.

Automation of path planning offers a number of advantages over the existing alternatives. It relieves human workers of the continual burden of detailed motion design and collision avoidance, and allows them to concentrate on the robotic tasks at a supervisory level. Robots with an automatic motion planner can accomplish tasks with fewer and higher-level operative commands.

Although humans have a superb capability to plan motions of their body and limbs effortlessly, the path planning problem turns out to be a high complexity problem[1]. A great number of different techniques has been and are still developed in order to carry out efficient robot path planning. One of the most popular path planning method is based on the potential functions utilization which is initially developed by Oussama Khatib in 1980. The basic idea behind the potential field approach is to treat the robot as a point particle in the configuration space under the influence of an artificial potential field U . The field U is constructed so that the robot is attracted to the final configuration while being repulsed from the obstacles. The negative gradient of the generated global potential field is interpreted as an artificial force acting on the robot and causing variation on its movement. Nevertheless, as a main presented drawback, these methods can result to a trapped robot in local minima generated by the same potential functions[2]. This limits the applicability of the artificial potential approach. Looking for a potential field without local minima has become a central concern in this approach. Harmonic functions are solutions to Laplace's equation. Such functions can be used as advantage for potential-field path planning, since they do not exhibit spurious local minima. This is proposed by Connolly C. and Grupen A.[3]. Harmonic functions that are presented in their work have a number of properties which are essential to robotics applications and the derived paths are generally smooth. They also show how these functions can be used as the basis for a reactive admittance control. Such schemes allow incremental updating of the environment model and respond well to sensed changes in the environment. However, the computational cost of the Laplace potential method will grow as an exponential function if the degree of freedom (DOF) of the robotic arms becomes larger[4]. A new methodology named Evolutionary Artificial Potential Field (EAPF) is introduced by Vadakkepat P., Tan K., and Liang W.[5] for real-time robot path planning. The suggested artificial potential field method is combined with genetic algorithms, to derive optimal potential field functions. To avoid local minima associated with EAPF, they produce an algorithm named escape-force. In their work, the potential field functions for obstacles contain tunable parameters and multi-objective evolutionary algorithm is also utilized to identify the optimal potential field functions. The results showed that the proposed methodology is efficient and robust for robot path planning with non-stationery goals and obstacles.

Wang Y. and Chirikjian G.[6] present a new artificial potential method. The model simulates steady state heat transfer with variable thermal conductivity, and then the optimal path problem is the same as a heat flow with minimal thermal resistance. The novelty of the presented technique

is that the thermal resistance in the configuration space for all different orientations of the robot can be superimposed and this reduce a search on $R^n \times SO(n)$ to one on R^n followed by a search on $SO(n)$.

Artificial potential field is not applied only to planning paths for serial manipulators; it has been applied to collision avoidance for humanoid robot arms by Sahara A, and Anzai[7] named “CAHRA”. There is a difficulty when robot developers make motions of humanoid robot arms, because the right and left arms may collide each other. CAHRA uses a potential method with very small computational cost and can avoid collision between its arms in 97% without being nervous. Shimoda S. and Iagnemma K.[8] propose a potential field-based method for high speed navigation of unmanned ground vehicles (UGVs) on uneven terrain. The potential field is generated in the two-dimensional “trajectory space” of the UGV path curvature and longitudinal velocity. The proposed method is subjected to local maximum problems, rather than local minimum. The presented method succeeded in navigating a UGV between pre-defined waypoints at high speed, while avoiding unknown hazards.

Pervious approaches have made the path planning with artificial potential fields highly complex by employing distance functions that cause many local minima in the search methods for global minimum (the goal point). In this paper instead of using functions, vector methods have been used for searching for new and safe points in the manipulator’s workspace and avoiding collision with the obstacles that are found in the workspace. Three vectors are suggested. The first one is calculated between the end-effector and the goal points and it is used to attract the end-effector to the goal point, the second one is produced between the probable contacting point on the obstacle’s surface in the direction of the first vector and the end-effector point and it is used to repel the end-effector from the obstacle boundaries. Finally, third vector is found between the knowing (reference) point on the obstacle and the end-effector and it is used to repel the arm from the obstacle boundaries. Collision detection is a vital part of any path planner. Furthermore, because path planners spend most of their time on collision or distance queries, the efficiency of the collision detection algorithm will greatly affect the overall efficiency of the planner[9]. In this work, the collision detection between the manipulator (its end-effector and arm) and the surrounding obstacles is sensed by determining the distances between them, and this is done by finding the nearest point on the obstacle’s surface that collision would occur with it in the direction of the first vector and find the distance between it and the end-effector. For the arm, the third vector is used to find the distance perpendicular to the arm from the obstacle’s reference point. For multi-obstacles, the distances between the hand and each obstacle are found in the direction of the first vector and to reduce the computations, the above procedure is done for the closer one. In the beginning of the presented method, the straight line from the starting and goal configurations is checked, and if a part of the line is found to be outside the manipulator’s workspace, a new goal point is proposed and named a *virtual goal point* that connects the start and goal points with straight lines within the workspace.

This paper is organized as follows: kinematics and workspace of the manipulator are presented first while method principles and vector computations are introduced secondly. Then, collision detections and virtual goal point are explained. Focus on how joint variable computations and obstacle representations are done followed by Experimental works and results. Finally, conclusions are given.

2- Manipulator Forward Kinematics and Workspace:

2-1 Manipulator Forward Kinematics:

The coordinate frame assignment is depicted in Fig (1). Details about definitions of the coordinates and Denavit-Hartenberg (DH) parameters can be found in [10]. The current cylindrical manipulator has $a_4 = 0$ which differs from that is in [10] and this is done to reduce some computations. Following the DH methodology, the general transformation matrix (forward kinematics), which expresses the position and orientation of the gripper with respect to the arm's base (frame 0), is given as:

$$T_4^0 = \begin{bmatrix} -Cq_1Sq_4 & -Cq_1Cq_4 & -Sq_1 & -d_4Sq_1 - q_3Sq_1 + a_2Cq_1 \\ -Sq_1Sq_4 & -Sq_1Cq_4 & Cq_1 & d_4Cq_1 + q_3Cq_1 + a_2Sq_1 \\ -Cq_4 & Sq_4 & 0 & -a_3 + d_1 + q_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ..(1)$$

and the DH matrix that gives the position and orientation of the frame 2 with respect to the arm's base is:

$$T_2^0 = \begin{bmatrix} Cq_1 & 0 & -Sq_1 & a_2Cq_1 \\ Sq_1 & 0 & Cq_1 & a_2Sq_1 \\ 0 & -1 & 0 & d_1 + q_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ..(2)$$

which is useful in pervious sections. In matrices (1) and (2), Cq_i & Sq_i denote $\cos(q_i)$ and $\sin(q_i)$ respectively, and a_2, a_3, d_1, d_4 are constants and depend on manipulator dimensions and geometry. The joint variables are: the relative angles between links one and two q_1 , and between links three and four q_4 and links two and three extensions q_2 , & q_3 .

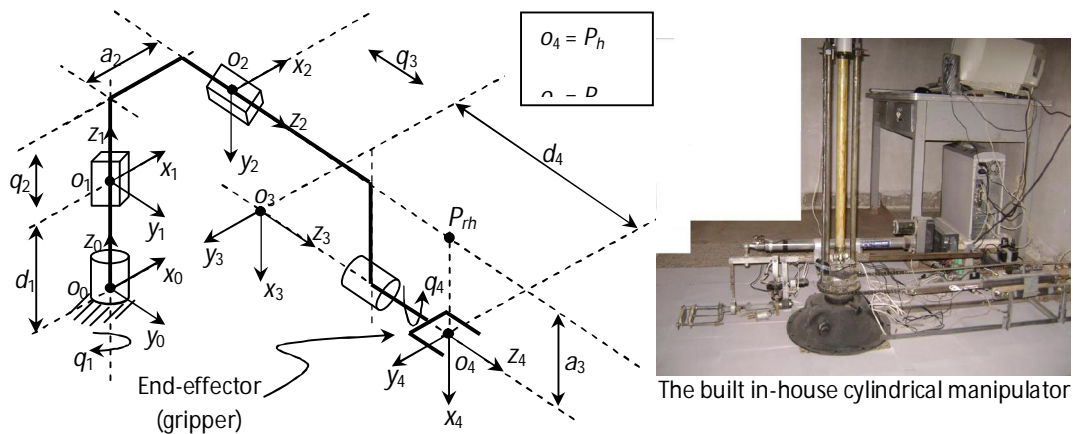


Fig (1) Reference Coordinate Assignments and the Built in-House 4-link 4-DOF Cylindrical Manipulator

2-2 Workspace:

The total workspace of a manipulator is defined as the space reachable by a reference point in the hand[11]. The forward kinematics DH conversion can be written as:

$$T_n^0 = \begin{bmatrix} R_n^0 & \Phi(q) \\ 0 & 1 \end{bmatrix} \quad ..(3)$$

where R_n^0 is a (3x3) rotational matrix and $\Phi(q)$ is a (3x1) translational vector. The position vector of a point on the end-effector of the manipulator can be written in the terms of joint coordinates as:

$$X = \Phi(q) \quad ..(4)$$

where X is a position vector in Cartesian space, $q \in R^n$, q is a vector of joint variables and n is the number of DOF. For the current cylindrical manipulator, Eq (4) can be written as:

$$\Phi(q) = \begin{bmatrix} -d_4 \cdot S q_1 - q_3 \cdot S q_1 + a_2 \cdot C q_1 \\ d_4 \cdot C q_1 + q_3 \cdot C q_1 + a_2 \cdot S q_1 \\ -a_3 + d_1 + q_2 \end{bmatrix} \quad ..(5)$$

where $q = [q_1 \ q_2 \ q_3]^T$ and its constraints are $0^\circ \leq q_1 \leq 190^\circ$, $0 \text{ mm} \leq q_2 \leq 431.8 \text{ mm}$ and $0 \text{ mm} \leq q_3 \leq 584.2 \text{ mm}$. To draw the manipulator workspace, each two joint variables are assumed to be in its limits and combined to produce a set as $[q_i^{limit}, q_j^{limit}]$, for $i, j: 1 \rightarrow n; i \neq j$. By substituting each set into Eq (5) and varying the third one, apart of the workspace boundary is drawn. Fig (2) shows the drawn workspace.

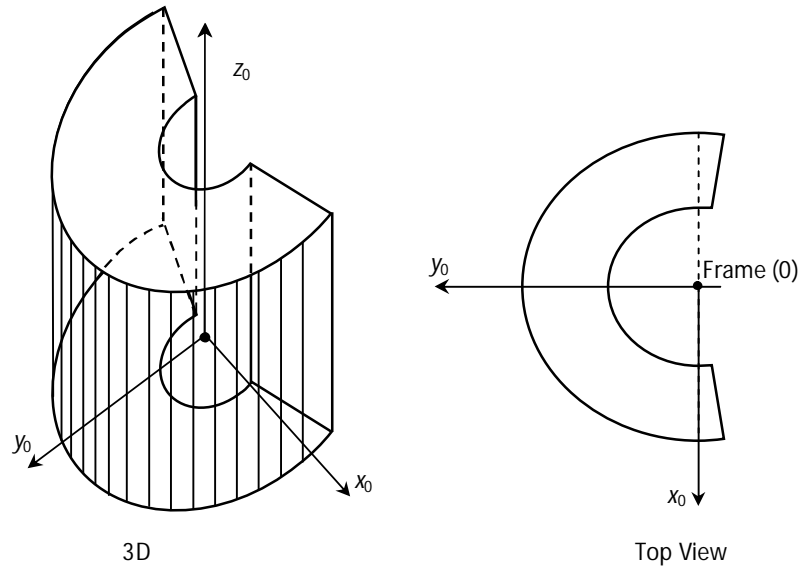


Fig (2) The undertaken cylindrical manipulator workspace

3- Method Principles:

In traditional artificial potential field methods, potential field U is assumed around the robot. When the robot is close to obstacles, the potential becomes large, and when the robot is close to goal point, the potential becomes smaller. Then the robot moves to the direction F which is the sum of an attraction vector $F_{attract}(x)$ to the goal and repulsion vector(s) $F_{rep}(x)$ from obstacles at the current position (see Fig (3)). In other words, the robot moves to the direction where the potential U is smaller. In this work, instead of using potential function which can be a function of distance (for $F_{attract}$ & F_{rep}), a vector is used so that there will not be need of an optimization search method that can easily trap into local minima. The vector that attracts the end-effector to the goal configuration is computed between the end-effector point (the origin point of frame 4) and the goal point, it always directs to goal point, as following:

$$\overrightarrow{P_h \cdot P_g} = (\Delta x)_{hg} i + (\Delta y)_{hg} j + (\Delta z)_{hg} k \quad ..(6)$$

where $[\Delta X]_{hg} = X_g - X_h$, X_g & X_h are coordinate vectors of the goal and current end-effector points. To repel the gripper from obstacle's boundaries, a second vector is used. This vector is defined between the end-effector point and the probable collision point on the obstacle's surface in the direction of the first vector and it directs from the obstacle to the gripper. Eq.(7) gives the repel vector.

$$\overrightarrow{P_o \cdot P_g} = (\Delta x)_{og} i + (\Delta y)_{og} j + (\Delta z)_{og} k \quad ..(7)$$

where $[\Delta X]_{og} = X_h - X_o$, P_o is the probable contacting point and X_o is its coordinate vector. Eq (7) is computed and starts affecting the solution when a certain distance between the two points reaches a given value. That is:

$$\overrightarrow{P_o \cdot P_h} = \begin{cases} \text{Eq (7) if } D_a \leq \rho \\ 0 \quad \text{if } D_a > \rho \end{cases} \quad ..(8)$$

where D_a is the distance between P_o & P_h and is given by Eq.(9) and ρ is the influencing distance of the obstacle.

$$D_a = \sqrt{(x_h - x_o)^2 + (y_h - y_o)^2 + (z_h - z_o)^2} \quad ..(9)$$

For each obstacle found in the manipulator's workspace and satisfies Eq (8), there will be a repel vector. After each vector is computed (including the third vector in the next section), the resultant of Eqs (6) & (7) is calculated and a new and save point can be found in the direction of the resultant as shown:

$$\vec{R} = \overrightarrow{P_h \cdot P_g} + \sum \overrightarrow{P_o \cdot P_h} \quad ..(10)$$

and

$$P_h)_{new} = \alpha_i \cdot \frac{\vec{R}}{|\vec{R}|} \quad ..(11)$$

where \vec{R} is the resultant vector, $P_h)_{new}$ is the new point (position) of the gripper, and α_i is a scalar determines the step size. It is important that α_i be small enough that the robot is not allowed to

jump into obstacles while being large enough that it does not require excessive computation time. Therefore, α_i is considered to have variable values depending on the distance to the nearest obstacle (Eq (9)) and empirical basis.

$$\alpha_i = \begin{cases} 0.2 & ; D_a \leq \rho \\ 1.2 & ; D_a > \rho \end{cases} \quad ..(12)$$

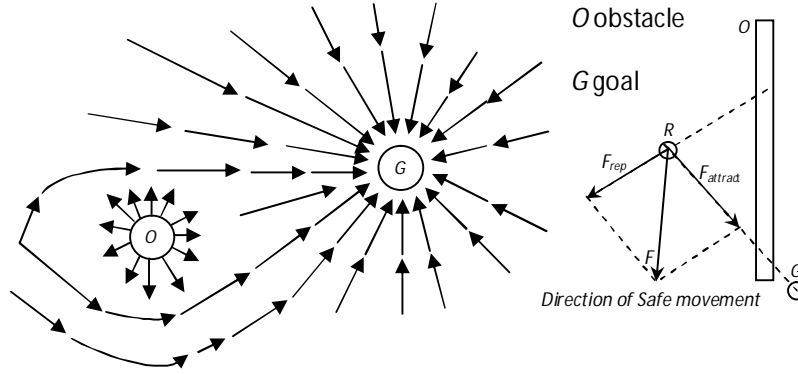


Fig (3) Evolutionary Artificial Potential Field and Resultant Direction

4- Collision Detections and Virtual Goal Point:

4-1 Collision Detections:

A collision occurs when the robot contacts an obstacle in its workspace. If this workspace, that is, the Cartesian space in which the robot moves, is denoted by W and the obstacle region, the set of all points in W that belong to an obstacle i , is denoted by \bar{O}_i , i.e. $\bar{O}_i \in W$. Also, denoting the Cartesian space by X , the manipulator by M and the sub set of the workspace that is occupied by the end-effector by $M(X)$. The set of Cartesian space for which the robot collides with an obstacle is referred to as the space obstacle and it is defined by:

$$X \bar{O} = \{X \in W \mid M(X) \cap \bar{O} \neq \emptyset\} \quad ..(13)$$

in which $\bar{O} = \bigcup \bar{O}_i$. The remaining portion of the Cartesian space is called the set of collision-free space and is simply the set:

$$X_{free} = X \setminus X \bar{O} \quad ..(14)$$

Note that the definition of space obstacle in Eq (13) includes arm-collisions without self-collisions, since; the undertaken cylindrical manipulator does not self-collide. The collision detections that are presented below are true if the obstacles, which have cylindrical shapes, stand with its base parallel to the x - y plane of manipulator reference frame (frame 0).

4-2 End-Effector Collision Detection:

The parametric equations of the line that joints P_g & P_h are calculated as:

$$[X] = [X_h] + [\Delta X]_{hg} \cdot t, \quad 0 \leq t \leq 1 \tag{15}$$

where t is a parameter. From Fig (4), the following computations can be done:

$$\left. \begin{aligned} D_{hg} &= \sqrt{(x_g - x_h)^2 + (y_g - y_h)^2 + (z_g - z_h)^2} \\ D_{Og} &= \sqrt{(x_g - x_o)^2 + (y_g - y_o)^2 + (z_g - z_o)^2} \\ \theta &= \cos^{-1} \left(\frac{\overrightarrow{P_h \cdot P_g} \cdot \overrightarrow{P_g \cdot P_o}}{|\overrightarrow{P_h \cdot P_g}| \cdot |\overrightarrow{P_g \cdot P_o}|} \right) \\ D_{Cg} &= D \cdot \cos(\theta) \\ D_{CO} &= D \cdot \sin(\theta) \\ P_C &= \left[(X_g + X_h) \cdot \frac{D_{Cg}}{D_{hg}} \right] \end{aligned} \right\} \tag{16}$$

where D_{hg} , D_{Og} are distances between P_g & P_h and P_g & P_{ro} respectively, P_{ro} is the obstacle's reference point, θ is the angle between $\overrightarrow{P_h \cdot P_g}$ & $\overrightarrow{P_g \cdot P_o}$. D_{Cg} & D_{CO} are the distance between a point (P_C) on $\overrightarrow{P_g \cdot P_h}$ to P_g and the smallest distance from P_C to the obstacle's (\bar{O}_i) reference point respectively and P_C is the coordinates of nearest point to P_{ro} . Now, if $P_C \in \bar{O}_i$, this means that the end-effector is going to have collision with \bar{O}_i in its direction toward P_g . Therefore, the closer point on the \bar{O}_i surface to the end-effector must be found. This is done by finding the parametric equations of the line that extends between P_C & P_h :

$$[X] = [X_C] + [\Delta X]_{hc} t, \quad 0 \leq t \leq 1 \tag{17}$$

where $[\Delta X]_{hc} = X_g - X_C$. By changing t in step of (0.01), a new point $P(k)$ on the $\overline{P_C \cdot P_h}$ is determined and if $P(k) \in \bar{O}_i$, a new point $P(k+1)$ is found and so on until a point $P(k+m)$ is computed that $\notin \bar{O}_i$, then, $P(k+m-1)$ is an approximation of the closer point which is denoted by P_o .

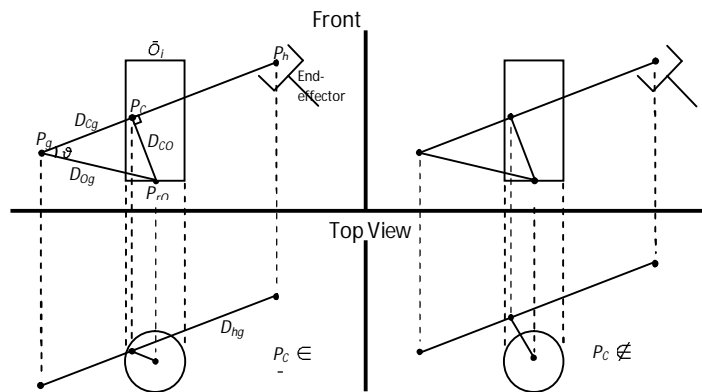


Fig (4) Two Cases of End-Effector Collision Detections

4-3 Arm Collision Detection:

From a point above P_h directly at distance of a_3 (Fig (5)), denoted by P_{rh} and P_{rO} on \bar{O}_i a vector is computed as:

$$\overrightarrow{P_{rh} \cdot P_{rO}} = (\Delta x)_{rh,rO} i + (\Delta y)_{rh,rO} j + (\Delta z)_{rh,rO} k \quad ..(18)$$

where $[\Delta X]_{rh,rO} = X_{rO} - X_{rh}$, and from P_{rh} and the original point of frame (2), P_{o2} , another vector is noted as:

$$\overrightarrow{P_{rh} \cdot P_{o2}} = (\Delta x)_{rh,o2} i + (\Delta y)_{rh,o2} j + (\Delta z)_{rh,o2} k \quad ..(19)$$

where $[\Delta X]_{rh,o2} = X_{o2} - X_{rh}$. P_{o2} is given by the first three elements in last column in Eq (2). Fig (5) shows the above and the additional following computations:

$$\left. \begin{aligned} \beta &= \cos^{-1} \left(\frac{\overrightarrow{P_{rh} \cdot P_{rO}} \cdot \overrightarrow{P_{rh} \cdot P_{o2}}}{|\overrightarrow{P_{rh} \cdot P_{rO}}| \cdot |\overrightarrow{P_{rh} \cdot P_{o2}}|} \right) \\ D_{rO,h} &= \sqrt{(x_{rh} - x_{rO})^2 + (y_{rh} - y_{rO})^2 + (z_{rh} - z_{rO})^2} \\ D_{rO,R} &= D_{Or,h} \cdot \sin(\beta) \\ D_{real} &= \sqrt{D_{rO,R}^2 - (z_h - z_{rO})^2} \end{aligned} \right\} \quad ..(20)$$

where β is the angle between $\overrightarrow{P_{rh} \cdot P_{rO}}$ & $\overrightarrow{P_{rh} \cdot P_{o2}}$, $D_{rO,h}$ is the distances between P_{rh} & P_{rO} , $D_{rO,R}$ is the perpendicular distance between P_{rO} and the arm and D_{real} is the projection of $D_{rO,R}$ on the x_0 - y_0 plane. After the shortest distance between the obstacle reference point and the manipulator arm is determined, it must be ensured that the point which locates above or down P_{rO} with distance equal to $\Delta z = (z_h - z_{rO})$, i.e. $(x_{rO}, y_{rO}, \Delta z)$, $\in \bar{O}_i$, Fig (5), if it does, and D_{real} be $\leq R_{Ovi}$, q_2 must be actuated to avoid collision of the arm, where $R_{Ovi} = \text{radius of } \bar{O}_i + F$ and F is a scalar. The movement of q_2 is suggested to be randomly as follows:

A random step from q_2 is obtained by randomly adding and subtracting a small fixed constant u to q_2 at same time:

$$q_2)_{\text{random-step}} = q_2 \pm i.u \quad ..(21)$$

where $i = 1,2,\dots,m$. At each direction, the arm collision is detected and if there is detection, Eq (21) is repeated with a step of $2u$ and so on until a collision in one direction is not detected. Then, that side is taken as a recommended movement for q_2 .

The above procedures for collision detections are repeated for every obstacle found inside the workspace. For multi-obstacles, the collision computations become very expensive in time and some computations are useless. To reduce the computations and save time, it has been suggested to have collision calculations in the direction of the first vector $\overrightarrow{P_h \cdot P_g}$ for end-effector detections

by assuming a cone around the vector with half angle of γ_h , Fig (6a). The angle between $\overrightarrow{P_h \cdot P_g}$ & $\overrightarrow{P_h \cdot P_{roi}}$ for each obstacle \bar{O}_i is found as:

$$\varphi_i = \cos^{-1} \left(\frac{\overrightarrow{P_h \cdot P_g} \bullet \overrightarrow{P_h \cdot P_{roi}}}{|\overrightarrow{P_h \cdot P_g}| \cdot |\overrightarrow{P_h \cdot P_{roi}}|} \right) \quad ..(22)$$

and if $\varphi_i \leq \gamma_h$ that \bar{O}_i is undertaken. γ_h has a critical value since it must reduce the computations and in same time it must not ignore obstacles. If a collision is detected for more that one obstacle, $D_{rO,h}$ is projected on $\overrightarrow{P_h \cdot P_g}$ for each \bar{O}_i , Eq (22), then obstacle with shortest distance is used for collision detection.

$$\left| \text{proj}_{\overrightarrow{P_h \cdot P_g}} \overrightarrow{P_h \cdot P_{Or}} \right| = \frac{\overrightarrow{P_h \cdot P_g} \bullet \overrightarrow{P_h \cdot P_{Or}}}{(\overrightarrow{P_h \cdot P_g})^2} \quad ..(23)$$

For arm collision, Fig (6) shows how obstacles are chosen. Every \bar{O}_i makes angle β_i that $\notin 0^\circ < \beta_i < 90^\circ$ is neglected.

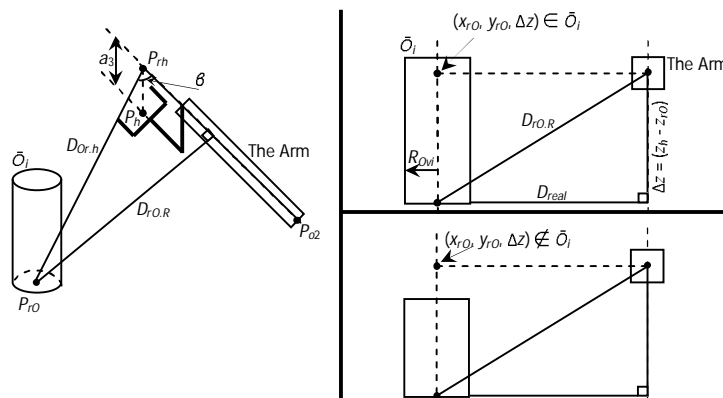
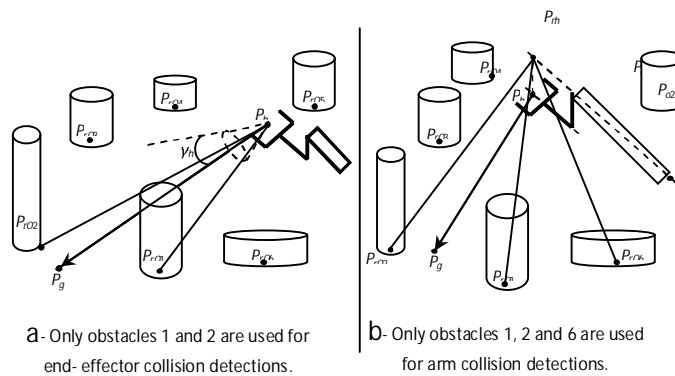


Fig (5) The Arm Collision Detection and Checking For Point $(x_{ri}, y_{ri}, \Delta z)$



a- Only obstacles 1 and 2 are used for end- effector collision detections.

b- Only obstacles 1, 2 and 6 are used for arm collision detections.

Fig (6) The Choice of Obstacles for Collision Detections

4-4 Virtual Goal Point:

When a part of the line that joining P_h & P_g , $\overline{P_h \cdot P_g}$ lies outside the manipulator workspace, the presented path planning approach fails to give a path because the points on the produced path $\notin W$, and to avoid that, a so called *virtual goal point* is suggested. This point is connected to P_h from side and with P_g from other side (Fig (7)). The lines that join the three points must be inside the workspace. For the current cylindrical manipulator, to find out if a part of $\overline{P_h \cdot P_g}$ lies outside the workspace, Eq (15) is used by eliminating the z - coordinate part and changing t at step of 0.1, then if $r(t) = \sqrt{[x(t)]^2 + [y(t)]^2}$ is $< r_{ib}$ a part of the line lies outside the workspace where r_{ib} is the radius of the inside workspace boundary. From Fig (7), the following computations can be done:

$$m = ((X_h + X_g)/2) \quad ..(24)$$

where m is the mid point of $\overline{P_h \cdot P_g}$. The parametric equations of lines that connect points: B & m , P_h & V_g , P_g & V_g are:

$$\left. \begin{aligned} [X_B] &= [X_m] + [\Delta X] \cdot t_m, & 0 \leq t_m \leq 1 \\ [X_{h.vg}] &= [X_h] + [\Delta X] \cdot t, & 0 \leq t \leq 1 \\ [X_{g.vg}] &= [X_g] + [\Delta X] \cdot t, & 0 \leq t \leq 1 \end{aligned} \right\} \quad ..(25)$$

where B is a point on the outside workspace boundary, V_g is the virtual goal point that produced after each changing in t_m , and $\Delta X = X_B - X_m$, $\Delta X = X_{Vg} - X_h$, $\Delta X = X_{Vg} - X_g$ respectively. At each changing of t_m , a new V_g is determined at which the lines $\overline{P_h \cdot V_g}$ and $\overline{P_g \cdot V_g}$ must be checked if they are $\in W$ and this is done as with $\overline{P_h \cdot P_g}$ using the last two parametric equations in Eq (24). In addition, V_g must $\notin \bar{O}_i$, therefore, V_g will be proper if:

$$\overline{P_h \cdot V_g} \ \& \ \overline{P_g \cdot V_g} \in W \ \text{and} \ V_g \notin \bar{O}_i \quad ..(26)$$

If conditions (25) are not satisfied for each value of t_m , then point m is replaced by:

$$m = ((X_h + X_g) \cdot n_d) \quad ..(27)$$

where $0 < n_d < 1$ is a random quantity. When the virtual goal point is presented, the path planning is divided into two parts; first part is from initial P_h point to V_g ($(P_g)_{\text{secondary}} = V_g$) where V_g plays the turn of goal point, and second part from V_g to P_g ($(P_g)_{\text{original}}$) where V_g represents the P_h point.

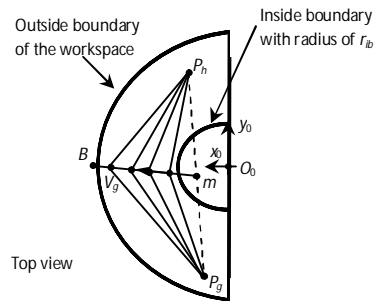


Fig (7) The Computations of Virtual Goal Point

5- Joint Variables and Obstacle Space:

5-1 Joint Variables Computations:

The aim of any path planning algorithm is finding the manipulator configurations (joint variables) along the determined path. In the current implementation, three joint variables might be calculated (q_1 , q_2 , and q_3) since q_4 is used for changing the orientation of the end-effector only as can be seen from Eq (1). The inverse kinematics (IK) is the inverse problem of finding joint variables in terms of the end-effector's position and orientation. The presented approach gives only the end-effect' position (Eq (11)); therefore, using IK is not possible. To solve this problem, the following simple methods have been used:

For q_1 : After $P_{h_{new}}$ has been found, two vectors can be drawn from frame (0) to P_h and $P_{h_{new}}$ denoted by

$\overrightarrow{P_{o1} \cdot P_h}$ & $\overrightarrow{P_{o1} \cdot P_{hm}}$ which can simply dotted to find q_1 without z - coordinates:

$$q_1 = \cos^{-1} \left(\frac{\overrightarrow{P_{o1} \cdot P_h} \cdot \overrightarrow{P_{o1} \cdot P_{hm}}}{\left| \overrightarrow{P_{o1} \cdot P_h} \right| \cdot \left| \overrightarrow{P_{o1} \cdot P_{hm}} \right|} \right) \quad ..(28)$$

For q_2 : Movement in the z_0 -direction is the responsibility of joint two only. Therefore, its value can be found from Eq (5) as:

$$q_2 = z_{hm} + a_3 - d_1 \quad ..(29)$$

For q_3 : Since q_1 and q_2 have been found, the second (or first) element in Eq (5) can be used to find q_3 .

$$\left. \begin{aligned} q_3 &= \frac{y_{hm} - d_4 \cdot \cos(q_1) - a_2 \cdot \sin(q_1)}{\cos(q_1)} \\ \text{or} \\ q_3 &= \frac{x_{hm} + d_4 \cdot \sin(q_1) - a_2 \cdot \cos(q_1)}{-\sin(q_1)} \end{aligned} \right\} \quad ..(30)$$

Always $0^0 \leq q_1 < 90^0$ (a_i is small) and if there is a division by zero ($q_1 = 0^0$) switching in using of Eq (29) is suggested.

5-2 Obstacle Space:

In last sections, X_{free} is determined among obstacles that are assumed to have cylindrical shapes with different sizes and are located in the manipulator's workspace with their bases parallel to x_0 - y_0 plane. All points that $\in W$ and \bar{O}_i , must be defined to the path planner so that $X \bar{O}$ be known. In this work, \bar{O}_i is simply represented by a circle equation with z - coordinate for height as:

$$\left. \begin{aligned} (x_i - g_i)^2 + (y_i - u_i)^2 &\leq R_i^2 \\ z_i &\leq H_i \end{aligned} \right\} \bar{O}_i \quad ..(31)$$

where (g_i, u_i) and R_i & H_i are the reference point, radius and height of \bar{O}_i , respectively. (g_i, u_i) is always assumed to be located at the base of \bar{O}_i and the sign \leq is used to ensure that all point $\in \bar{O}_i$ be represented.

6- Experimental Work and Results:

In this work, the manipulator movement from an initial configuration to final (goal) one is done among two modes. In the off – line path mode, a path is planned (if a one exists) through the given obstacles. At each point \in path, the three joint variables (manipulator configuration) are determined and saved in sets. In real-time mode, each saved set is sequentially fed as command signals into the control system so that the end-effector can track the path. Joint actuators, optical sensors and a PC (as a controller) form the control system. More details about the control system and interfacing can be found in [10]. The overall

path planning approaches are illustrated as a flow chart and diagram in Fig (8). Two different sets of initial and goal configurations are used as inputs to the path planner. In set one, calculation of virtual goal point is needed and in set two it is not. The resulted manipulator actions can be seen in Fig (9).

7- Conclusions:

This paper presents a path planning for cylindrical manipulator of 4-DOF based on artificial potential field. Instead of using functions for safe path searching, vectors have been used with distance computations for collision detections. The suggested vector based potential field does not need an optimization search method for searching of a goal point which can easily fall into local minima. It is possible to build X_{free} through the suggested \vec{O} with non parallel base to the x_0-y_0 plane (arbitrary) by introducing the Euler rotation matrix with respect to frame (0). The proposed collision detections (with little improving) can be used for other obstacle shapes. If the orientation of the end-effector is known (desired), it will be possible to use the proposed path planner with IK or any other improved solutions to plan a path for robotic arms with more DOF. For not very complex mediums of obstacle shapes and numbers, the vector based potential field can be used successfully for path planning.

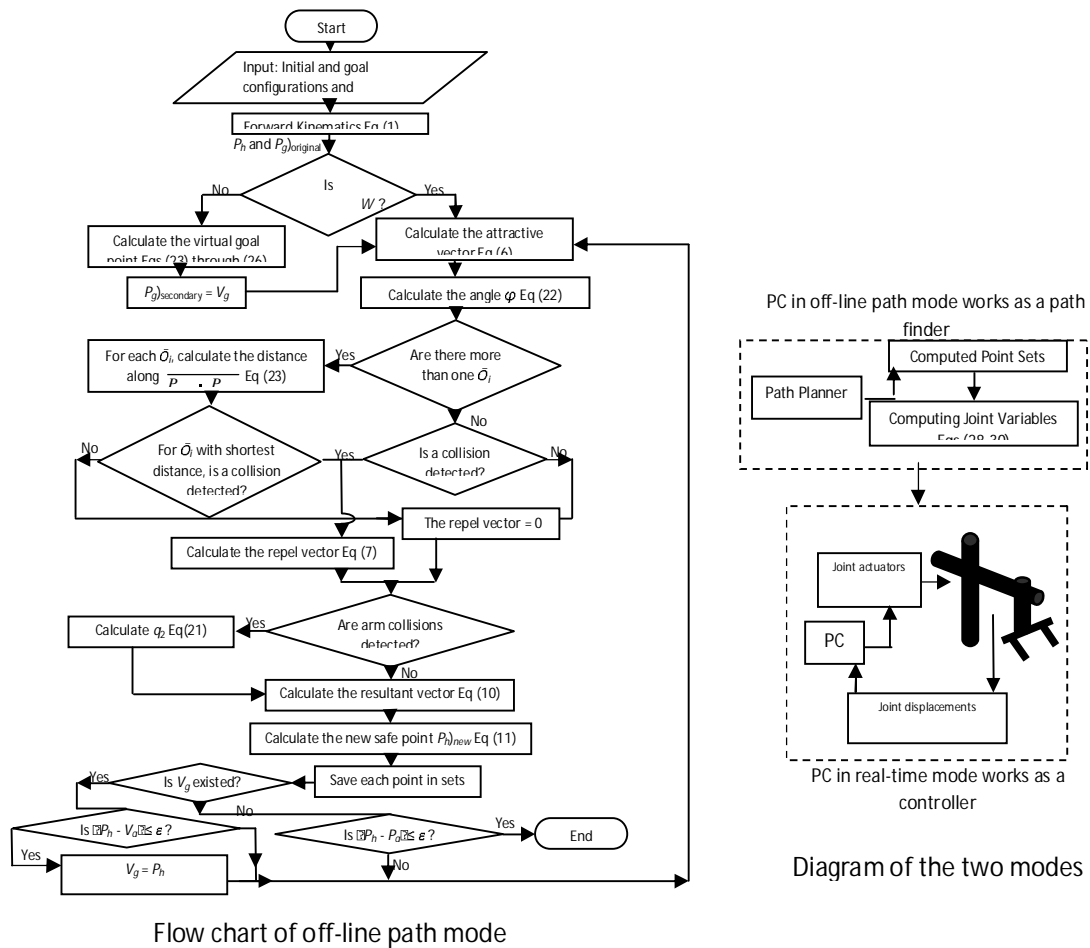


Fig (8) The Two modes of path finding and tracking



Path planning
among six
obstacles
without V_g
calculation

Manipulator
navigation with
 V_g calculation

Fig (9) Results of experimental works

References:

- [1] Pang C. Chen and Yong K. Hwang “SANDROS: A Dynamic Graph Search Algorithm for Motion Planning” IEEE Transaction on Robot and Automation, Vol. 14, No. 3 June 1998, 390-403.
- [2] Planas R.M., Fuertes J.M. and Martinez A.B. “Qualitative Approach for Mobile Robot Path Planning Based on Potential Field Method” Automatic Control Dept. Technical University of Catalonia, Spain, 2002.
- [3] Christopher I. Connolly and Roderic A. Grupen “On the Applications of Harmonic Functions to Robotics” Journal of Robotic System, Vol. 10, No. 7, 1993, 931-946.
- [4] Carlos Vazquez and Jan Rosell “Use of Path Planning Techniques Based on Harmonic Functions for the Haptic Guidance of Teleoperated Assembly Tasks” Mechatronics and robotics Dept. University of Politecnica, Germany, 2007.
- [5] Prahlad Vadakkepat, and Wang Ming-Liang “Evolutionary Artificial Potential Fields and Their Applications in Real Time Robot Path Planning” Dept. of Electrical Eng. National University of Singapore, Singapore, 1999.
- [6] Yunfeng W. and Gregory S. Chirikjian “A new Potential Field Method for Robot Path Planning” International Conference on Robotics & Automation, San Francisco, USA, April, 2000, 977-982.
- [7] Akiyoshi Sahara and Yuichiro Anzai “CAHRA: Collision Avoidance System for Humanoid Robot Arms with Potential Field” Faculty of Science and Technology, Kieo University, Japan, 2004.
- [8] Shingo S. and Karl I. “Potential Field Navigation of High Speed Unmanned Ground Vehicles on Uneven Terrain” International Conference on Robotics & Automation, Barcelona, Spain, April, 2005, 2839-2844.
- [9] Morten S. “Robot Path Planning: an Object-Oriented Approach” PhD thesis, Dept. Of Signals, Sensors and Systems, Royal Institute of Technology, Sweden 2004.
- [10] Rawand E. J. “Displacement Detection Application of HDNS A2051 Optical Sensor for Controlling a 4-DOF Cylindrical Manipulator” will be published in Journal of Al-Rafidain Eng. College of Eng. Mosul University, 2008.
- [11] Kishna C. “Mechanics and Control of Robots” Springer, USA, 1997, 171p.