IEEE Sponsored Conference on Computational Intelligence, Control and Computer Vision in Robotics & Automation

A Novel Method for Representing Robot Kinematics using Quaternion Theory

S. Sahu1, B.B.Biswal1 and Bidyadhar Subudhi

A Novel Method for Representing Robot Kinematics using Quaternion Theory

S. Sahu[1], B.B.Biswal[1] and Bidyadhar Subudhi[2]

Abstract-The traditional methods for representing forward kinematics of manipulators have been the homogeneous matrix in line with the D-H algorithm. In this paper a new method known as quaternion algebra is described and it is used for three dimensional representation and orientation in robot kinematics. This method is compared with homogeneous transform in terms of easiness of representation, computational efficiency and storage requirement. Conclusion drawn was that quaternion is more compact and efficient method of representation than the matrix form.

***Keywords*- forward kinematics, homogeneous matrix, spatial motion, quaternion algebra**

I. INTRODUCTION

THE problem of finding mathematical tools to represent rigid body motions in space has long been on the agenda of physicists and mathematicians and is considered to be a well-researched and well-understood problem. The first and most common method in the robotics community is based on homogeneous matrix transformation. In robotics, this matrix is used to describe one coordinate system with respect to another one. This has been the basis of tracking the position of the end-of-arm tool since ages. Robotics, computer vision, graphics, and other engineering disciplines require concise and efficient means of representing and applying generalized coordinate transformations in three dimensions. So a number of different representations have been developed. However for the purpose of on-line control and manipulation of devices, it is important to have alternative method. Such method should be compact and computationally efficient for representations of spatial transformations.

Michael W. Walker [1] present the position of a manipulator expressed as either in joint coordinates or in cartesian coordinates. Funda, Taylor and Paul [2] implement three-dimensional modeling of rotations and translations by using an alternate approach, employing quaternion/vector pairs as spatial operators, and compare with homogeneous transforms in terms of computational efficiency and storage economy. It is found that quaternion/vector pairs are efficient, more compact, and more elegant than their matrix counterparts. However for online operation and manipulation of the robotic manipulator in a flexible manner the computational time plays an important role. Again the appeal of homogeneous transforms is mostly due to their matrix formulation, which is familiar and lends itself to easy manipulation by a computer.

[1]S. Sahu, Dept. of Mechanical Engg. (santosini.sahu@gmail.com),
[1]B.B.Biswal , Dept. of Mechanical Engg. (bbbiswal@nitrkl.ac.in),
[2]Bidyadhar Subudhi, Dept. of Electrical Engg.
(bidyadhar@nitrkl.ac.in), NIT, Rourkela-769008, India [2],

On the other hand, such matrices are highly redundant to represent six independent degrees of freedom. This redundancy can introduce numerical problems in calculations, wastes storage, and often increases the computational cost of algorithms.

In parallel implementations, the extra data required to fetch the operands can also be a significant factor. Despite these drawbacks in mind, alternative methods are being sought by various researchers for representing the same and reducing the computational time to make the system fast responsive in a flexible environment. Researchers in robot kinematics tried alternative methods in order to represent rigid body transformations based on concepts introduced by mathematicians and physicists such as Euler angle, Epsilon algebra, dual quaternion algebra, lie algebra. Nicholas and Dimitros [3] present three methods for the formulation of the kinematics equations of robots with rigid links, the second one is based on Lie algebra, and the third one on screw theory expressed via dual quaternion algebra. These three methods are compared for their use in the kinematics analysis of robot arms by using analytic algorithms and are presented for the solution of the direct kinematics problem. However the application has not been done in higher DOF manipulators. Funda and Paul [4] propose a computational analysis and comparison of line-oriented representations of general i.e. rotational and translational spatial displacements of rigid bodies. Aissaoui, Mecheri and Hagemeister [5] present a study to investigate the accuracy of a new algorithm based on dual quaternion algebra for the estimation of the finite screw axis. Although quaternion has been used extensively in kinematic analysis they have been relatively neglected in practical robotic systems due to some complications in dealing with the problem of orientation representation. Perrier, Dauchez and Pierrot [6] suggest a mobile manipulator, composed of a manipulator mounted on a vehicle, is a very useful system to achieve tasks in dangerous environments. They find applications in many areas of geometric analysis and modeling. Although most of the work has been done on the issue of computational efficiency of effecting three-dimensional rotations and their compositions using quaternion, none of them addresses parallel implementations of the corresponding algorithms in robot kinematics.

The homogeneous matrix method has been used widely by the robotics community, and analytical algorithms have been written for the solution of the kinematic problem of robot arms. The popular method of homogeneous transformation matrix has so far overshadowed any other method. It is also true that quaternion algebra provides a strong base for the similar purpose and also it is useful in representation of rotation and translation of points and members in spatial plane. However its use for studying the kinematic behavior of rigid manipulator has not been made. The use of this approach can be made advantageously for kinematic representations of mechanisms largely because of two reasons: computationally the quaternion algebra is quite faster as it involves less number of parameters compared to the homogeneous representation and secondly, less mathematical complicacies involved in understanding the representations. The present work aims at developing a novel representation of the manipulator kinematics using quaternion algebra and comparing the same with the homogeneous transformation method for its effectiveness in terms space and time. An example problem is taken up to illustrate the practical implementation of this new method. The geometrical significance of the transformation operators and parameters are studied in order to show the physical meaning of them apart from the algebraic analysis. Finally a comparison in terms of computational time has been made with the homogeneous matrix method that shows the superiority of the new method over the traditional method.

II. METHODS OF REPRESENTATION

The first method based on homogeneous transformation is formulated by using D-H algorithm which depends upon already derived transformation operators such as matrices or vector. The second method in general is a line transformation method. In this method step by step calculation of the transformation operator is done along with line vectors. Since this is a open chain of rigid links connected successively by revolute or prismatic joints, the kinematic parameter definition is based on Denavit-Hartenberg notation.

A. Homogeneous Matrix Method

The mechanical manipulator can be modeled as several rigid bodies or links connected in series by either revolute or prismatic joints. In most robotic application; spatial description of the end-effector of the manipulator with respect to a fixed reference coordinate system is required. The total spatial displacement of the end effectors is due to the angular rotation and linear translations of the links. Homogeneous matrix method is the classical method to describe the relationship between two adjacent rigid mechanical links. To use homogeneous matrix method for displacement analysis of a spatial linkage we need to attach a coordinate frame to each link. These coordinate systems are established in a systematic manner following Denavit-Hartenberg's algorithm. Basic rotation matrices can be multiplied together to represent a sequence of finite rotations about the principal axes.

The basic rotation around x-axis is represented by

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} \quad (1)$$

Where, θ = angle of rotation

R_x =rotation matrix about ox axis.

The resultant rotation matrix is given by multiplying the three basic rotation matrices.

$$R = R_{y,\phi} R_{Z,\theta} R_{x,\alpha} \quad (2)$$

where

$R_{y,\phi}$ = rotation about y-axis with an angle ϕ

$R_{z,\theta}$ = rotation about z-axis with an angle θ

$R_{x,\alpha}$ = rotation about x-axis with an angle α

This can be written in short as;

$$\mathrm{R} = \begin{pmatrix} c\phi c\theta & s\phi s\alpha - c\phi s\theta c\alpha & c\phi s\theta c\alpha + s\phi s\alpha \\ s\theta & c\theta c\alpha & -c\theta s\alpha \\ -s\phi c\theta & s\phi s\theta c\alpha & c\phi c\alpha - s\phi s\theta s\alpha \end{pmatrix} \quad (3)$$

The matrix representation for rotation of a rigid body simplifies many operations but it needs nine elements to completely specify the orientation of a rotating rigid body it does not lead directly to a complete set of generalized coordinates, such a set of generalized coordinate can describe the orientation of a rotating rigid body with respect to a reference coordinate frame. The 3X3 matrix does not give any provision for translation and scaling, so a fourth coordinate or component is introduced to a position vector p where,

$$p = (p_x, p_y, p_z) \quad (4)$$

The homogeneous transformation matrix is a 4X4 matrix which maps a position vector expressed in homogeneous coordinates from one coordinate system to another coordinate system. The basic homogeneous matrix is represented by

$$^{t-1}A_t = \begin{pmatrix} \cos\theta_t & -\cos\alpha_t \sin\theta_t & \sin\alpha_t \sin\theta_t & a_t \cos\theta_t \\ \sin\theta_t & \cos\alpha_t & -\sin\alpha_t \cos\theta_t & \sin\theta_t \\ 0 & \sin\alpha_t & \cos\alpha_t & d_t \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5)$$

In a kinematic chain the transformation matrix $^{t-1}A_t$ describes the local coordinate frame for t^{th} link of the manipulator with respect to the local frame of the previous link $t-1$.The forward kinematic equation of the manipulator can be developed by multiplying the above matrix $^{t-1}A_t$ calculated sequentially for each link.

Using the $^{t-1}A_t$ matrix one can relate a point p_t at rest in link t and expressed in homogeneous coordinates with respect to the coordinate system $t-1$ established at link $t-1$ by $p_{t-1} = {}^{t-1}A_t\, p_t$. Alternatively the orientation matrix can be represented through Euler angles. There are different sequences of Euler angle representation. Here the sequence of Euler angle followed is

$$R_{\phi,\theta,\psi} = R_{z,\phi} R_{y,\theta} R_{x,\psi}$$

$$= \begin{pmatrix} c\phi c\theta & c\phi s\theta s\psi - s\phi c\psi & c\phi s\theta c\psi + s\phi s\psi \\ s\phi c\theta & s\phi s\theta s\psi + c\phi c\psi & s\phi s\theta c\psi - c\phi s\psi \\ -s\theta & c\theta s\psi & c\theta c\psi \end{pmatrix} \quad (6)$$

The orientation of a body in three dimensions is difficult to visualize and describe, so alternative methods have been tried. In the orientation matrix nine parameters are used to represent three degrees of freedom to specify the orientation of a body. In general the homogeneous method of representation is highly redundant since it requires 12 numbers to completely represent six degree of freedom.

The basic homogeneous matrix can be represented as

$$T = \begin{pmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (7)$$

where, n =normal vector of the hand
s =sliding vector of the hand
a =approach vector of the hand
p =position vector of the hand

An algorithm is presented for the derivation of kinematic equation of a n-link robot which is based on homogeneous transformation.

i) Assignment of a local coordinate system to every link and a global one to the base of the robot.

ii) Determination of the kinematic parameters for the links 1 to n.

iii) Determination of the transformation matrices ${}^{t-1}T_t$ for $t=1$ to n, describing each local coordinate system with respect to its previous one by

$${}^{t-1}T_t = \begin{bmatrix} c\theta_t & -s\theta_t & 0 & l_{t-1} \\ s\theta_t c\alpha_{t-1} & c\theta_t c\alpha_{t-1} & -s\alpha_{t-1} & -s\alpha_{t-1} d_t \\ s\theta_t s\alpha_{t-1} & c\theta_t s\alpha_{t-1} & c\alpha_{t-1} & c\alpha_{t-1} d_t \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

iv) Calculation of the final transformation matrix ${}^{0}T_n$ of the end effector coordinate system with respect to the base frame is done using following equation.

$${}^{0}T_n = {}^{0}T_1 \times {}^{1}T_2 \times {}^{2}T_3 \times {}^{n-1}T_n \,. \quad (9)$$

The position of the end effector is given by the last column of the matrix and orientation is given by upper left 3 X 3 sub matrix. The foundation of this algorithm is formulation of the combined homogeneous matrix. Once this matrix is obtained subsequent matrices are calculated sequentially for higher order links. The final matrices are formulated by multiplying the transformation matrices representing simple rotation about the principal axis and translations along the principal direction of the local coordinate system of the links. Homogeneous matrix method thus provides a systematic way to understand and implement the algorithm step by step.

B. Quaternion Algebra Method

Quaternion were introduced by Hamilton in late 1843. In this section quaternion algebra is presented and it is used to formulate the forward kinematic problem of the robot arm. Rotation quaternion can be used to calculate the rotated point from the original position of the point; this allows translation of points without using matrices. Since then they have found application in many areas of geometric analysis and modeling. Recently general properties of quaternion like special type of rotations, formulations of reflection and translation are discussed a lot by many authors.

Quaternion plays a vital role in the representation of rotations in computer graphics, primarily for animation. However, quaternion rotation is often left as an advanced topic in computer graphics education due to difficulties in portraying the four dimensional space of the quaternion. Interpolating the quaternion representation of a sequence of rotations is more natural than doing so for the familiar Euler angles, such as yaw, pitch, and roll. The quaternion occupies a smooth, seamless, isotropic space which is a generalization of the surface of a sphere.

Before going into detail analysis of quaternion and steps of formulation of kinematic equation for robot arm some properties of quaternion algebra is presented. Quaternion can be represented as

$$q = w + ix + jy + kz \quad (10)$$

Here w is the real part and x, y, z are imaginary parts. Each of these imaginary dimensions has a unit value of square root -1, all are mutually perpendicular to each other known as i, j, k.

i) Conjugate of quaternion:

The conjugate of a quaternion number is a quaternion with the same magnitudes but with the sign of the imaginary parts changed. So conjugate of $q = w + ix + jy + kz$ is

$$q' = w - ix - jy - kz \,. \quad (11)$$

ii) Magnitude:

The magnitude of a quaternion $q = w + ix + jy + kz$ is

$$\sqrt{(w^2 + x^2 + y^2 + z^2)} \quad (12)$$

iii) Norm:

Norm of a quaternion is defined by

$\| q \|$ = square root of (q *conj(q))

$$= \sqrt{(w^2 + x^2 + y^2 + z^2)} \quad (13)$$

iv) Inverse:

The inverse of a quaternion refers to the multiplicative inverse $\frac{1}{q}$ and can be computed by $q^{-1} = q'/(q * q')$ (14)

If a quaternion q has length 1, we say that q is a unit quaternion. The inverse of a unit quaternion is its conjugate, i.e., $q^{-1} = q' \ q^{-1}$. One of the major properties of quaternion is that they are anti commutative. Quaternion algebra can be understood as an extension of complex number. As we know complex number consist of one real part and one imaginary part similarly quaternion has four dimensions i.e. one real part and three imaginary part.

A dual number can be defined as

$$q + \varepsilon q^0 \quad (15)$$

where,

q and q^0 are real numbers with ε as a dual unit having property $\varepsilon^2 = 1$.

Representation using quaternion

Quaternion can be used to represent rotation and quaternion multiplication can be used to get the result of subsequent rotation.

(a)Representation of rotation.

Let q_1 and q_2 are unit quaternion representing two rotations. Then subsequent rotation can be done by, rotating first q_1 and then q_2. The composite rotation is represented by the quaternion $q_2 * q_1$.

$$\begin{aligned} P_2 &= q_2 * (q_1 * P * q_1^{-1}) * q_2^{-1} \\ &= (q_2 * q_1) * P * (q_1^{-1} * q_2^{-1}) \\ &= (q_2 * q_1) * P * (q_2 * q_1)^{-1} \end{aligned} \quad (16)$$

where,

p_1= vector representing the initial position of a point being transformed.

p_2 =vector representing the final position of the point after translation. The quaternion can represent 3D reflections, rotations and scaling, however a single quaternion operation cannot include translations with rotation. So for rotation, reflection or scaling around a point other than the origin, we would have to handle the translation part separately.

(b) Representation of pure translation

The translation in quaternion algebra is done by using a quaternion operator and it is defined by

$$q = 1 + (x_1/2)i\varepsilon + (y_1/2)j\varepsilon + (z_1/2)k\varepsilon \quad (17)$$

where,

x_1, y_1, z_1 are the translation carried out along x,y,z direction respectively.

Quaternion transformation is represented as

$$p_2 = q * p_1 * q' \quad (18)$$

where,

p_1 and p_2 are initial and final position of a point

q =dual quaternion operator representing transform.

q' =conjugate of q.

Quaternion follows certain rules while performing multiplication.

Here, $i^2 = j^2 = k^2 = -1$.

$$i * j = k$$
$$j * k = i$$
$$k * i = j$$

Multiplication of quaternion numbers together behaves similarly to cross product of the unit basis vectors. Unit line vector is a vector which is constrained to lie on a definite line in Fig.1, u is a unit vector and $u^0 = r \times u$ is the moment vector, where r is the position vector of an arbitrary point P on the line. The vectors u, u_0 are often called plucker vectors.

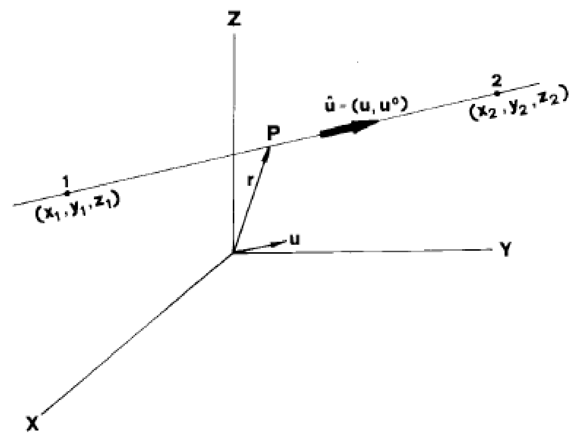

Fig. 1. A unit line vector

A unit line vector in its dual form can be represented as;

$$\hat{u} = u + \varepsilon u^0 . \quad (19)$$

From the geometrical point of view a quaternion can also be represented as;

$$q = \cos\frac{\theta}{2} + u\sin\frac{\theta}{2} \quad (20)$$

where, $\cos\frac{\theta}{2}$, the first part of the equation is the real part and the second part $\sin\frac{\theta}{2}$ is the complex part. Quaternion can be thought of as an axis angle u along which rotation is consided. It is quite difficult to give physical meaning to a quaternion but it forms an interesting mathematical system. A quaternion can also be written as

$$q = q_0 + q_x i + q_y j + q_z k \quad (21)$$

which is a combination of a scalar q_0, and $q_x i + q_y j + q_z k$ represents the vector component along three mutually perpendicular directions. A dual quaternion may be written as

$$Q = (q_0 + \varepsilon q_0^0)1 + (q_x + \varepsilon q_x^0)i + (q_y + \varepsilon q_y^0)j + (q_z + \varepsilon q_z^0)k \quad (22)$$

The combined rotation and translation can be represented by quaternion operator. The relationship between two non parallel and non intersecting unit line vector can be obtained as follows.

Let,

$\hat{u}$ = A unit line vector.

$\hat{v}$ = Unit line vector obtain by translation of $\hat{u}$ by a distance d, followed by rotation of a dual angle $\varphi = \varphi + \varepsilon d$.

The transformation of $\hat{u}$ into $\hat{v}$ is given by

$$\hat{v} = \hat{Q}\hat{u} \tag{23}$$

The transformation operator is defined as

$$\hat{Q} = \cos\hat{\varphi} + \hat{e}\sin\hat{\varphi} \tag{24}$$

which is also a dual quaternion. So by multiplying a unit line vector by the transformation operator $\hat{Q}$ the image of that line is obtained in a new location defined by the parameters of this transformation operator.

(c) Application of quaternion in formulation of kinematic equation of 3R robot arm

A dual quaternion is the set of four dual numbers in a definite manner. Just like we extend 3 X 3 matrices to 4 X 4 matrices to allow them to translation in addition to rotation we extend quaternion to dual quaternion to allow them to represent translations in addition to rotation. The dual of a quaternion can model the movement of a solid object in 3 dimensions which can rotate and translate without changing the shape.

Algorithm to formulate forward kinematic equation:

i) Assignment of coordinate system to every link and to the base.

ii) Determination of the kinematic parameters for the links 1 to n.

iii) Calculation of unit line vector which is coincident with the common normal between t^{th} and $(t+1)^{\text{th}}$ axis and s_t as the unit line vector along z axis of the t^{th} joint.

$$a_{t,t+1} = \hat{Q}_t \hat{a}_{t-1,t} \tag{25}$$

$$\text{and } \hat{s}_{t+1} = \hat{Q}_{t,t+1}\hat{s}_t \tag{26}$$

for base frame unit vectors

$$a_{0,1} = i \text{ and } s_1 = k\,. \tag{27}$$

The transformation operators are

$$\hat{Q}_t = \cos\hat{\theta}_t + s_t \sin\hat{\theta}_t \text{ and} \tag{28}$$

$$\hat{Q}_{t,t+1} = \cos\hat{\alpha}_{t,t+1} + \hat{a}_{t,t+1}\sin\hat{\alpha}_{t,t+1} \tag{29}$$

where $\hat{\alpha}_{t,t+1}$ the dual angle between $\hat{s}_t$ and $\hat{s}_{t+1}$ is defined as

$$\hat{\alpha}_{t,t+1} = \hat{\alpha}_t + \varepsilon \mathrm{L}_t \tag{30}$$

and $\hat{\theta}_t$, the dual angle between $\hat{a}_{t-1,t}$ and $\hat{a}_{t,t+1}$ is defined as

$$\hat{\theta}_t = \theta_t + \varepsilon \mathrm{d}_t \tag{31}$$

in terms of four D-H kinematic parameters.

iv) The position vector of the end effector is given by

$$P_n = \sum_{t=1}^{n}(d_t s_t + L_t a_{t,t+1})\,. \tag{32}$$

v) The orientation matrix R of the end effector coordinate system by the three vectors

$$n_n = a_{n,n+1}$$
$$o_n = s_{n+1} \times a_{n,n+1}$$
$$a_n = s_{n+1}\,. \tag{33}$$

Following the steps of the given algorithm the kinematic equation of any spatial manipulator can be evaluated. The unit line vector s_t defines the axis of the joint t and $\hat{a}_{t,t+1}$ defines the common perpendicular to the axes of the t and $t+1$ joints. The dual quaternion transforms the unit line vector $\hat{a}_{t-1,t}$ to $\hat{a}_{t,t+1}$. In other words, the operator translates the x-axis of the frame $t-1$ along the axis of the joint t by an angle θ_t .the dual quaternion $\hat{Q}_{t,t+1}$ is similar. It translates and rotates the joint axis along and about the common perpendicular to this joint axis and the next one.

The vectors s and a are the unit vectors defining the orientation of the z and x axis respectively. The unit vector of the Y axis of the last coordinate system is calculated by the vector product of the s and a vectors.

III. APPLICATION OF THE ALGORITHM IN A 3R ROBOT

The algorithms explained for quaternion algebra is implemented to a 3R robot. The position and orientations are calculated by applying both homogeneous and quaternion algebra method. Let us consider a 3R robot having joint parameters as given in Table 1.

TABLE 1

KINEMATIC PARAMETERS OF 3R ROBOT

Kinematic parameters of 3R robot				
t	θ_t	$\alpha_{t,t+1}$	$L_{t,t-1}$	d_t
1	90	-90	0	0
2	0	0	431.8 mm	149.09 mm
3	90	90	-20.32	0

In quaternion algebra method by using equation (24) to (32), the calculation can be made as follows.

For base coordinate,

$t = 0$

$a_{01} = i$

$s_t = k$

This result has been obtained by using equation (32).

For joint $t = 1$

$Q_1 = C_1 + S_1 k$

$a_{1,2} = C_1 i + S_1 j$

$Q_{1,2} = -a_{1,2}$

$s_2 = -S_1 i + C_1 j$

$P_1 = d_1 s_1 + L_1 a_{1,2} = 0$

$n_1 = a_{1,2} = j$

$o_1 = s_2 \times a_{1,2} = -k$

$a_1 = s_2 = -i$

For joint t =2,

Position of the end-effector is calculated by using equation (33).

$P_2 = -149.09i + 431.8j$

$n_2 = a_{2,3} = j$

$o_2 = s_3 a_{2,3} = -k$

$a_2 = s_3 = -i$

For joint t=3,

The position of the end effecter is calculated as

$P_3 = -149.09i + 431.8j + 20.32k$

The orientation of the end effecter is given by

Applying homogeneous matrix transformation matrix

$T = {}^0T_1 \times {}^1T_2 \times {}^2T_3$

$$= \begin{pmatrix} 0 & -1 & 0 & -149.09 \\ 0 & 0 & 1 & 431.8 \\ -1 & 0 & 0 & 20.32 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Here we can observe that the position of the end effector is represented as last column of the homogeneous matrix is same with that obtained in quaternion algebra method.

IV. DISCUSSION

The observation made from the models and the subsequent solution prompt a comparision of the two methods. In the homogeneous transformation method, four trigonometric function calls and six multiplications are required for calculation of the transformation matrix ${}^{t-1}T_t$. The operation used in this algorithm is the product of 4 x 4 transformation matrices. The multiplication of two 4 x 4 transformation matrices needs 48 multiplications and 36 additions and subtraction, since the elements of the last row of the matrix are constants. In n-link robot arm the number of transformation matrices is n, so n-1 number of matrix products is required in order to detrmine the toal transfrmation matrix. Hence, the determination of the end effector position and orientation needs (48(n-1) + 6n) multiplications and 36(n-1) additions and subtractions, while only the orientation needs 31(n-1) multiplications and 18(n-1) additions and subtractions. For the case of 3-R robot (n=3), 114 multiplications and 72 additions and subtractions are necessary. This speaks about the number of mathematical operations required for computing the homogeneous matrix and hence the time needed for the same.

In dual quaternion method only the primary parts of the quaternion are required for the computation of the position and orientation of the end-effector. From 3rd step mentioned in section II, of the algorithm, it can be seen that all the necessary unit vectors are determined successively in a loop with t = 1 to n. In every step of this loop two main operations are performed. The first one is the determination of the transformation quaternion $\hat{Q}_t$ and $\hat{Q}_{t+1}$ using equation (28) and equation (29) respectively. For determination of each of these 3 multiplications are needed. The second operation is the quaternion product used to determine the unit line vectors $a_{t,t+1}$ and s_t from equation (28) and equation (29). Here, the vectors a and s are considered as quaternions with zero scalar part. The quaternion product needs 8 additions and subtractions and 12 multiplications. After the ending of this loop, the position vector of the end-effector is determined by adding the n position vectors of every joint in the open kinematic chain of the robot as represented in equation (32). To determine the position vector of every joint and to add it to the previous one, 6 additions and subtractions and 6 multiplications are needed. The first and last column of the orientation matrix of the end-effector is known from the unit vectors $a_{t,t+1}$ and s_{t+1} . The determination of the second column of the orientation matrix, which is the cross product of the other two columns, needs 3 additions and subtractions and 6 multiplications. In summary, (22n + 3) additions and subtractions and (36n + 6) multiplications are required in order to determine the position and orientation of the end-effector by the quaternion method. For the case of 3-R manipulator 114 multiplications and 69 additions and subtractions are necessary.

It is supposed that the computational time to perform an addition is half of the time required for one multiplication. A comparison of the number of mathematical operations required for computation of end-effector position using the two methods is presented in Fig.2.It is clear from this comparison that the number of operations is almost same with manipulators having less DOF.

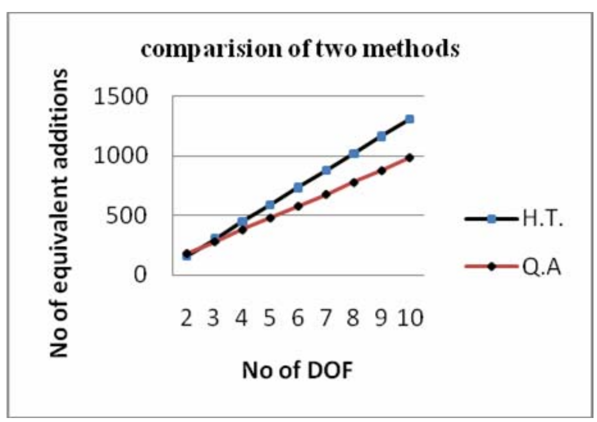

Fig. 2. Mathematical operations required with increasing DOF

As the number of DOF goes on increasing and the complicacies of computations increase the quaternion method scores significantly better than the homogeneous matrix method. It is obvious that for manipulators with more than three degrees of freedom, the quaternion theory based algorithm requires less computational time than the traditional homogeneous algorithms. Therefore, it can be concluded that for manipulators with high number of DOF the dual quaternion theory method is more cost effective than the homogeneous transformation. Further, in dual quaternion method the storage cost is minimum because it is not necessary to store all the transformation quaternions from the beginning. Quaternions require eight memory location for the representation of position while three memory location for orientation.but the homogeneous method requires 16 memory location for both position and orientation. The storage requirement affects the computational time as the cost of fetching an operand from memory exceeds the cost of performing a basic arithmatic operation.

V. CONCLUSION

It is evident from the results that a matrix product requires many more operations than a quaternion product. So a lot of time can be saved and at the same time more numerical accuracy can be preserved with quaternion than with matrices. In the example mentioned in the present work, it is clear that quaternion algebra provides a very effective and efficient method for representation of forward kinematics equation. Further, the method is cost effective as it requires less computer memory and saves lot of time by reducing the number of mathematical calculation. Comparing the two methods, it is observed that the quaternion method gives exactly same result as that of homogeneous method. This is a general method applied specifically to robot manipulator in the present work. However this can also be extended to any other open kinematic chain for the purpose of kinematic analysis. Therefore this can be used as a powerful tool in the solution of kinematic problems.

This paper introduces a new formulation for the kinematic synthesis of open link robots having three joints. The standard kinematic equations of the chain are transformed into successive quaternion transformation and then expressed using dual quaternion. Presently there are many quaternion applications in the area of aerospace sequence, spherical trigonometry, calculus for kinematics and dynamics, rotation in phase space etc. A lot more research has to be done in this aspect and the days are not far behind when quaternion will replace the traditional homogeneous method of representation.

VI. REFERENCES

[1] M. W. Walker, "Manipulator kinematics and the epsilon algebra," *IEEE,J. Robot. Automat.*, vol. 4, Apr. 1988.

[2] J. Funda and R. P. Paul, "A computational analysis of screw transformations in robotics," *IEEE Trans. Robot. Automat.* vol. 6, pp. 348–356, June 1990.

[3] Nicholas A. Aspragathos and John K Dimitros, "A Comparative Study of Three Methods for Robot Kinematics", *transactions on systems, man, and cybernetics—part b: cybernetics*, vol. 28, no. 2, april 1998.

[4] J. Funda, R. H. Taylor, and R. P. Paul, "On homogeneous transforms, quaternions, and computational efficiency," *IEEE Trans. Robot. Automat.* vol. 6, pp. 382–388, June 1990.

[5] Aissaoui R., Mecheri H., Hagemeister N., de Guise J.A. "Robust estimation of screw axis from 3D pose using dual quaternion algebra".1996.

[6] C. Perrier, P. Dauchez, F. Pierrot," Nonholonomic motion planning for mobile manipulators**",** *IEEE*, 1997.

[7] J. J. Craig, *IntroductiontoRobotics, Mechanics and Control*. Reading, MA: AddisonWesley, 1986.

[8] O. P. Agrawal, "Hamilton operators and dual-number-quaternions inspatial kinematics," *Mech. Mach. Theory*, vol. 22, no. 6, pp. 569–575,1987.