Jelena Z. Vidakovic
Research assistant
University of Belgrade
Lola Institute

Mihailo P. Lazarevic
Full Professor
University of Belgrade
Faculty of Mechanical Engineering

Vladimir M. Kvrgic
Research associate
University of Belgrade
Lola Institute

Zorana Z. Dancuo
Research assistant
University of Belgrade
Lola Institute

Goran Z. Ferenc
Research assistant
University of Belgrade
Lola Institute

Advanced Quaternion Forward Kinematics Algorithm Including Overview of Different Methods for Robot Kinematics

Formulation of proper and efficient algorithms for robot kinematics is essential for the analysis and design of serial manipulators. Kinematic modeling of manipulators is most often performed in Cartesian space. However, due to disadvantages of most widely used mathematical constructs for description of orientation such as Euler angles and rotational matrices, a need for unambiguous, compact, singularity free, computationally efficient method for representing rotational information is imposed. As a solution, unit quaternions are proposed and kinematic modeling in dual quaternion space arose. In this paper, an overview of spatial descriptions and transformations that can be applied together within these spaces in order to solve kinematic problems is presented. Special emphasis is on a different mathematical formalisms used to represent attitude of a rigid body such as rotation matrix, Euler angles, axis-angle representation, unit quaternions, and their mutual relation. Benefits of kinematic modeling in quaternion space are presented. New direct kinematics algorithm in dual quaternion space pertaining to a particular manipulator is given. These constructs and algorithms are demonstrated on the human centrifuge as 3 DoF robot manipulator.

Keywords: *robot, orientation, direct kinematics, quaternion, dual quaternion.*

1. INTRODUCTION

Robot kinematics pertains to the motion of bodies in a robotic mechanism without regard to the forces/torques that cause the motion. Since robotic mechanisms are by their very essence designed for motion, kinematics is the most fundamental aspect of robot design, analysis, control and simulation [1]. Special emphasis in the study of robot kinematics is on geometry of the structure of robotic system which is modelled as multi-degree of freedom kinematic chain. A kinematic chain is an assemblage of rigid bodies-links, connected by joints providing pure rotation or translation [2], [3].

In general, a rigid displacement may consist of both translation and rotation of the object [4]. Thus, a rigid body in three-dimensional space generally has six degrees of freedom; its location is completely described by its position and orientation-the pose of a rigid body. It is of greatest importance in robotic systems to know the correct location of the end-effector in every moment. The actuated joint variables and their time derivatives can be obtained by reading of sensors installed on the joints. Direct kinematics problem is to find end-effector position and orientation relative to the base when positions of all joints are given [5]. During the development of kinematics, a number of mathematic theories, tools and algorithms have been introduced and implemented [4], [6]. In this paper, two different spaces, Cartesian space and quaternion space, are used for overview of mathematical constructs representing rigid transformation.

Orientation of the rigid body can be described in several different ways. A rotation is a displacement in which at least one point of the rigid body remains in its initial position and not all lines in the body remain parallel to their initial orientations [1]. Psyhically, a rotation may be explained as reorientation of a body without changing the body's size and shape [7]. All mathematical formalisms used to describe attitude of the rigid body are derived from *Euler's rotation theorem*.

According to Euler's rotation theorem, any rotational movement of a rigid body in three-dimensional space such that a point of the rigid body remains fixed troughout the movement is equivalent to a single rotation about particular axis passing through the fixed point. The axis of rotation is represented by a unit vector $\hat{\boldsymbol{u}}$ called Euler axis [8]. Hence, any rotation in three dimensional space can be represented via unit vector indicating the direction of a rotational axis and a scalar θ (angle of rotation). This rotation representation is called *axis-angle* representation (Figure 1). From axis-angle representation another representation-*rotation vector*, or *Euler vector* is derived. Rotation vector is represented by a vector directed towards the rotational axis whose magnitude is value of angle of rotation θ, $\boldsymbol{v} = \theta\hat{\boldsymbol{u}}$.

Received: November 2013, Accepted: Jnauary 2014
Correspondence to: Jelena Vidakovic
Lola Institute
Kneza Viseslava 70a, 11120 Belgrade 35, Serbia
E-mail:vidakovic.jelena@gmail.com
doi:10.5937/fmet1403189V

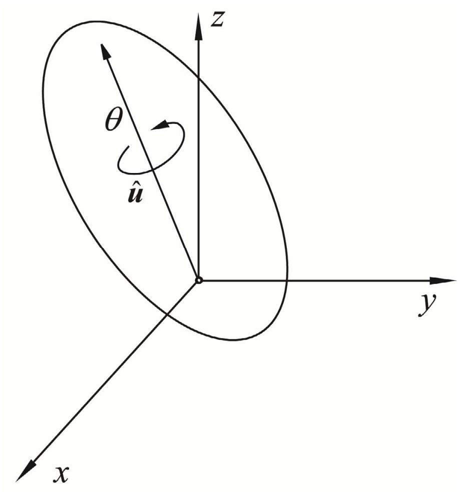

Figure 1. Axis-angle representation

Besides the axis-angle and rotational vector representations, several sets of three-parameter representations have been reported in the literature. These three-parameter representations are called *minimal representation*s. Perhaps, the most commonly used minimal representation is set of *Euler angles* [9].

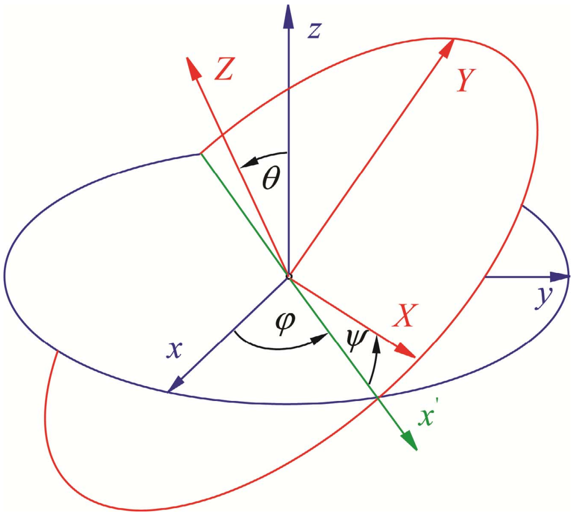

Figure 2. Proper Euler angles

In linear algebra, the Euler theorem states that in three-dimensional space, any two Cartesian coordinate systems with a common origin can be put to coincidence by a finite rotation about certain axis passing through the origin. This approach is used in mechanics of rigid body. Frame can be attached to the rigid body and moving together with it, it defines orientation of the body. This frame is called *local* and it is described by three basis vectors. Orientation of these basis vectors can be described by means of their direction cosines with respect to the basis of the fixed frame. These parameters can be composed into 3 × 3 matrix called a *rotation matrix* [1]. A rotation matrix may also be referred to as a *direction cosine matrix*, because the elements of this matrix are the cosines of the unsigned angles between the body-fixed axes and the axes of the fixed frame.

Combining mathematical constructs for position in the form of position vector in homogeneous coordinates and orientation represented by rotational matrix yields 4x4 *homogeneous transformation matrix*. This real orthonormal matrix is widely used in robotics community [10].

The most popular mathematical constructs describing attitude of a rigid body possess certain disadvantages. Euler angles suffer from wrist singularities which are unfortunate consequence of using minimal representation [11]. Rotational matrix can be numerically unstable [12]. These disadvantages are overcome by usage of *unit quaternions*. The relevant functions of the unit quaternions have no singularities and they are numerically stable end efficent. When used to integrate incremental changes in attitude over time, they are more accurate than Euler angles [9]. Counterpart for homogenous transformation matrix in quaternion space is *dual quaternion.* Dual quaternions inherit advantages of unit quaternions while representing rigid transforms [13].

Although quaternions constitute an elegant representation of rotation, they are not used as much as other constructs in robotics community [14]. Main reason for this is lack of physicality, i.e. physical meaning of four quaternion parameters is not apparent which is the case with, for an example, Euler angles. On the other hand, extracting the angle and axis of rotation is simple.

The purpose of this paper is to provide simple explanation of possibilities of usage of unit and dual quaternions for spatial descriptions and transformations in robotics. Their relation with corresponding constructs in Cartesian space is given. Features of unit and dual quaternions are considered as much as necessary to be used in robot kinematics relations and algorithms. New direct kinematics algorithm in dual quaternion space is presented and demonstrated on human centrifuge which is modelled as 3 DoF robot manipulator.

It is important to say that all mathematical formalisms described in this paper apply only to coordinate systems of determined orientation. Here, right handed frames are used.

2. SPATIAL DESCRIPTIONS AND TRANSFORMATIONS IN CARTESIAN SPACE

To identify the location of a body in Cartesian space, a reference coordinate system should be established. This reference coordinate system is called world coordinate system or fixed coordinate system and here it is denoted by $O_j x_j\ y_j\ z_j$. After defining a reference coordinate system, the position and orientation of the rigid body are fully described by the position of the origin of the local frame $O_i\, x_i y_i z_i$ and the orientation of its axes.

2.1 Position of the local frame origin

The position of the origin of coordinate frame i relative to coordinate frame j can be denoted by the 3 × 1 vector:

$$ {}^{j}\boldsymbol{p}_i = \left({}^{j}p_i^x \quad {}^{j}p_i^y \quad {}^{j}p_i^z \right)^T . \tag{1} $$

The components of this vector are the Cartesian coordinates of the position vector of the frame origin O_i with respect to reference j-th frame.

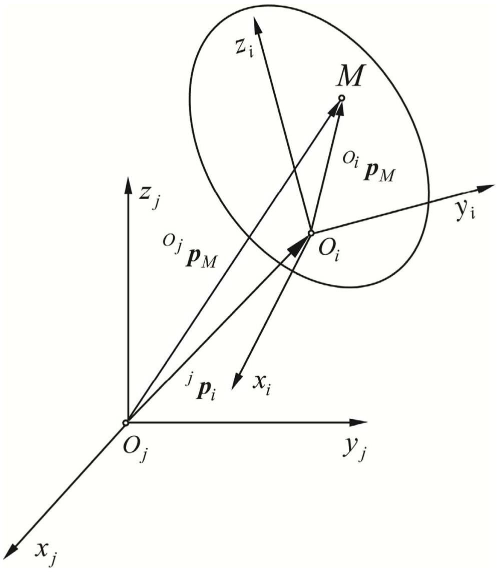

Figure 3. Fixed and local coordinate system

2.2 Attitude presentations: Rotational matrix

The orientation of coordinate frame *i* relative to coordinate frame *j* can be determined by defining the basis vectors of local frame $(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{y}}_i, \hat{\boldsymbol{z}}_i)$ with respect to the basis vectors of the fixed frame $(\hat{\boldsymbol{x}}_j, \hat{\boldsymbol{y}}_j, \hat{\boldsymbol{z}}_j)$. A rotation matrix ${}^{j}\boldsymbol{R}_i$ is a matrix whose multiplication with a vector rotates the vector while preserving its length [9]. The elements of ${}^{j}\boldsymbol{R}_i$ are dot products of basis vectors of the two coordinate frames:

$$ {}^{j}\boldsymbol{R}_i = \begin{bmatrix} cos(\hat{\boldsymbol{x}}_j, \hat{\boldsymbol{x}}_i) & cos(\hat{\boldsymbol{x}}_j, \hat{\boldsymbol{y}}_i) & cos(\hat{\boldsymbol{x}}_j, \hat{\boldsymbol{z}}_i) \\ cos(\hat{\boldsymbol{y}}_j, \hat{\boldsymbol{x}}_i) & cos(\hat{\boldsymbol{y}}_j, \hat{\boldsymbol{y}}_1) & cos(\hat{\boldsymbol{y}}_j, z_i) \\ cos(\hat{\boldsymbol{z}}_j, \hat{\boldsymbol{x}}_i) & cos(\hat{\boldsymbol{z}}_j, \hat{\boldsymbol{y}}_i) & cos(\hat{\boldsymbol{z}}_j, \hat{\boldsymbol{z}}_i) \end{bmatrix}. \quad (2) $$

The columns of this matrix are unit vectors of local frame with respect to the fixed frame:

$$ {}^{j}\boldsymbol{R}_i = \begin{bmatrix} {}^{j}\hat{\boldsymbol{x}}_i & {}^{j}\hat{\boldsymbol{y}}_i & {}^{j}\hat{\boldsymbol{z}}_i \end{bmatrix}. \quad (3) $$

If ${}^{i}\boldsymbol{p} \in \mathbb{R}^3$ and ${}^{j}\boldsymbol{p} \in \mathbb{R}^3$ are coordinate vectors representing positions of the same point in three-dimensional space with respect to the frames *i* and *j* and the origins of the two frames coincides, then following relation holds:

$$ {}^{j}\boldsymbol{p} = {}^{j}\boldsymbol{R}_i \, {}^{i}\boldsymbol{p}. \quad (4) $$

In this paper, notation is adopted in which if reference frame is fixed frame, superscript *j* is denoted by 0 or denoting superscript is simply omitted.

Rotation matrices are square matrices with real entries. More specifically, they can be characterized as orthogonal matrices $(\boldsymbol{R}^T = \boldsymbol{R}^{-1})$ with determinant 1. The set of all such matrices of size 3 forms a special orthogonal group SO(3).

The following three rotation matrices rotate vector in space about the *x*, *y*, or *z* axis respecitvely:

$$ \boldsymbol{R}_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\,\theta & -sin\,\theta \\ 0 & sin\,\theta & cos\,\theta \end{bmatrix}, \quad (5) $$

$$ \boldsymbol{R}_{y,\theta} = \begin{bmatrix} cos\,\theta & 0 & sin\,\theta \\ 0 & 1 & 0 \\ -sin\,\theta & 0 & cos\,\theta \end{bmatrix}, \quad (6) $$

$$ \boldsymbol{R}_{z,\theta} = \begin{bmatrix} cos\,\theta & -sin\,\theta & 0 \\ sin\,\theta & cos\,\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (7) $$

Other rotation matrices can be obtained from (5), (6) and (7) using successive matrix multiplication. The result of multiplication is another rotational matrix representing one resultant rotation which produces the same effect as sequential application of the original rotation matrices.

The rotation matrix ${}^{j}\boldsymbol{R}_i$ contains nine elements, while only three parameters are required to define the orientation of a body in space. Therefore, six auxiliary relationships exist between the elements of the matrix [1]. This redundancy can introduce numerical problems in calculations and often increase computational cost of an algorithm [12].

2.3 Attitude presentations: Euler angles

The most common way to represent the attitude of a rigid body in minimal representation is a set of three Euler angles φ, θ, and ψ, known respectively as spin, nutation, and precession. These are popular because they are easy to understand and easy to use [9].

In the Euler angle representation, three rotations in sequence about the coordinate axis of either a fixed or a moving coordinate frame can describe any rotation provided that two successive rotations are not performed about parallel axes. If these elemental rotations are performed about the axes of the fixed frame, they are called extrinsic rotations. Otherwise, if the frame about which axes rotations are performed is local frame, these elemental rotations are called intrinsic. These two sets can be classified in two groups, called *proper Euler angles* and *Tait–Bryan angles* also known as roll, pitch and yaw *(RPY angles)*. In the case of proper Euler angles (Figure 2), the first and third rotational axis are the same (e.g. *x-y-x*, or *z-y'-z"*). RPY angles represent rotations about three different axes (e.g. *x-y-z*, or *x-y'-z"*). Here, superscripts *'* and *"* are used in the case of intrinsic rotations to denote axes of moving frame after first and second rotation respectively.

Traditionally, rotational matrices have been used to represent Euler angles because the basic rotation matrices for rotation about the x, y, and z axes are simple and well-known [15]. Let us consider RPY convention in which elemental rotations are performed about *z, y'* and *x"* axis, with angles ψ , θ , and φ. The function that maps this transformation to its corresponding rotation matrix $\boldsymbol{R}_{\mathrm{xyz}}: \mathbb{R}^3 \rightarrow \mathrm{SO}(3)$ is:

$$\boldsymbol{R}_{\mathrm{xyz}} = \boldsymbol{Rot}\left(z,\psi\right)\boldsymbol{Rot}\left(y^{'},\theta\right)\boldsymbol{Rot}\left(x^{''},\varphi\right). \qquad (8)$$

Here $\boldsymbol{Rot}\left(z,\psi\right)\boldsymbol{Rot}\left(y^{'},\theta\right)\boldsymbol{Rot}\left(x^{''},\varphi\right)$, are given by (7), (6), (5) and following rotational matrix is obtained:

$$\boldsymbol{R}_{\mathrm{xyz}} = \begin{bmatrix} c\psi\, c\theta & c\psi\, s\theta\, s\varphi - s\psi\, c\varphi & c\psi\, s\theta\, c\varphi + s\psi\, s\varphi \\ s\psi\, c\theta & s\psi\, s\theta\, s\varphi + c\psi\, c\varphi & s\psi\, s\theta\, c\varphi - c\psi\, s\varphi \\ -s\theta & c\theta\, s\varphi & c\theta\, c\varphi \end{bmatrix}. \qquad (9)$$

Here and throughout the paper, abbreviations *c* and *s* are used to denote cosine and sine of an angle. If orientation of end-effector is given in the form of rotation matrix:

$$\boldsymbol{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. \qquad (10)$$

RPY angles are obtained from following equations:

$$\psi = a\,tan\left(r_{21},r_{11}\right), \qquad (11)$$

$$\theta = a\,tan2\left(-r_{31}, cos\,\psi r_{11} + sin\psi r_{21}\right), \qquad (12)$$

$$\varphi = a\,tan2\left(sin\psi r_{13} - cos\,\psi r_{23}, -sin\psi r_{12} + cos\,\psi r_{22}\right). \qquad (13)$$

The conversion from a general rotation to Euler angles is ambiguous since the same rotation can be obtained with different sets of Euler angles [15]. Another disadvantage of this method is singularities found in the various Euler angle representations which occur in problem of gimbal lock situation [9]. A gimbal is a physical device consisting of spherical concentric hoops with pivots connecting adjacent hoops, allowing them to rotate within each other [16]. A gimbal is constructed by aligning three rings and attaching them orthogonally. Gimbals are often seen in gyroscopes used by the aeronautical industry. Gimbal lock is loss of one degree of freedom which occurs when two rotational axis align (Figure 4). This phenomenon can be considered mathematically through rotation matrix representation of Euler angles (9). If angle θ is $\pi/2$, angles ψ and φ cannot be determined, only their difference can be determined.

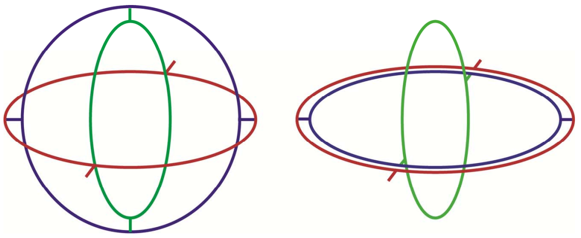

Figure 4. Gimbal with three degrees of freedom and gimbal lock situation in which one degree of freedom is lost

It is very important to avoid gimbal lock situation because the unpredictability could potentially cause fatal consequences [17].

In robotics, gimbal lock is commonly referred to as "wrist flip" or wrist singularity and it happens when two axes align, for example first and third or fourth and sixth axis. As joints rotate and approach to wrist flip, the singularity will cause numerical ill-conditioning. In order to maintain constant orientation, second (or fifth) axis makes a spin of 360° or 180°. This high velocity motion could cause damages and should be avoided.

Another problem using Euler angles is that they are less accurate than unit quaternions when used to integrate incremental changes in attitude over time [9]. This creates a problem in relating the angular velocity vector of a body to the time derivatives of Euler angles, which in some way limits their usefulness in modelling robotic systems.

2.4 Homogeneous transformations matrix using D-H convention

Given the above considerations, a general transformation of any vector ${}^{i}\boldsymbol{p}$ given with respect to a moving frame *i* into its representation in reference frame *j* denoted by ${}^{j}\boldsymbol{p}$ can be described by following vector equation:

$${}^{j}\boldsymbol{p} = {}^{j}\boldsymbol{p}_{i} + {}^{j}\boldsymbol{R}_{i}\,{}^{i}\boldsymbol{p}. \qquad (14)$$

where ${}^{j}\boldsymbol{p}_{i}$ is position vector of the origin of the moving frame with respect to reference frame (1). The same applies to position vector of the arbitrary point M of the robotic segment.

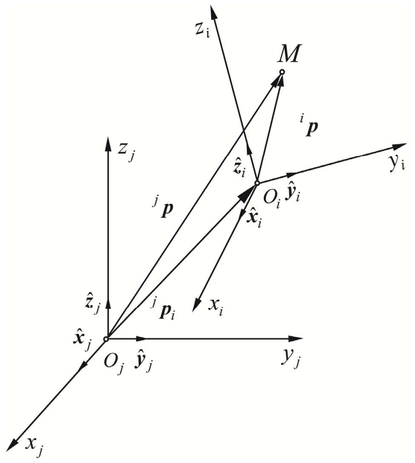

Figure 5. Transformation of vector from one coordinate system to another

Using homogeneous coordinates, by which position vector of a point is given as [5]:

$$\boldsymbol{p}^{*} = \begin{bmatrix} p_x & p_y & p_z & 1 \end{bmatrix}^{T}, \qquad (15)$$

equation (14) can be written if the following compact matrix form:

$${}^{j}\boldsymbol{p}^{*} = {}^{j}\boldsymbol{T}_{i}\,{}^{i}\boldsymbol{p}^{*}. \qquad (16)$$

where:

$${}^{j}\boldsymbol{T}_{i} = \begin{bmatrix} {}^{j}\boldsymbol{R}_{i} & \vdots & {}^{j}\boldsymbol{p}_{i} \\ \cdots & \vdots & \cdots \\ 000 & \vdots & 1 \end{bmatrix}. \qquad (17)$$

This 4x4 matrix is homogenous transformation matrix which maps homogenous position vector from one coordinate system to another. If several transformations are performed successively, transformation matrix describing total transformation is obtained by multiplication of transformation matrices describing those successive transformations. Homogenous transformation matrix describing rotation of angle θ_i provided by revolute joint whose axis is z_i is denoted by:

$$\mathbf{T}_{rot}(z_i,\theta_i)\ . \tag{18}$$

while homogenous transformation matrix describing translation for d_i provided by prismatic joint whose axis is $\mathbf{z}_i$ is denoted by :

$$\mathbf{T}_{trans}(z_i,d_i)\ . \tag{19}$$

Denavit and Hartenberg showed that general transformation between two coordinate systems requires knowledge of the 4 parameters [18]. These parameters called the Denavit-Hartenberg parameters (DH parameters) became standard for describing the robot kinematics. Figure 6 shows the DH parameters for the case of two joints. Within this convention, z axis is in the direction of the joint axis, x axis is parallel to the normal vector z_{n-1}×z_n, and y axis follows from the x and z axis to establish right-handed coordinate system.

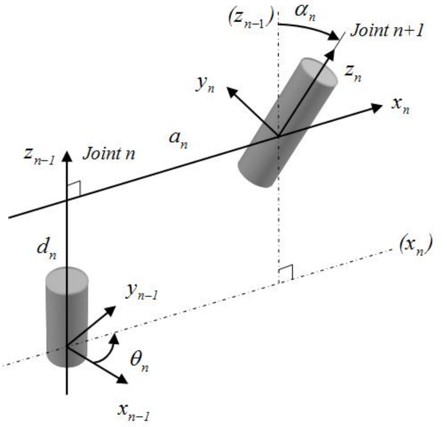

Figure 6. Frames and parameters in D-H convention

D-H parameters are:

- a_n - the distance between the axis z_{n-1} and z_n,
- α_n - the angle between joint axes z_{n-1} and z_n,
- d_n - the offset along z_{n-1} to the x_n
- θ_n - the angle between the joint axes x_{n-1} and x_n.

Since joints used in manipulators provide usually rotation or translation, only one parameter is variable, while other three are constant. In general, homogeneous transformation matrix based on D-H convention describing motion of link *n* provided by actuator in joint *n-1* with respect to frame attached to joint *n-1*, ${}^{n-1}\boldsymbol{T}_n$, is obtained by matrix multiplication:

$$\boldsymbol{T}_{rot}\left(z_{n-1},\theta_n\right)\boldsymbol{T}_{trans}\left(z_{n-1},d_n\right)\boldsymbol{T}_{trans}\left(x_n,a_n\right)\boldsymbol{T}_{rot}\left(x_n,\alpha_n\right),\tag{20}$$

$${}^{n-1}\boldsymbol{T}_n=\begin{bmatrix} c\theta_n & -s\theta_n\,c\alpha_n & s\theta_n\,s\alpha_n & a_n\,c\theta_n \\ s\theta_n & c\theta_n\,c\alpha_n & -c\theta_n\,s\alpha_n & a_n\,s\theta_n \\ 0 & s\alpha_n & c\alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix}.\tag{21}$$

2.5 Rodrigues' approach

Conversion from angle and vector representation of orientation to rotation matrix may be achived using *Rodrigues' rotational formula* [3]:

$$\mathbf{R}=\mathbf{I}+(1-\cos q)\left(\mathbf{e}^d\right)^2+\sin q\mathbf{e}^d\ , \tag{22}$$

where q is angle of rotation about axis determined by unit vector $\hat{\mathbf{e}}$, $\mathbf{e}^d$ is skew symmetric matrix of vector $\hat{\mathbf{e}}$ and $\mathbf{I}$ is 3×3 identity matrix. In order to obtain general transformation matrix in the form of (22) that would apply for revolute as well for prismatic joint *j*, additional parameter $\overline{\xi}_j$ is introduced so that it has value 1 for rotation and 0 for translation, and following equation is obtained:

$${}^{j-1}\mathbf{A}_j=\mathbf{I}+\overline{\xi}_j\left[(1-\cos q^j)\left(\mathbf{e}_j^d\right)^2+\sin q^j\mathbf{e}_j^d\right],\tag{23}$$

Position of the centre of mass C_i of the *i*-th segment w.r.t. to fixed frame is given by equation:

$$\overline{OC_i}=\vec{r}_{Ci}=\sum_{k=1}^{i}\left(\boldsymbol{\rho}_{kk}+\xi_k\mathbf{e}_k q_k\right)+\boldsymbol{\rho}_i\ , \tag{24}$$

where $\boldsymbol{\rho}_{kk}=\overrightarrow{O_k'O_{k+1}}$ and $\overrightarrow{O_{i+1}C_i}=\boldsymbol{\rho}_i$ are suitable position vectors (Figure 7). Parameter ξ_k has value 0 for rotation and 1 for translation; q is translational or angular displacement.

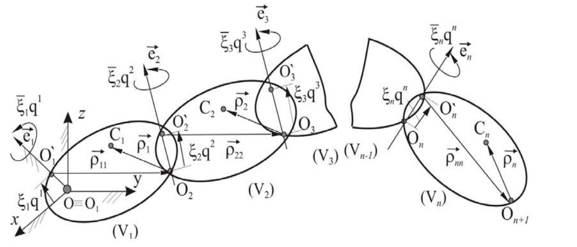

Figure 7. Rodrigues' approach: representative vectors in open chain robot mechanism

Position vector of the end-effector tip w.r.t. to fixed frame is:

$$\boldsymbol{r}_E=\sum_{k=1}^{n}\boldsymbol{A}_{0,k}\left\{\left(\boldsymbol{\rho}_{kk}+\xi_k q_k\boldsymbol{e}_k\right)\right\}, \tag{25}$$

where $\boldsymbol{A}_{0,k}={}^0\boldsymbol{A}_1\,{}^1\boldsymbol{A}_2\cdots{}^{k-1}\boldsymbol{A}_k$. It is important to say that all vectors are given with respect to local frame $C_i\xi_i\eta_i\zeta_i$.

3. SPATIAL DESCRIPTIONS AND TRANSFORMATIONS IN QUATERNION SPACE

3.1. Unit quaternions

The quaternion representation of orientation due to *Hamilton*, while largely superseded by the simpler vector representations of *Gibbs* and *Grassmann* is extremely useful for problems in robotics that result in representational singularities in the vector/matrix notation [1]. Unit quaternion is a convenient and compact mathematical notation for describing the attitude of rigid bodies in three dimensions.

Quaternions are hyper complex numbers that can be written as the linear combination $a+bi+cj+dk$ with $i^2 = j^2 = k^2 = -1$. Here a represents real part, while b, c and d are imaginary parts or the componests of pure quaternion [19]. They can be also interpreted as quadruple $\boldsymbol{q} = (q_1, q_2, q_3, q_4)$ or via vector and scalar representation $\boldsymbol{q} = (s, \boldsymbol{v})$. Late formulation is very suitable for rotation description.

Expressing rotations in 3D via unit quaternions instead of matrices has following advantages [20]:

- Concatenating rotations is computationally faster and numerically more stable.
- Since gimbal lock is innate to the matrix representation of Euler angles, this problem does not appear in the quaternion representation.
- Unambiguousness
- Extracting the angle and axis of rotation is simple
- Interpolation is more straightforward.

As indicated before, from Euler rotational theorem it can be concluded that any rotation in three dimensional space can be described by unit vector indicating the direction of rotational axis and a rotational angle θ. Using extension of Euler formula, this transformation in unit quaternion space becomes:

$$\begin{aligned} \boldsymbol{q} &= e^{\frac{1}{2}\theta\left(u_x\hat{\boldsymbol{i}}+u_y\hat{\boldsymbol{j}}+u_z\hat{\boldsymbol{k}}\right)} = \\ &= \left[cos(\theta/2), sin(\theta/2)\left(u_x, u_y, u_z\right)\right] = (s, \boldsymbol{v}), \quad (26) \\ &= \left[cos\,\bar{\theta}, sin\,\bar{\theta}\left(u_x, u_y, u_z\right)\right] \end{aligned}$$

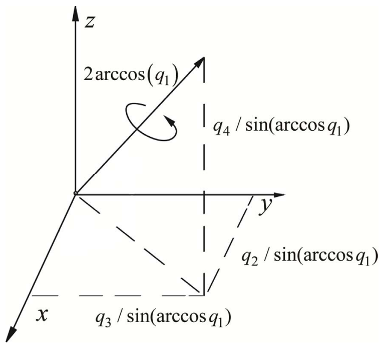
z
$2\arccos(q_1)$
$q_4 / \sin(\arccos q_1)$
y
$q_2 / \sin(\arccos q_1)$
x
$q_3 / \sin(\arccos q_1)$

Figure 8. Visualisation of unit quaternion parametres through relation with axis-angle presentation

hereinafter, $\bar{\theta} = \frac{\theta}{2}$. Thus, if rotation is given in the axis-angle form where rotational axis is defined by unit vector $\hat{\boldsymbol{u}} = u_x\hat{\boldsymbol{i}} + u_y\hat{\boldsymbol{j}} + u_z\hat{\boldsymbol{k}}$ and by angle of rotation θ, quaternion representing this rotation is:

$$\boldsymbol{q} = \left(cos\,\bar{\theta}, sin\,\bar{\theta}\left(u_x, u_y, u_z\right)\right) = (s, \boldsymbol{v}). \quad (27)$$

To define pure rotation, quaternions must satisfy $|\boldsymbol{q}| = 1$, i.e. pure rotation is defined by unit quaternions.

There is an extensive review of quaternion mathematic features in the literature [10]. In this paper, only basic quaternion operations required for deriving kinematic algorithms in robotics are considered.

For given unit quaternions $\boldsymbol{q}_1 = (s_1, \boldsymbol{v}_1)$ and $\boldsymbol{q}_2 = (s_2, \boldsymbol{v}_2)$, quaternion addition is defined as:

$$\boldsymbol{q}_1 + \boldsymbol{q}_2 = (s_1, \boldsymbol{v}_1) + (s_2, \boldsymbol{v}_2) = (s_1 + s_2, \boldsymbol{v}_1 + \boldsymbol{v}_2), \quad (28)$$

and quaternion product is defined as:

$$\boldsymbol{q}_1 \otimes \boldsymbol{q}_2 = \left(s_1 s_2 - \boldsymbol{v}_1\boldsymbol{v}_2, s_1\boldsymbol{v}_2 + s_2\boldsymbol{v}_1 + \boldsymbol{v}_1 \times \boldsymbol{v}_2\right). \quad (29)$$

Quaternion multiplication is associative and distributive across addition, but not commutative.

Two successive pure rotations represented by unit quaternion representatives $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ can be achieved by a single rotation about an appropriately chosen axis. Unit quaternion corresponding to this transformation is obtained from quaternion product $\boldsymbol{q}_1 \otimes \boldsymbol{q}_2$, analogously to the resultant rotational matrix representing total rotation in Cartesian space is obtained by successive rotational matrix multiplication.

For a given unit quaternion $\boldsymbol{q} = (q_1, q_2, q_3, q_4)$, corresponding rotational matrix $\boldsymbol{R}(\boldsymbol{q})$ is obtained by following formula:

$$\begin{bmatrix} q_2^2 - q_3^2 - q_4^2 + q_1^2 & 2\left(q_2q_3 - q_4q_1\right) & 2\left(q_2q_4 + q_3q_1\right) \\ 2\left(q_2q_3 + q_4q_1\right) & -q_2^2 + q_3^2 - q_4^2 + q_1^2 & 2\left(q_3q_4 - q_2q_1\right) \\ 2\left(q_2q_4 - q_3q_1\right) & 2\left(q_3q_4 + q_2q_1\right) & -q_2^2 - q_3^2 + q_4^2 + q_1^2 \end{bmatrix} \quad (30)$$

3.2. Dual quaternions

Dual Quaternions (DQ) were proposed by *William Kingdom Clifford* in 1873. and they can be considered as extension of quaternions since they represent both rotations and translations of rigid body-it's spatial displacement [12]. The dual-quaternion model is an accurate, computationally efficient, robust, and flexible method of representing rigid transforms and should not be overlooked [16]. They can be defined as quaternions whose parameters are dual numbers and thus are represented by eight-dimensional vector:

$$\hat{\boldsymbol{q}} = \left(s, x, y, z, s_\varepsilon, x_\varepsilon, y_\varepsilon, z_\varepsilon\right) = \boldsymbol{q} + \varepsilon\boldsymbol{q}_\varepsilon = \left(\hat{s}, \hat{x}, \hat{y}, \hat{z}\right). (31)$$

where ε is dual factor, $\varepsilon^2=0$, $\varepsilon\neq0$. Since spatial displacements of a rigid body in three-dimensional space have six degrees of freedom, dual quaternions are subject to two constraints:

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1, \quad (32)$$

$$q_1q_{1\varepsilon} + q_2q_{2\varepsilon} + q_3q_{3\varepsilon} + q_4q_{4\varepsilon} = 0. \quad (33)$$

Some important operations with dual quaternions (from the robotics point of view) are presented bellow. Dual quaternion multiplication is defined as:

$$\begin{aligned} \hat{\boldsymbol{q}}_1 \otimes \hat{\boldsymbol{q}}_2 &= \left(\boldsymbol{q}_1 + \varepsilon\boldsymbol{q}_{1\varepsilon}\right) \otimes \left(\boldsymbol{q}_2 + \varepsilon\boldsymbol{q}_{2\varepsilon}\right) = \\ &= \boldsymbol{q}_1 \otimes \boldsymbol{q}_2 + \varepsilon\left(\boldsymbol{q}_1 \otimes \boldsymbol{q}_{2\varepsilon} + \boldsymbol{q}_{1\varepsilon} \otimes \boldsymbol{q}_2\right) \end{aligned}. \quad (34)$$

Dual conjugate of $\hat{\boldsymbol{q}}$ is given by:

$$\overline{\hat{\boldsymbol{q}}} = \boldsymbol{q} - \varepsilon \boldsymbol{q}_{\varepsilon} . \tag{35}$$

A second conjugation operator of dual quaternion is defined as [13]:

$$\overline{\hat{\boldsymbol{q}}}^{*} = \left(\left(s, -x, -y, -z, -s_{\varepsilon}, x_{\varepsilon}, y_{\varepsilon}, z_{\varepsilon}\right)\right) . \tag{36}$$

If vector $\boldsymbol{p}$ is defined in Cartesian space as $\boldsymbol{p} = p_x \boldsymbol{i} + p_y \boldsymbol{j} + p_z \boldsymbol{k}$, representation of this vector in unit quaternion space is:

$$\boldsymbol{p} = 0 + p_x \boldsymbol{i} + p_y \boldsymbol{j} + p_z \boldsymbol{k} , \tag{37}$$

and in in dual quaternion space:

$$\hat{\boldsymbol{p}} = 1 + \varepsilon \left(p_x \boldsymbol{i} + p_y \boldsymbol{j} + p_z \boldsymbol{k} \right) , \tag{38}$$

or:

$$\hat{\boldsymbol{p}} = \boldsymbol{p} + \varepsilon \boldsymbol{p}_{\varepsilon} = \left(1,0,0,0\right) + \varepsilon \left(0, p_x, p_y, p_z\right) . \tag{39}$$

Representing pure rotation

Rotation of the vector $\boldsymbol{p}$ about axis $\boldsymbol{a}$ for angle α is represented in dual quaternion space by following equation:

$$\hat{\boldsymbol{p}}' = \hat{\boldsymbol{q}}_R \otimes \hat{\boldsymbol{p}} \otimes \overline{\hat{\boldsymbol{q}}}_R^{*} , \tag{40}$$

where dual quaternion of transformed vector $\boldsymbol{p}$ in the form:

$$\hat{\boldsymbol{p}}' = 1 + \varepsilon \left(p'_x \boldsymbol{i} + p'_y \boldsymbol{j} + p'_z \boldsymbol{k} \right) , \tag{41}$$

is obtained from:

$$\begin{aligned} \hat{\boldsymbol{q}}_R &= \boldsymbol{q}_R + \varepsilon \boldsymbol{q}_{R\varepsilon} = \boldsymbol{q}_R = \\ &= \left(\cos \overline{\alpha}, \sin \overline{\alpha} \left(a_x, a_y, a_z \right) \right), \boldsymbol{q}_{R\varepsilon} = \left(0,0,0,0\right) \cdot \end{aligned} \tag{42}$$

Representing pure translation

Dual quaternion vector corresponding to translation described by translation vector $\boldsymbol{T} = \left(t_x \boldsymbol{i} + t_y \boldsymbol{j} + t_z \boldsymbol{k} \right)$ is:

$$\begin{aligned} \hat{\boldsymbol{q}}_T &= \left(1,0,0,0\right) + \frac{\varepsilon}{2}\left(0, t_x, t_y, t_z\right) = \\ &= 1 + \frac{\varepsilon}{2}\left(t_x \boldsymbol{i} + t_y \boldsymbol{j} + t_z \boldsymbol{k} \right) \end{aligned} , \tag{43}$$

Dual quaternion representing vector $\boldsymbol{p}$ after translation defined by $\boldsymbol{T}$ is:

$$\hat{\boldsymbol{p}}' = \hat{\boldsymbol{q}}_T \otimes \hat{\boldsymbol{p}} \otimes \overline{\hat{\boldsymbol{q}}}_T^{*} , \tag{44}$$

from which dual quaternion vector of transformed vector $\boldsymbol{p}$ is obtained in the form:

$$\hat{\boldsymbol{p}}' = 1 + \varepsilon \left(p'_x \boldsymbol{i} + p'_y \boldsymbol{j} + p'_z \boldsymbol{k} \right) . \tag{45}$$

A pure translation of a vector can be represented as vector addition of a given vector $\boldsymbol{p}$ and translational vector $\boldsymbol{T}$, thus translation of the vector in dual quaternion space can be represented with another identity:

$$\hat{\boldsymbol{p}}' = 1 + \varepsilon \left(\left(p_x + t_x \right) \boldsymbol{i} + \left(p_y + t_y \right) \boldsymbol{j} + \left(p_z + t_z \right) \boldsymbol{k} \right) . \tag{46}$$

Representing rigid transformations

Let us assume that vector $\boldsymbol{p}$ was rotated about axis $\boldsymbol{a}$ for an angle α and after that translated by translation vector $\boldsymbol{T}$. This transformation is described in dual quaternion space in a following manner:

$$\hat{\boldsymbol{p}}' = \hat{\boldsymbol{q}}'_T \otimes \left(\hat{\boldsymbol{q}}_R \otimes \hat{\boldsymbol{p}} \otimes \overline{\hat{\boldsymbol{q}}}_R^{*} \right) \otimes \overline{\hat{\boldsymbol{q}}}_T^{\prime *} , \tag{47}$$

where $\hat{\boldsymbol{q}}_R$ is given by (42). Special attention should be paid that $\hat{\boldsymbol{q}}'_T$ is dual quaternion describing pure translation defined by translational vector ***T′***-which is rotated vector $\boldsymbol{T}$:

$$\boldsymbol{T}' = \left(\hat{\boldsymbol{q}}_R \otimes \hat{\boldsymbol{T}} \otimes \overline{\hat{\boldsymbol{q}}}_R^{*} \right) . \tag{48}$$

If we assume transformation in which translation given by $\boldsymbol{T}$ is performed first, and rotation given by $\hat{\boldsymbol{q}}_R$ second, transformated vector $\hat{\boldsymbol{p}}'$ will be obtained from following:

$$\hat{\boldsymbol{p}}' = \hat{\boldsymbol{q}}_R \otimes \left(\hat{\boldsymbol{q}}_T \otimes \hat{\boldsymbol{p}} \otimes \overline{\hat{\boldsymbol{q}}}_T^{*} \right) \otimes \overline{\hat{\boldsymbol{q}}}_R^{*} . \tag{49}$$

3.3. Direct kinematics algorithm in quaternion space

In this paragraph, it will be considered how vector transformations in dual quaternion space described before are related to direct kinematics algorithm.

To define an efficient kinematic algorithm, a suitable reference frame should be established, as well as local frames moving together with robot segments. Rotations are described by following dual quaternion:

$$\begin{aligned} &\hat{\boldsymbol{q}}_R = \boldsymbol{q}_R + \varepsilon \boldsymbol{q}_{R\varepsilon} = \left(\cos \overline{\alpha}, \sin \overline{\alpha} \left(a_x, a_y, a_z \right) \right), \\ &\boldsymbol{q}_{R\varepsilon} = \left(0,0,0,0\right) \end{aligned} , \tag{50}$$

where α is value of joint angle for rotational axis represented by vector $\boldsymbol{a}$ in three dimensional space. These vectors are determined with respect to the fixed frame for a robot in a initial position. Rotations for angle α about axis parallel to *x,y,z* axis of fixed coordinate system will be described respectively:

$$\hat{\boldsymbol{q}}_{R,x,\alpha} = \left(\cos \overline{\alpha}, \sin \overline{\alpha} \left(1, 0, 0\right) \right) , \tag{51}$$

$$\hat{\boldsymbol{q}}_{R,y,\alpha} = \left(\cos \overline{\alpha}, \sin \overline{\alpha} \left(0, 1, 0\right) \right) , \tag{52}$$

$$\hat{\boldsymbol{q}}_{R,z,\alpha} = \left(\cos \overline{\alpha}, \sin \overline{\alpha} \left(0, 0, 1\right) \right) . \tag{53}$$

As stated before, we consider clockwise rotations. That mentioned if the axis of rotation is in opposite direction of unit axis of fixed coordinate system, we adopt that the rotation is performed for angle -α about axis in the same direction as referent axis of coordinate frame and substitute $\sin\left(\alpha / 2\right)$ by $-\sin\left(\alpha / 2\right)$ in $\hat{\boldsymbol{q}}_R$ expression.

In case of prismatic joints, translation described by translational vector $\boldsymbol{T} = \left(t_x\boldsymbol{i} + t_y\boldsymbol{j} + t_z\boldsymbol{k}\right)$ will be represented by dual quaternion:

$$\hat{\boldsymbol{q}}_T = 1 + \frac{\varepsilon}{2}\left(t_x\boldsymbol{i} + t_y\boldsymbol{j} + t_z\boldsymbol{k}\right), \tag{54}$$

Translations for value *a* along axis parallel to *x,y,z* axis of fixed frame will be respectively:

$$\hat{\boldsymbol{q}}_{T,x,a} = 1 + \frac{\varepsilon}{2}\left(a\boldsymbol{i} + 0\boldsymbol{j} + 0\boldsymbol{k}\right), \tag{55}$$

$$\hat{\boldsymbol{q}}_{T,y,a} = 1 + \frac{\varepsilon}{2}\left(0\boldsymbol{i} + a\boldsymbol{j} + 0\boldsymbol{k}\right). \tag{56}$$

$$\hat{\boldsymbol{q}}_{T,z,a} = 1 + \frac{\varepsilon}{2}\left(0\boldsymbol{i} + 0\boldsymbol{j} + a\boldsymbol{k}\right). \tag{57}$$

Consider a manipulator with *n* degrees of freedom, and let n_R be a number of rotational joints. Dual quaternion which defines orientation of the end-effector for given joint inputs is given by following equation:

$$\hat{\boldsymbol{q}}_{0,n_R} = \hat{\boldsymbol{q}}_{1_R} \otimes \hat{\boldsymbol{q}}_{2_R} \otimes \hat{\boldsymbol{q}}_{3_R} \ldots \otimes \hat{\boldsymbol{q}}_{n_R}, \tag{58}$$

where $\hat{\boldsymbol{q}}_{i_R}$ is dual quaternion representing transformation provided by *i*-th rotational joint. This transformation can be represented in unit quaternion space as following:

$$\boldsymbol{q}_{0,n_R} = \boldsymbol{q}_{1_R} \otimes \boldsymbol{q}_{2_R} \otimes \boldsymbol{q}_{3_R} \ldots \otimes \boldsymbol{q}_{n_R}. \tag{59}$$

To verify these results, we can transform dual quaternion $\boldsymbol{q}_{0,n}$ into rotation matrix using (30).

To determine position of the end-effector, we consider radius vector of the end-effector $\boldsymbol{p}$ as the sum of vectors along respective segments moving together with segments. Dual quaternion vector transformations given in subparagraph 3.2 are used. Equations (47) and (49) are applied successifully to define correspondent transformations provided by rotational and prismatic joints. Special attention should be paid to the order of applied successive vector position transformations and especcially to the case where rotation is followed by translation.

4. SIMULATION RESULTS PERFORMED ON HUMAN CENTRIFUGE (3 DOF MANIPULATOR)

To demonstrate presented direct kinematics algorithm in dual quaternion space, human centrifuge (Figure 9) will be considered. Human centrifuge is dynamic flight simulatior which purpose is to simulate as closely as possible real flight conditions and to provide an effective pilot training [21]. This device can be modeled as 3 DoF manipulator with rotational joints [22]. Main motion is rotation of the centrifuge arm about vertical, planetary axis. Arm carries gondola which is able to rotate about two axes, pitch and roll. These two axes intersect in the center of the gondola. Pilot seat is considered as end-effector and it is assumed to be placed in the center of the gondola. First, a reference fixed frame is determined and rotational axis are defined (Figure 10).

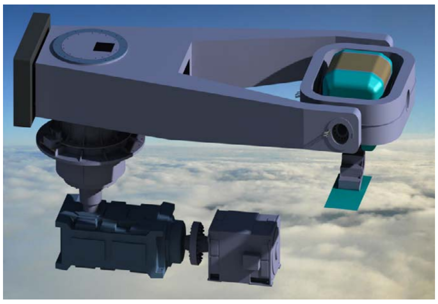

Figure 9. Human centrifuge

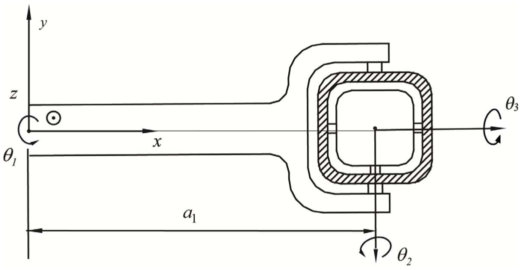

Figure 10. Human centrifuge: Rotational axes in initial position (a view from above)

By usage of Rodrigues' approach [3], following rotational matrices and position vector of the end-effector are obtained:

$${}^0\boldsymbol{R}_1 = \begin{bmatrix} c_1 & -s_1 & 0 \\ s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{60}$$

$${}^1\boldsymbol{R}_2 = \begin{bmatrix} c_2 & 0 & -s_2 \\ 0 & 1 & 0 \\ s_2 & 0 & c_2 \end{bmatrix}, \tag{61}$$

$${}^2\boldsymbol{R}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_3 & -s_3 \\ 0 & s_3 & c_3 \end{bmatrix}, \tag{62}$$

$${}^0\boldsymbol{R}_3 = \begin{bmatrix} c_1c_2 & -s_1c_3 - c_1s_2s_3 & s_1s_3 - c_1s_2c_3 \\ s_1c_2 & c_1c_3 - s_1s_2s_3 & -c_1s_3 - s_1s_2c_3 \\ s_2 & c_2s_3 & c_2c_3 \end{bmatrix}. \tag{63}$$

$${}^0\boldsymbol{r}_E = \sum_{k=1}^{n}\left[\boldsymbol{A}_{0,k}\right]\left\{\left(\boldsymbol{\rho}_{kk} + \xi_k q_k \boldsymbol{e}_k\right)\right\} = \left[a_1c_1 \quad a_1s_1 \quad 0\right]^T. \tag{64}$$

This rotational matrices and position vector are given for the purpose of comparison with method given here. Planetary, roll and pitch rotations in dual quaternion space are given by:

$$\hat{\boldsymbol{q}}_{1_R} = \hat{\boldsymbol{q}}_{rot}\left(z_0, \theta_1\right) = \left(\cos\bar{\theta}_1, \sin\bar{\theta}_1\left(0,0,1\right)\right), \tag{65}$$

$$\begin{aligned}\hat{\boldsymbol{q}}_{2_R} &= \hat{\boldsymbol{q}}_{rot}\left(z_1, \theta_2\right) = \hat{\boldsymbol{q}}_{rot}\left(y_0, -\theta_2\right) = \\ &= \left(\cos\bar{\theta}_2, -\sin\bar{\theta}_2\left(0,1,0\right)\right)\end{aligned}, \tag{66}$$

$$\begin{aligned}\hat{\boldsymbol{q}}_{3_R} &= \hat{\boldsymbol{q}}_{rot}\left(z_2, \theta_3\right) = \hat{\boldsymbol{q}}_{rot}\left(x_0, \theta_3\right) = \\ &= \left(\cos\left(\bar{\theta}_3\right), \sin\left(\bar{\theta}_3\right)\left(1,0,0\right)\right)\end{aligned}, \tag{67}$$

$$\hat{\boldsymbol{q}}_{0,3} = \hat{\boldsymbol{q}}_{\boldsymbol{1}_R} \otimes \hat{\boldsymbol{q}}_{\boldsymbol{2}_R} \otimes \hat{\boldsymbol{q}}_{\boldsymbol{3}_R} = \boldsymbol{q}_{0,3} + \varepsilon \boldsymbol{q}_{0,3\varepsilon} . \qquad (68)$$

Here, rotational axis z_0 is parallel to z axis of fixed frame, z_1 is parallel to y axis, and z_2 is parallel to x axis axis of fixed frame. Dual quaternion representing orientation of the end-efector is:

$$\boldsymbol{q}_{0,3} = \begin{bmatrix} c\overline{\theta}_3 c\overline{\theta}_2 c\overline{\theta}_1 - s\overline{\theta}_1 s\overline{\theta}_2 s\overline{\theta}_3 \\ c\overline{\theta}_2 c\overline{\theta}_1 s\overline{\theta}_3 + c\overline{\theta}_3 s\overline{\theta}_1 s\overline{\theta}_2 \\ c\overline{\theta}_1 c\overline{\theta}_3 s\overline{\theta}_2 + s\overline{\theta}_1 c\overline{\theta}_2 s\overline{\theta}_3 \\ c\overline{\theta}_2 c\overline{\theta}_3 s\overline{\theta}_1 + c\overline{\theta}_1 s\overline{\theta}_2 s\overline{\theta}_3 \end{bmatrix}, \boldsymbol{q}_{0,3\varepsilon} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \qquad (69)$$

Using (30), equality of results obtained by (63) and (69) can be obtained.

Since end-effector is placed in the center of the gondola where roll and pitch axis intersect, rotation of the second and the third axis will not affects its position. Vector position in initial position of the robot in Cartesian space is given by vector $\boldsymbol{p} = \begin{bmatrix} a_1 & 0 & 0 \end{bmatrix}^T$ and in dual quaternion space by:

$$\hat{\boldsymbol{p}} = \left(1,0,0,0\right) + \varepsilon\left(0, a_1, 0, 0\right) = 1 + \varepsilon\left(a_1 \boldsymbol{i}\right). \qquad (70)$$

After rotation of the centrifuge arm, by the usage of (40), a new position of the end-effector in dual quaternion space is:

$$\hat{\boldsymbol{p}}' = \hat{\boldsymbol{q}}_{1_R} \otimes \hat{\boldsymbol{p}} \otimes \overline{\hat{\boldsymbol{q}}}^*_{1_R} = 1 + \varepsilon\left(0, a_1 c\theta_1, a_1 s\theta_1, 0\right), \quad (71)$$

or in Cartesian space with respect to fixed frame:

$$\boldsymbol{p}' = \begin{bmatrix} a_1 c_1 & a_1 s_1 & 0 \end{bmatrix}^T . \qquad (72)$$

This example shows the simplicity and compactness of presented direct kinematics algorithm. Comparing this approach with Rodrigues' approach applied to this particular manipulator [3], [23], forward kinematics algorithm presented here proved to be simpler and computational cost is smaller. This approach is particularly advantageous in overcoming the problem of singularity. Namely, in previous work [23] it is determined that this manipulator has a singular position in $q_2 = \pm\pi/2$. In this position first and third axes align and from (63), (11), (12) and (13) it can be seen that it is impossible to determine or control orientation by the means of three Euler angles. If we notice that axes of this manipulator are directed in the same manner as zy'x" sequence of Euler angles in (8), from directions of rotational axes it can be concluded that $\theta = -q_2$. In this singular position, one degree of freedom is lost and only the difference $(\psi - \varphi)$ can be controlled. This results in numerical ill-conditioning and unpredictable behaviour causing potentially damage situation. On the other hand, from (69) it can be seen that all terms of quaternion representing this orientation are determined.

5. CONCLUSION

In this paper, compact overview of mathematical constructs used most often for representation of the position and attitude in robotics is given. Advantages of robot kinematic modelling in quaternion and dual quaternion space stemmed from singularity free, numerically stable and suitable for interpolation representation of attitude. Vector transformations using dual quaternions are presented. These transformations are used as foundation for new, simple and compact direct kinematic algorithm in dual quaternion space. Developed algorithm is demonstrated on human centrifuge which is modelled as 3 DoF manipulator with rotational joints. According to the results from the section 4, it can be concluded that the advantage of presented quaternion based algorithm, with respect to Rodrigues' approach, is smaller computational cost. Also, it can be seen that in application of presented quaternion algorithm the problem of singularity, innate to the matrix representation of Euler angles, and particularly important to control of actuators in robotics does not appear.

ACKNOWLEDGMENT

This work was created within the research project that is supported by the Ministry of Education, Science And Technological Environment of Republic of Serbia: No (35023) and is partially supported by the Ministry of Education, Science And Technological Environment of Republic of Serbia, No (35006).

REFERENCES

[1] Siciliano, B. and Oussama K.: *Springer handbook of robotics*. Springer Verlag Berlin Heidelberg, 2008.

[2] Paul, R.: *Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators*. MIT Press, Cambridge, MA, 1981.

[3] Čović V. and Lazarević M.: *Mechanics of Robots*, Faculty of Mechanical Engineering, University of Belgrade, 2009.

[4] Murray, R. and Sastry, S.: *A mathematical introduction to robotic manipulation*, CRC Press, Inc. Boca Raton, FL, USA, 1994.

[5] Tsai, L-W.: *Robot analysis: the mechanics of serial and parallel manipulators*, John Wiley and Sons, New York, 1999.

[6] Dai, J.S.: An historical review of the theoretical development of rigid body displacements from Rodrigues parameters to the finite twist, Mechanism and Machine Theory, Vol. 41, No. 1, pp. 41-52, 2006.

[7] Brannon, R.M.: A review of useful theorems involving proper orthogonal matrices referenced to three dimensional physical space, www.mech.utah.edu/brannon/public/rotation.pdf, 2002.

[8] Debnath, L.: The legacy of Leonhard Euler–a tricentennial tribute, International Journal of Mathematical Education in Science and Technology, Vol. 40, No. 3, pp. 353-388, 2009.

[9] Diebel, J.: Representing attitude: Euler angles, unit quaternions, and rotation vectors, http://ai.stanford.edu/ ~diebel/, 2006.

[10] Dam, E.B., Koch, M. and Lillholm, M.: *Quaternions, interpolation and animation*, Technical Report DIKU-TR-98/5, Datalogisk Institut, Københavns Universitet, 1998.

[11] Corke, P. *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Springer Tracts in Advanced Robotics, Vol. 73, Springer Verlag Berlin Heidelberg, 2011.

[12] Funda, J., Russell H. T. and Richard P.P.: On homogeneous transforms, quaternions, and computational efficiency, IEEE Transactions on Robotics and Automation, 6(3), pp. 382–388, 1990.

[13] Gouasmi, M.: Robot Kinematics, using Dual Quaternions, IAES International Journal of Robotics and Automation (IJRA), Vol. 1, No. 1, pp. 13-30, 2012.

[14] Kucuk, S, and Zafer B.: Robot kinematics: Forward and inverse kinematics, Industrial Robotics Theory Modelling & Control, ARS/pIV, pp. 117-148, Germany, 2006.

[15] Foley, J.D., van Dam A., Feiner, S.K. and Hughes, J.F.: *Computer Graphics: Principles and Practice*, Addison-Wesley, Reading, MA, pp. 689–693, 1990.

[16] Kenwright, B.: A beginners guide to dual-quaternions: what they are, how they work, and how to use them for 3D character hierarchies, *Proceedings of the 20th International Conference on Computer Graphics, Visualization and Computer Vision,* 2012, Plzen, pp. 1-10 or paper A-29.

[17] Stets, J.D.: *Imaging Robot*, Dissertation, Technical University of Denmark, DTU, DK-2800 Kgs. Lyngby, Denmark, 2010.

[18] Hartenberg, R.S. and Denavit, J.: *Kinematic synthesis of linkages*, McGraw-Hill, New York, 1964.

[19] Zorić, N., Lazarević M., Simonović. A.:Multi-body kinematics and dynamics in terms of quaternions: Langrange formulation in covariant form: Rodriguez approach, FME Transactions, Vol. 38, No. 1, pp. 19-28, 2010.

[20] Hanson, A.J.: *Visualizing quaternions*, Morgan Kaufmann Publishers, Elsevier, San Francisco, 2006.

[21] Dančuo, Z., Rašuo, B., Kvrgić, V. and Zeljković, V.: Methodology of the Main Drive Selection for a Human Centrifuge, FME Transactions, Vol. 40, No. 2, pp. 69-74, 2012.

[22] Vidakovic, J., Lazarevic, M., Kvrgic, V., Dančuo, Z. and Lutovac, M.:Comparison of Numerical Simulation Models for Open Loop Flight Simulations in the Human Centrifuge. *PAMM*, *13*(1), pp. 485-486, 2013.

[23] Vidaković, J., Kvrgić, V., Ferenc, G., Dančuo, Z. and Lazarević, M.: Kinematic and Dynamic Model of the Human Centrifuge, *Proceedings of the 4th International Congress of Serbian Society of Mechanics*, 2013, Vrnjačka Banja, pp. 627-632.

НАПРЕДНИ КВАТЕРНИОНСКИ АЛГОРИТАМ ДИРЕКТНЕ КИНЕМАТИКЕ РОБОТА УКЉУЧУЈУЋИ ПРЕГЛЕД РАЗЛИЧИТИХ МЕТОДА КИНЕМАТИКЕ РОБОТА

Јелена Видаковић, Михаило Лазаревић, Владимир Квргић, Зорана Данчуо, Горан Ференц

Формулисање одговарајућих и ефикасних алгоритама кинематике робота је од суштинског значаја за анализу и развој серијских манипулатора. Кинематичко моделовање манипулатора се најчешће врши у Декартовом простору. Међутим, услед недостатака најзаступљенијих математичких оператора за дефинисање оријентације као што су Ојлерови углови и ротационе матрице, намеће се потреба за једнозначним, компактним, рачунски ефикасним методом за одређивање оријентације. Као решење овог проблема предлажу се јединични кватерниони као и развој кинематичких модела у простору дуалних кватерниона. У овом раду је дат преглед геометријских описа и трансформација које се могу применити у оквиру наведених простора како би се решили проблеми кинематике робота. Посебан акценат је на различитим математичким формализмима који се користе за дефинисање оријентације крутог тела, као што су ротационе матрице, Ојлерови углови, оса и угао ротације, јединични кватерниони, као и на њиховој узајамној вези. Предности кинематичког моделирања у простору кватерниона су истакнуте. Особине јединичних и дуалних кватерниона се анализирају са становишта роботике. Такође, дат је нови алгоритам директне кинематике робота у простору дуалних кватерниона. Овај алгоритам је примењен на хуманој центрифуги која је моделирана као троосни манипулатор.