On the Use of Quaternions, in the Translated Reference Frame Formalism

Claudiu Pozna

Department of Automation, Széchenyi István University,
Egyetem tér 1, 9026 Győr, Hungary

Department of Automation and Information Technology, Transilvania University of Brasov, Str. Mihai Viteazu 5, Corp V, et. 3, 500174 Brasov, Romania

E-mail: cp@unitbv.ro

Radu-Emil Precup

Department of Automation and Applied Informatics, Politehnica University of Timisoara, Bd. V. Parvan 2, 300223 Timisoara, Romania

Center for Fundamental and Advanced Technical Research, Romanian Academy – Timisoara Branch, Bd. Mihai Viteazu 24, 300223 Timisoara, Romania

E-mail: radu.precup@upt.ro

Abstract: The approaches dedicated to the geometric model of the object, in the object-oriented modeling framework, specific to robots, are focused on the computation of the position of the elements, relative to a global reference frame. One important problem, in this regard, is the direct geometry, which starts with the initial data represented by the generalized coordinates and parameters of each element (in its own reference frame) and calculates the configuration of the structure in the global reference frame. This paper extends the Authors' geometrical modeling approach, referred to as the translated reference frame formalism, by changing the mathematical apparatus of homogeneous transformations with that specific to quaternions. This new approach has three main advantages over the state-of-the art, namely, it simplifies substantially the configuration modeling, the transformation parameters are intuitive, and the computation is substituted with the operation of choosing a suitable transformation that belongs to a set of six homogeneous transformations. The suggested approach is validated by the computation of the geometric models of two illustrative robotic examples. The source codes are available in a public repository.

Keywords: direct geometry; geometric modeling; homogeneous operators; quaternions; translated reference frame formalism

1 Introduction

Modeling is a process of knowledge, that uses two mechanisms of thinking, approximation and conceptualization. The approximation refers to the simplification of the phenomenon by eluding the features considered insignificant (by the model designer), and the conceptualization refers to the replacement of the particular by the general, by the concept. It is widely acknowledged that the model is an abstraction of reality; the difference between the model and the reality is inherent and called perturbation. In order to be aware of this difference, the model design is preceded by the specification of the hypotheses, i.e., the conditions that state when the model fully reflects the phenomenon.

In the case of robotics, the object-oriented modeling framework organizes the model construction through a succession of increasingly precise model classes. The analysis and design principles specific to object-oriented framework are set in [1]. These principles are applied in [2] and [3] to create a software library that is used in engineering and teaching applications as suggestively illustrated in [4]. An application of object-oriented modeling to delta robots is described in [5]. The analysis and design principles are implemented in Modelica in [6], where predefined objects of Lego type are assembled in different configurations.

In the context of object-oriented modeling, a succession of three classes of models, namely geometric, kinematic and dynamic ones, is formulated in [7]. Each class contains the related assumptions (hypothesis), the properties they use, as well as the methods they offer. The mentioned classes are in a parent-child relationship, which means that they can inherit their methods and properties. The basis of this succession is the geometric model, which is the father of the kinematic one, which in turn is the father of the dynamic one.

The geometric model object contains the richest collection of hypotheses (eliminated, to a certain extent, by kinematic and dynamic models). Such examples of hypotheses are: the phenomena are out of time, there is no movement, bodies are rigid, interactions with the environment are non-existent, and there are no inertial effects. The properties of the geometric model refer to abstractions of the size and orientation of the bodies obtained by attaching reference (or coordinate) frames (or systems) to each element, as, for example, the lengths of the elements, the angles of rotation, and the displacements in the joints (i.e., the generalized coordinates).

The approaches specific to the geometric model object deal with the computation of the position of the elements relative to a global reference frame. Two types of problems are important in this regard:

(i) *The direct geometry*, which starts with the initial data represented by the generalized coordinates and parameters of each element (in its own reference frame), and calculates the configuration of the structure in the global reference frame.

(ii) *The inverse geometry*, which starts with the input data represented by the position of the gripper (effector), and calculates the generalized coordinates of the robot joints.

Several formalisms have been proposed to solve the direct geometry problem. They will be briefly discussed as follows. The Denavit-Hartenberg (DH) formalism [8] [9] is the most popular one, it is imagined for a structure with arbitrary axes (the general case), but it proves to be difficult to be applied to structures with parallel or perpendicular axes. The DH formalism model describes the technological dimensions of robot's elements through a sequence of four parameters, three constant ones and one variable one. The DH formalism is followed by Paul's approach [10], which uses DH parameters but is focused on structures with translation and rotation couplets, and Khalil and Kleinfinger's approach [11], which uses the same DH parameters but generalizes the formalism to closed and tree structures. A significant change is made by Craig in [12] in terms of moving the origin of the reference frame of the element from the upstream joint to the downstream joint. However, it is difficult to define the three parameters and to mathematically compute the homogeneous transformation. The DH and Craig's formalisms are included in software packages as that developed by Corke in [13] for Matlab and in the Robot Analyzer coordinated by Saha [14]. Another formalism, proposed by Gogu and his co-authors in [15] and [16], conceives a significant simplification that refers to particular structures with parallel or perpendicular joints axes. It is also important to mention the screw theory-based formalism [17-19], which proves to be a modern and efficient alternative to the DH formalism.

The paper is an extension of authors' recent approach given in [20] and referred to as translated reference frame formalism by changing the mathematical apparatus of homogeneous transformations with that specific to quaternions. This novel approach to solve the direct geometry problem is important with respect to the state-of-the-art discussed above because of three reasons, (a), (b) and (c). (a) It simplifies substantially the configuration modeling. (b) The transformation parameters are intuitive. (c) The computation is substituted with the operation of choosing a suitable transformation that belongs to a set of six homogeneous transformations. In addition, the quaternion-based approach proposed by this paper is advantageous as it is attractive for rotational modeling and for generalization to structures with arbitrary axes.

The paper is organized as follows: the stages of the translated reference frame formalism proposed in [20] are briefly recalled in the next section. The presentation of the mathematical apparatus associated to the formalism is prepared in Section 3 focused on quaternions. Sections 4 and 5 describe the proposed approach, in a particular version applied to structures with parallel and or perpendicular axes, and its generalization to arbitrary axes as well, respectively, and an algorithm is formulated in this regard. Section 6 gives the use of the suggested mathematical tools to calculate the Jacobian matrix. The theoretical results are validated in Section 7 in terms of two illustrative examples, their analytical solution and the software implementation and application of the algorithm. The conclusions highlighted in Section 8 conclude the paper.

2 Overview of Translated Reference Frame Formalism

Since each element of the robot's structure is defined in its own reference (or coordinate) frame (or system), the formalisms mentioned in the previous section are expressed as algorithms that ultimately determine the transformation operators from one reference frame to another one. More precisely, the parameters of each element are defined in its own (local) reference frame, and it is of interest to express their values in other reference frames (local or global ones). In this context, the position of the effector in the global reference frame is of interest, i.e.

$$\begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{o} \end{bmatrix} = \boldsymbol{f}(q_1, \ldots, q_n) \tag{1}$$

where **p** is the position of the gripper, **o** is the orientation of the gripper (in the Cartesian space), and q_1, …, q_n are the generalized coordinates (variables of joint space). The result in (1) is important in the representation of the configuration, the solution to the kinematic model (the calculation of the Jacobian matrix), and the construction of the dynamic model of the structure.

The translated reference frame formalism [20] is described and organized in terms of several rules that involve the variables illustrated in Figure 1. For example, for the joint i, the generalized variable is q_i, the origin of the reference frame is O_i, and the unit vector of the axis i is $\hat{\mathbf{u}}_{i-1}^{i}$ (defined in the reference frame $\{i–1\}$).

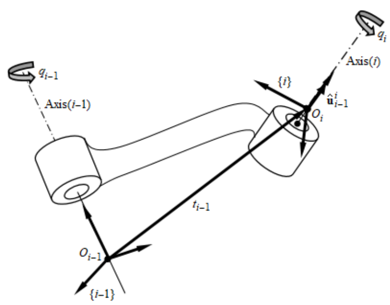

Figure 1
The link between the reference frames $\{i–1\}$ and $\{i\}$ (adapted from [20])

The formalism is organized in the following steps [20]:

1. The reference frames are attached to each element.

1.a. The origins of the reference frames are defined as the points O_1, …, O_n, on the axes of rotation or translation of each joint:

$$O_i \in Axis(i). \tag{2}$$

1.b. The generic reference frame of the first joint is defined so that one of the axes of the system coincides with the axis of rotation and / or translation of the joint:

$$\{0\} \mid \hat{\mathbf{x}}^0 \vee \hat{\mathbf{y}}^0 \vee \hat{\mathbf{z}}^0 \equiv \hat{\mathbf{u}}_0^1. \tag{3}$$

1.c. The chosen reference frame is translated to each chosen origin, obtaining a set of n reference frames. Since the joints have mutually parallel or perpendicular axes, each axis of the joints will overlap on one of the axes of the reference frame corresponding to that axis. The reference frame of the element i, namely $\{i\}$, is located on the i axis:

$$\{0\} \parallel \{i\}, \forall i = 1 \ldots n. \tag{4}$$

2. The transformation parameters from the reference frame $\{i\}$ to the reference frame $\{i–1\}$ is defined.

2.a. The position vector of the point O_i is expressed in the reference frame $\{i–1\}$.

2.b. The unit vector of the i axis is expressed in the reference frame $\{i–1\}$.

2.c. The generalized variable q_i is expressed, i.e., the angle of rotation or the length of translation at joint i.

3. The mathematical formulae of the formalism are applied.

3.a. Six homogeneous transformations have been identified to solve all configurations:

$$ {}^{i-1}_{i}\boldsymbol{T} = \begin{bmatrix} {}^{i-1}_{i}\boldsymbol{Q} & {}^{i-1}_{i}\boldsymbol{P} \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix} \tag{5}$$

where $\mathbf{Q} \in \{\mathbf{R}_x, \mathbf{R}_y, \mathbf{R}_z, \mathbf{I}\}$ is the rotation component of the operator, $\mathbf{R}_x$, $\mathbf{R}_y$ and $\mathbf{R}_z$ are the elementary angle rotations around the X, Y and Z axes attached to q_i:

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(q_i) & -\sin(q_i) \\ 0 & \sin(q_i) & \cos(q_i) \end{bmatrix}, \mathbf{R}_y = \begin{bmatrix} \cos(q_i) & 0 & \sin(q_i) \\ 0 & 1 & 0 \\ -\sin(q_i) & 0 & \cos(q_i) \end{bmatrix},$$
$$\mathbf{R}_z = \begin{bmatrix} \cos(q_i) & -\sin(q_i) & 0 \\ \sin(q_i) & \cos(q_i) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{6}$$

If the joint i is prismatic, three translations in the directions X, Y, Z are possible, $\mathbf{Q} = \mathbf{I}$, and the translation component of the transformation is

$$ {}^{i-1}_{i}\mathbf{P} = \mathbf{t}_{i-1} + q_i \varepsilon_i \hat{\mathbf{u}}^{i}_{i-1}. \tag{7}$$

If the joint i is revolute, then $\varepsilon_i = 0$ in (7).

3.b. The generalization of the formalism, its use for structures with arbitrary axes (not necessarily parallel or perpendicular) is conducted by means of a homogeneous operator that rotates the axis i–1 to the axis i:

$$ {}^{i-1}_{i}\boldsymbol{T} = \begin{bmatrix} \boldsymbol{R}(\alpha, \hat{\boldsymbol{v}}) & {}^{i-1}_{i}\boldsymbol{P} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} {}^{i-1}_{i}\boldsymbol{Q} & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{8}$$

where $\mathbf{R}(\alpha, \hat{\mathbf{v}})$ is a Rodriguez-type rotation matrix, α is the angle of rotation and $\hat{\mathbf{v}}$ is the unit vector of the axis of rotation (also illustrated in Figure 2).

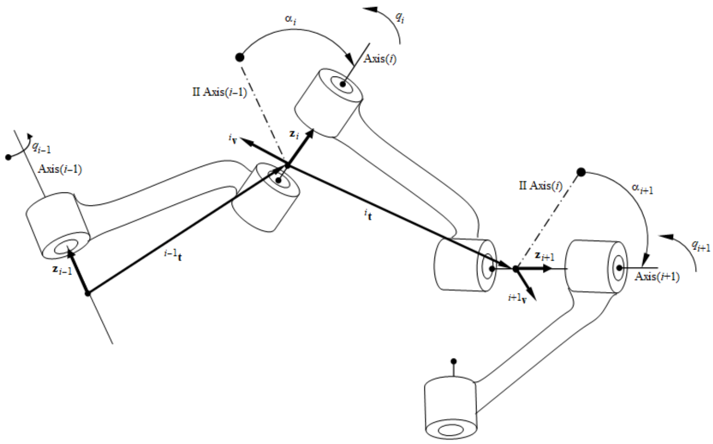

Figure 2

The general form of the formalism (adapted from [20])

This formalism, which actually carries out the transformation from the reference frame $\{i\}$ to the reference frame $\{i–1\}$, can be intuited as a journey of the reference frame $\{i–1\}$ to the reference frame $\{i\}$. In other words, it performs the translation of the reference frame $\{i–1\}$ to the origin of the reference frame $\{i\}$, followed by the rotation with the angle α around the unit vector $\hat{\mathbf{v}}$, and finally by the rotation or translation corresponding to the generalized variable q_i.

The subject of this paper is the transformation of the proposed formalism by modifying the mathematical formulae given above. This modification concerns the relations that use homogeneous matrices with new ones that manipulate quaternions. The steps 1 and 2 of the formalism will be kept, and the step 3 will be modified in terms of introducing new relations.

3 Definitions and Operations on Quaternions

Quaternions are hypercomplex numbers defined by Hamilton in [21] to describe the three-dimensional rotations of objects. The notation ***q*** will be used as follows for a quaternion, and it employs the following equivalent forms:

$$\begin{aligned}&\mathbf{q} = a + b\,i + c\,j + d\,k,\\ &\mathbf{q} = (a \quad b \quad c \quad d),\\ &\mathbf{q} = (a \quad \mathbf{v}),\end{aligned} \tag{9}$$

where a, b, c and d are real numbers, i, j and k are the imaginary elements (a generalization of the imaginary element i of a complex number), a is the real part of the quaternion, $\mathbf{v}$ is its vector part

$$\mathbf{v} = b\,i + c\,j + d\,k, \tag{10}$$

and the imaginary elements fulfill

$$\begin{aligned}&i^2 = j^2 = k^2 = -1,\\ &i \times j = k,\ j \times k = i,\ k \times i = j,\\ &j \times i = -k,\ k \times j = -i,\ i \times k = -j.\end{aligned} \tag{11}$$

Two operations are defined on the set of quaternions, namely addition and multiplication (non-commutative):

$$\begin{aligned}&\mathbf{q_1} + \mathbf{q_2} = (a_1 + a_2 \quad \mathbf{v_1} + \mathbf{v_2}),\\ &\mathbf{q} + \mathbf{0} = \mathbf{q},\\ &\mathbf{q_1 q_2} = (a_1 a_2 - \mathbf{v_1 v_2} \quad a_1\mathbf{v_2} + a_2\mathbf{v_1} + \mathbf{v_1} \times \mathbf{v_2}) \neq \mathbf{q}_2\mathbf{q}_1,\\ &\mathbf{q0} = \mathbf{0},\\ &\mathbf{q_1}(\mathbf{q_2 q}_3) = (\mathbf{q}_1\mathbf{q}_2)\mathbf{q}_3,\end{aligned} \tag{12}$$

where:

$$\begin{aligned}&\mathbf{0} = (0 \quad 0 \quad 0 \quad 0),\\ &\mathbf{1} = (1 \quad 0 \quad 0 \quad 0),\end{aligned} \tag{13}$$

which means that they belong to a non-commutative algebraic structure.

Rotations are described by rotation operators. For example, the rotation of the vector $\mathbf{p}$, its transformation into the vector ***q*** around the unit vector $\mathbf{v}$ with the angle α is expressed as

$$\mathbf{q} = \left(\cos\frac{\alpha}{2} \quad \mathbf{v}\sin\frac{\alpha}{2}\right)(0 \quad \mathbf{p})\left(\cos\frac{\alpha}{2} \quad \mathbf{v}\sin\frac{\alpha}{2}\right)^{*}. \tag{14}$$

The operations will be integrated in matrix computation in the next section.

4 Integration of Quaternions in the Translated Reference Frame Formalism

The proposed formalism preserves the first two steps that define the local reference frame (or coordinate systems) but modifies the mathematical apparatus. The transformation from the reference frame $\{k\}$ to the reference frame $\{i–1\}$ is carried out using

$$ {}^{i-1}_{k}\mathbf{T} = [{}^{i-1}_{i}\mathbf{r} \quad \mathbf{1}] \cdot \begin{bmatrix} {}^{i}_{k}\mathbf{T} & \mathbf{0} \\ \mathbf{0} & {}^{i-1}\mathbf{t} \end{bmatrix} \cdot \begin{bmatrix} {}^{i-1}_{i}\mathbf{r}^{*} \\ \mathbf{1} \end{bmatrix}, \tag{15} $$

where:

$$ \begin{aligned} &{}^{i-1}_{i}\mathbf{r} = (\cos(q_i/2) \quad \hat{\mathbf{u}}^{i}_{i-1}\sin(q_i/2)), \\ &{}^{i-1}_{i}\mathbf{t} = (0 \quad \mathbf{t}_{i-1} + \varepsilon_i q_i \hat{\mathbf{u}}^{i}_{i-1}) = (0 \quad t_{x,i-1} \quad t_{y,i-1} \quad t_{z,i-1}) + \varepsilon_i q_i (0 \quad \hat{\mathbf{u}}^{i}_{i-1}), \\ &\varepsilon_i = \begin{cases} 0 & \text{for} \quad \text{revolute joint}, \\ 1 & \text{for} \quad \text{prismatic joint}, \end{cases} \end{aligned} \tag{16} $$

q_i is the generalized variable at joint i, $\hat{\mathbf{u}}^{i}_{i-1}$ is the unit vector on which the movement of the joint i takes place, expressed in the reference frame $\{i–1\}$, $t_{x,i–1}$, $t_{y,i–1}$ and $t_{z,i–1}$ are the components of the vector linking the origin of the reference frame $\{i–1\}$ to the reference frame $\{i\}$ expressed in the reference frame $\{i–1\}$. Pointing out that

$$ \begin{aligned} &{}^{k-1}_{k}\mathbf{T} = [\mathbf{1} \quad \mathbf{1}] \cdot \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & {}^{k-1}\mathbf{t} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix}, \\ &{}^{k}_{k}\mathbf{T} = \mathbf{0}, \end{aligned} \tag{17} $$

with **1** and **0** defined in (13), the relationship (15) is iterative in i, it starts with the reference frame k=1…n and iterates downstream, at the limit, to the global reference frame $\{0\}$.

The relationship (15) can be reversed as follows:

$$ {}^{i}_{k}\mathbf{T} = [{}^{i-1}_{i}\mathbf{r}^{*} \quad \mathbf{1}] \cdot \begin{bmatrix} {}^{i-1}_{k}\mathbf{T} - {}^{i-1}\mathbf{t} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} {}^{i-1}_{i}\mathbf{r} \\ \mathbf{1} \end{bmatrix}, \tag{18} $$

with the interpretation that knowing the transformation from the reference frame $\{k\}$ to the reference frame $\{i–1\}$, the transformation to the reference frame $\{i\}$ can be calculated. Specifically, an upstream iteration is carried out.

If the orientations of the effector are intended to be computed, making use of the calculation of the unit vectors of its reference frame in the basic reference frame, the following formula can be used:

$$\mathbf{w}_0^n = \left(\prod_{i=1}^{n} {}_{i}^{i-1}\mathbf{r} \right) \cdot \mathbf{w}_n^n \cdot \left(\prod_{i=1}^{n} {}_{i}^{i-1}\mathbf{r} \right)^* , \tag{19}$$

where $\mathbf{w}_0^n$ is the quaternion corresponding to the unit vectors in the basic reference frame, and

$$\mathbf{w}_n^n = \begin{cases} (0 \quad 1 \quad 0 \quad 0) & \text{for} \quad \hat{\mathbf{n}}, \\ (0 \quad 0 \quad 1 \quad 0) & \text{for} \quad \hat{\mathbf{o}}, \\ (0 \quad 0 \quad 0 \quad 1) & \text{for} \quad \hat{\mathbf{a}}, \end{cases} \tag{20}$$

where $\hat{\mathbf{n}}$, $\hat{\mathbf{o}}$ and $\hat{\mathbf{a}}$ are highlighted in Figure 3. It is underlined that the cuaternion product is associative but not commutative.

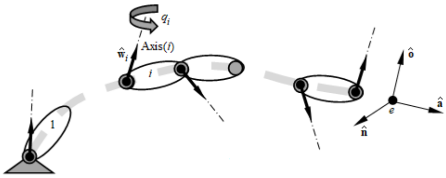

Figure 3
Jacobian matrix computation

5 Generalization to Arbitrary Axes

The translated reference frame formalism can be generalized to any type of structure as illustrated in Figure 2. The structures of serial robots have, generally, parallel or perpendicular to the axes of rotation or translation. This substantially simplifies the calculation of the rotation quaternion

$${}_{i}^{i-1}\mathbf{r} = (\cos(q_i/2) \quad \hat{\mathbf{u}}_{i-1}^{i} \sin(q_i/2)), \tag{21}$$

where $\hat{\mathbf{u}}_{i-1}^{i}$ is the unit vector of axis i, expressed in the reference frame $\{i{-}1\}$, with the possible expressions

$$\hat{\mathbf{u}}_{i-1}^{i} = \begin{cases} [1 \quad 0 \quad 0] & \text{if} \quad \text{Axis}(i) \parallel \hat{\mathbf{x}}^0, \\ [0 \quad 1 \quad 0] & \text{if} \quad \text{Axis}(i) \parallel \hat{\mathbf{y}}^0, \\ [0 \quad 0 \quad 1] & \text{if} \quad \text{Axis}(i) \parallel \hat{\mathbf{z}}^0. \end{cases} \tag{22}$$

But if a more general situation with two rotations is considered, the first overlaps the axis of the joint i–1 with the axis of the joint i, and the second rotates the joint with the generalized variable q_i in terms of

$$ {}^{i-1}_{i}\mathbf{r} = (\cos(\alpha_i/2) \quad \hat{\mathbf{v}}_{i-1}\sin(\alpha_i/2))(\cos(q_i/2) \quad \hat{\mathbf{u}}^{i}_{i-1}\sin(q_i/2)), \tag{23} $$

with:

$$ \begin{aligned} &\hat{\mathbf{v}}_{i-1} = \hat{\mathbf{z}}_i \times \hat{\mathbf{z}}_{i-1}, \\ &\alpha_i = \cos^{-1}(\hat{\mathbf{z}}_i \cdot \hat{\mathbf{z}}_{i-1}), \end{aligned} \tag{24} $$

where:

$$ \hat{\mathbf{z}}_{i-1} = [0 \quad 0 \quad 1]^T, \tag{25} $$

and $\hat{\mathbf{z}}_i$ is the unit vector of the axis of rotation, expressed in the reference frame $\{i–1\}$.

The proposed formalism and the steps presented in previous section along with the information in this section are organized systematically in the algorithm presented in Figure 4.

6 Jacobian Matrix Computation

The Jacobian matrix is the fundamental concept of the kinematic model

$$ \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{J}_0(\mathbf{q})\dot{\mathbf{q}}, \tag{26} $$

where $\mathbf{v}$ and $\boldsymbol{\omega}$ are the linear and the angular velocities, respectively, of the point of interest, respectively (at the limit of the origin of the effector the coordinate system), $\mathbf{J}_0 = \begin{bmatrix} {}^{v}\mathbf{J}_0 \\ {}^{\omega}\mathbf{J}_0 \end{bmatrix}$ is the Jacobian matrix expressed in the global reference frame, consisting of the two elements, the upper one referring to linear velocities, and the lower one corresponding to angular velocities, $\mathbf{q} = [q_1 \quad ... \quad q_n]^T$ is the vector of generalized variables, T indicates matrix transposition, and n is the number of degrees of freedom of the structure.

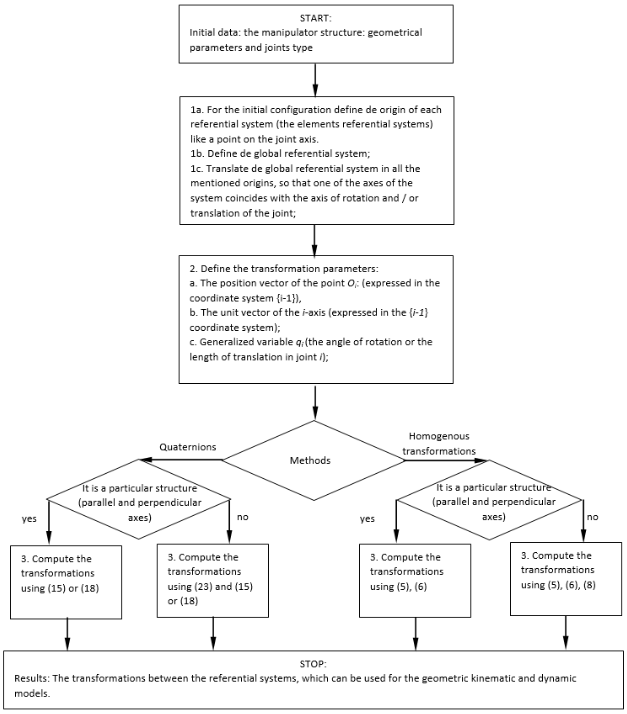

Figure 4

The block diagram of the proposed formalism

The computation of the Jacobian matrix is done as follows separately for the two elements mentioned above in relation with Figure 3. The upper element, corresponding to the linear velocities, is calculated using the partial derivatives of the components of the vector included in the quaternion ${}^{\mathbf{0}}_{\mathbf{e}}\mathbf{T} = (0 \quad {}^{0}p_{xe} \quad {}^{0}p_{ye} \quad {}^{0}p_{ze})$

$$ {}^{v}\mathbf{J}_0 = \begin{bmatrix} \frac{\partial p_{xe}}{\partial q_1} & \dots & \frac{\partial p_{xe}}{\partial q_n} \\ \frac{\partial p_{ye}}{\partial q_1} & \dots & \frac{\partial p_{ye}}{\partial q_n} \\ \frac{\partial p_{ze}}{\partial q_1} & \dots & \frac{\partial p_{ze}}{\partial q_n} \end{bmatrix}. \tag{27} $$

The lower element, corresponding to the angular velocities, is computed using the rotation axes

$$ {}^{\omega}\mathbf{J}_0 = [\varepsilon_1 \cdot \hat{\mathbf{w}}_0^1 \quad \dots \quad \varepsilon_n \cdot \hat{\mathbf{w}}_0^n], \tag{28} $$

where $\hat{\mathbf{w}}_0^i$ is the unit vector of the $\{i\}$ axis expressed in the base reference frame, and

$$\varepsilon_i = \begin{cases} 1 & \text{for} \quad \text{revolute joint}, \\ 0 & \text{for} \quad \text{prismatic joint}, \end{cases}$$
$$(0 \quad \hat{\mathbf{w}}_0^k) = \left(\prod_{i=1}^{k} {}_i^{i-1}\mathbf{r}\right) \cdot (0 \quad \hat{\mathbf{w}}_k^k) \cdot \left(\prod_{i=1}^{k} {}_i^{i-1}\mathbf{r}\right)^* . \tag{29}$$

Once again it is underlined that the cuaternion product is associative but not commutative.

7 Examples

Two simple preparatory cases are first presented. They refer to planar structures such as Rotation Rotation (RR) and Rotation Translation (RT).

Since the principles from which the two alternatives of the proposed formalism start (included in steps 1 and 2 of the algorithm) are identical, this allows the idea of carrying out their comparison. The solution to the direct geometry problem obtained by both approaches will be presented as follows.

Figure 5 illustrates an RR type structure, which according to the first two steps of the presented formalism contains the four coordinate systems {0 = 1}, {2}, {e} attached to the base, the elements and the gripper, respectively. There are also illustrated here the components of the translation vector that connects the mentioned coordinate systems, l_1 and l_2. The axes of the two joints are perpendicular to the plane of the structure $\hat{z} \equiv \hat{u}^1 \equiv \hat{u}^2$. Figure 5 also shows the two generalized angular variables q_1 and q_2. The use of the homogeneous transformations for the RR structure is synthesized in Table 1.

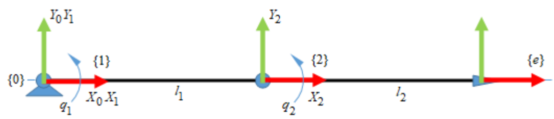

Figure 5
The RR structure

The approach proposed in this paper modifies the mathematical apparatus, i.e., it uses quaternions instead of homogeneous transformations in step 3. Table 2 gives the application of the transformation equations (15) to the RR structures. The transformations as well as the elements that are included in these transformations are defined. The quaternions are actually computed for each transformation.

Table 1

Homogeneous transformations for the RR structure

No.	The parameters	The transformations
1	${}^0_1\mathbf{Q} = \mathbf{R_z}(q_1)$ ${}^0_1\mathbf{P} = [0 \quad 0 \quad 0]$	${}^0_1\mathbf{T} = \begin{bmatrix} c_{q_1} & -s_{q_1} & 0 & 0 \\ s_{q_1} & c_{q_1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
2	${}^1_2\mathbf{Q} = \mathbf{R}_z(q_2)$ ${}^1_2\mathbf{P} = [l_1 \quad 0 \quad 0]$	${}^1_2\mathbf{T} = \begin{bmatrix} c_{q_2} & -s_{q_2} & 0 & l_1 \\ s_{q_2} & c_{q_2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
e	${}^2_e\mathbf{Q} = \mathbf{I}$ ${}^6_e\mathbf{P} = [l_2 \quad 0 \quad 0]$	${}^6_e\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Table 2

Use of quaternions for the RR structure

The transformation	**r** quaternions	**t** quaternions
Transformation e-0		
${}^0_e\mathbf{T} = [{}^0_1\mathbf{r} \quad \mathbf{1}] \cdot \begin{bmatrix} {}^1_e\mathbf{T} & \mathbf{0} \\ \mathbf{0} & {}^0\mathbf{t} \end{bmatrix} \cdot \begin{bmatrix} {}^0_1\mathbf{r}^* \\ \mathbf{1} \end{bmatrix}$	${}^0_1\mathbf{r} = \left(\cos\left(\frac{q_1}{2}\right) \quad \hat{\mathbf{u}}_1^1 \sin\left(\frac{q_1}{2}\right) \right)$ $\hat{\mathbf{u}}_1^1 = [0 \quad 0 \quad 1]$	${}^0\mathbf{t} = \mathbf{0}$
Transformation e-1		
${}^1_e\mathbf{T} = [{}^1_2\mathbf{r} \quad \mathbf{1}] \cdot \begin{bmatrix} {}^2_e\mathbf{T} & \mathbf{0} \\ \mathbf{0} & {}^1\mathbf{t} \end{bmatrix} \cdot \begin{bmatrix} {}^1_2\mathbf{r}^* \\ \mathbf{1} \end{bmatrix}$	${}^0_1\mathbf{r} = \left(\cos\left(\frac{q_2}{2}\right) \quad \hat{\mathbf{u}}_2^2 \sin\left(\frac{q_2}{2}\right) \right)$ $\hat{\mathbf{u}}_2^2 = [0 \quad 0 \quad 1]$	${}^1\mathbf{t} = (0 \quad l_1 \quad 0 \quad 0)$
Transformation e-2		
${}^2_e\mathbf{T} = [\mathbf{1} \quad \mathbf{1}] \cdot \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & {}^2\mathbf{t} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix}$		${}^2\mathbf{t} = (0 \quad l_2 \quad 0 \quad 0)$
Transformation 2-0		
${}^0_2\mathbf{T} = [{}^0_1\mathbf{r} \quad \mathbf{1}] \cdot \begin{bmatrix} {}^1_2\mathbf{T} & \mathbf{0} \\ \mathbf{0} & {}^0\mathbf{t} \end{bmatrix} \cdot \begin{bmatrix} {}^0_1\mathbf{r}^* \\ \mathbf{1} \end{bmatrix}$	${}^0_1\mathbf{r} = \left(\cos\left(\frac{q_1}{2}\right) \quad \hat{\mathbf{u}}_1 \sin\left(\frac{q_1}{2}\right) \right)$ $\hat{\mathbf{u}}_1 = [0 \quad 0 \quad 1]$	${}^0\mathbf{t} = \mathbf{0}$
Transformation 2-1		
${}^1_2\mathbf{T} = [\mathbf{1} \quad \mathbf{1}] \cdot \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & {}^1\mathbf{t} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix}$		${}^1\mathbf{t} = (0 \quad l_1 \quad 0 \quad 0)$

Figure 6 illustrates a RT type structure, which according to the first two steps of the presented formalism, contains the four coordinate systems {0 = 1}, {2}, {e} attached to the base, the elements and the gripper, respectively. In addition, Figure 6 highlights the components of the translation vector that connects the mentioned coordinate systems, l_1 and l_2, the axes of the two joints $\hat{\mathbf{z}} = \hat{\mathbf{u}}^1$ and $\hat{\mathbf{x}} = \hat{\mathbf{u}}^2$, and the two generalized angular variables q_1 and q_2. The use of the homogeneous transformations for the RT structure is synthesized in Table 3.

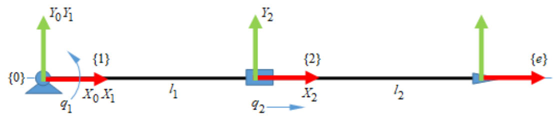

Figure 6
The RT structure

Table 3
Use of quaternions for the RT structure

No.	The parameters	The transformations
1	${}^0_1\mathbf{Q} = \mathbf{R}_z(q_1)$ ${}^0_1\mathbf{P} = [0 \quad 0 \quad 0]$	${}^0_1\mathbf{T} = \begin{bmatrix} c_{q_1} & -s_{q_1} & 0 & 0 \\ s_{q_1} & c_{q_1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
2	${}^1_2\mathbf{Q} = \mathbf{I}$ ${}^1_2\mathbf{P} = [l_1 + q_2 \quad 0 \quad 0]$	${}^1_2\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & l_1 + q_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
e	${}^2_e\mathbf{Q} = \mathbf{I}$ ${}^6_e\mathbf{P} = [l_2 \quad 0 \quad 0]$	${}^6_e\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Table 4 gives the application of the transformation equations (15) to the RT structures. The transformations as well as the elements that are included in these transformations are defined. The quaternions are computed, as in the previous example, for each transformation.

Table 4
Use of quaternions for the RR structure

The transformation	**r** quaternions	**t** quaternions
Transformation e-0		
${}^0_e\mathbf{T} = [{}^0_1\mathbf{r} \quad \mathbf{1}] \cdot \begin{bmatrix} {}^1_e\mathbf{T} & \mathbf{0} \\ \mathbf{0} & {}^0\mathbf{t} \end{bmatrix} \cdot \begin{bmatrix} {}^0_1\mathbf{r}^* \\ \mathbf{1} \end{bmatrix}$	${}^0_1\mathbf{r} = \left(\cos\left(\frac{q_1}{2}\right) \quad \hat{\mathbf{u}}^1_1 \sin\left(\frac{q_1}{2}\right)\right)$ $\hat{\mathbf{u}}^2_2 = [0 \quad 0 \quad 1]$	${}^0\mathbf{t} = \mathbf{0}$

Transformation e-1		
${}^{1}_{e}\mathbf{T} = [{}^{1}_{2}\mathbf{r} \quad \mathbf{1}] \cdot \begin{bmatrix} {}^{2}_{e}\mathbf{T} & \mathbf{0} \\ \mathbf{0} & {}^{1}\mathbf{t} \end{bmatrix} \cdot \begin{bmatrix} {}^{1}_{2}\mathbf{r}^{*} \\ \mathbf{1} \end{bmatrix}$	${}^{0}_{1}\mathbf{r} = (1 \quad 0 \quad 0 \quad 0) = \mathbf{1}$	${}^{1}\mathbf{t} = (0 \quad q_2 + l_1 \quad 0 \quad 0)$
Transformation e-2		
${}^{2}_{e}\mathbf{T} = [\mathbf{1} \quad \mathbf{1}] \cdot \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & {}^{2}\mathbf{t} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix}$		${}^{2}\mathbf{t} = (0 \quad l_2 \quad 0 \quad 0)$
Transformation 2-0		
${}^{0}_{2}\mathbf{T} = [{}^{0}_{1}\mathbf{r} \quad \mathbf{1}] \cdot \begin{bmatrix} {}^{1}_{2}\mathbf{T} & \mathbf{0} \\ \mathbf{0} & {}^{0}\mathbf{t} \end{bmatrix} \cdot \begin{bmatrix} {}^{0}_{1}\mathbf{r}^{*} \\ \mathbf{1} \end{bmatrix}$	${}^{0}_{1}\mathbf{r} = \left(\cos\left(\frac{q_1}{2}\right) \quad \hat{\mathbf{u}}_1 \sin\left(\frac{q_1}{2}\right)\right)$ $\hat{\mathbf{u}}_1 = [0 \quad 0 \quad 1]$	${}^{0}\mathbf{t} = \mathbf{0}$
Transformation 2-1		
${}^{1}_{2}\mathbf{T} = [\mathbf{1} \quad \mathbf{1}] \cdot \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & {}^{2}\mathbf{t} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix}$		${}^{2}\mathbf{t} = (0 \quad l_2 \quad 0 \quad 0)$

The translated reference frame formalism is described in [20] along with its comparison to the DH formalism for a PUMA type robot structure. This time, considering the same structure illustrated in Figure 7, a comparison will be carried out as follows between the two mathematical formulations of the same formalism, namely that proposed in [20] and the formulation proposed in this paper. Therefore, an answer is given to the question of comparing the mathematical apparatus of homogeneous transformations with that of quaternions.

Table 5 given in [22] describes the parameters and homogeneous transformations in accordance with (2). The second column specifies the elements of the homogeneous transformation, i.e., the rotation matrices and the translation vectors. Each row in the table refers to the transformation from the reference frame $\{i\}$ to the reference frame $\{i–1\}$.

Table 6 given in [22] gives details on the transformations defined using quaternions. The rows in the table offer, similar to the rows in Tables 2 and 4, the transformation applied, and the quaternions used by it as well.

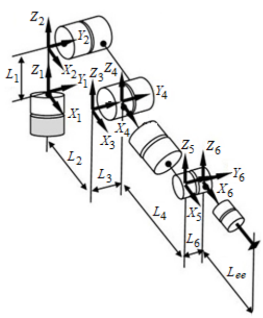

Figure 7
The RT structure

8 Simulation Results

The Matlab program used in the automation of the formalism proposed is freely available at https://autocarsim.com/use-of-quaternions/. The program refers to the PUMA type robot structure. The use of the program is described in the dedicated script.

The transformations specified in Table 6 are implemented in a Matlab program, which allows determining the position of each element, validating the theoretical results, and obtaining the graphical representation of the structure configuration. Table 7 given in [22] describes a sample of the results of several simulations conducted for different angular variables.

The figures included in Table 7, describe the reference frame of the base and that of the effector (the CAD convention was used to color the axes), the elements of the structure and the joints (circles). The values of the angular variables are specified in the title of each figure.

Conclusions

This paper further developed the authors' previous approach, regarding a new formalism for solving the direct geometry problem of manipulators suggested in [20]. The advantages of the proposed formalism are the simplicity to define the reference frames of the elements, and the simplicity of defining the mathematical transformations. More precisely, it has been shown in [20] that a set of six generic transformations is available to be easily customized to any transformation used by the formalism. The paper brings novelties in the mathematical apparatus of the formalism, i.e., it suggests an alternative to replace the computation of homogeneous transformations with that of quaternions.

The quaternions are employed in solving the direct geometry problem and the Jacobian matrix calculation. The proposed formalism was exemplified and compared in three structures, namely RR, RT and PUMA type. The examples also include the conversion of algorithms into computer programs, which allowed the simulation of solving the direct geometry problem.

Future research will be focused on the application of authors' formalism to the design of control systems for robots in real-world applications. Such applications are popular in several fields including path planning [23] [24], electric vehicles [25] [26], haptic interfaces [27], manufacturing processes [28-30]. Several optimization algorithms can be used in these applications for performance improvement and also reducing the heuristics in the design, as, for example, cellular genetic algorithms [31], tabu search based on quantum computing [32], Bacterial Foraging Optimization Algorithms [33] [34], Clonal Selection Algorithms [34], Grey Wolf Optimizers [35], Particle Swarm Optimization Algorithms [36], Slime Mold Algorithms [29], classical optimization algorithms

[37], multi-parametric quadratic programming [38], and Metaheuristic Algorithms with parameter adaptation [39].

Acknowledgement

This work was supported by a grant of the Romanian Ministry of Education and Research, CNCS - UEFISCDI, project number PN-III-P4-ID-PCE-2020-0269, within PNCDI III, the Széchenyi István University, Győr, Hungary, and a grant of the Romanian Ministry of Research, Innovation and Digitization, CNCS/CCCDI – UEFISCDI, project number PN-III-P2-2.1-PED-2019-4366 (431PED) within PNCDI III.

References

[1] G. Booch, R.A. Maksimchuk, M. W. Engle, B. J. Young, J. Conallen, K. A. Houston: Object-Oriented Analysis and Design with Applications, 3rd Edition. Upper Saddle River, NJ: Addison-Wesley Professional, 2007

[2] Q. Zhang, Y.-F. Zhang, S.-Y. Qin: The object oriented analysis and modeling for obstacle avoidance of a behavior-based robot, in Proceedings of 2007 IEEE International Conference on Systems, Man and Cybernetics, Montreal, QC, Canada, 2007, pp. 1545-1550

[3] M. Rickert, A. Gaschler: Robotics library: An object-oriented approach to robot applications, in Proceedings of 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vancouver, BC, Canada, 2017, pp. 733-740

[4] P. Ramaraj: Robots that help humans build better mental models of robots, in Proceedings of 2021 ACM/IEEE International Conference on Human-Robot Interaction, Virtual, 2021, pp. 595-597

[5] S. A. Bortoff: Object-oriented modeling and control of delta robots, in Proceedings of 2018 IEEE Conference on Control Technology and Applications, Copenhagen, Denmark, 2018, pp. 251-258

[6] P. Fritzon: Principles of Object Oriented Modeling and Simulation with Modelica 3.3: A Cyber-Physical Approach, 2nd Edition. Piscataway, NJ: IEEE Press-Wiley, 2015

[7] C. Pozna: Modelarea robotilor cu post fix (in Romanian). Bucharest: MatrixRom, 2013

[8] J. Denavit, R. S. Hartenberg: A kinematic notation for lower-pair mechanisms based on matrices, Journal of Applied Mechanics, Vol. 22, No. 2, 1955, pp. 215-221

[9] R. S. Hartenberg, J. Denavit: Kinematic Synthesis of Linkages. New York, NY: McGraw-Hill, 1964

[10] R. P. Paul: Robot Manipulators: Mathematics, Programming, and Control. Cambridge, MA, London: The MIT Press, 1981

[11] W. Khalil, J. F. Kleinfinger: A new geometrical notation for open and closed loop robots, in Proceedings of 1986 IEEE International Conference on Robotics and Automation, San Francisco, CA, USA, 1986, pp. 1174-1180

[12] J. J. Craig: Introduction to Robotics: Mechanics and Control, 3rd Edition. Upper Saddle River, NJ: Pearson Education, 2005

[13] P. Corke: Robotics, Vision and Control. Berlin, Heidelberg: Springer-Verlag, 2011

[14] RoboAnalyzer: 3D Model Based Robotics Learning Software, posted at http://www.roboanalyzer.com/ and accessed at July 12, 2021

[15] G. Gogu, P. Coiffet: Representation du movement des corps solids (in French). Paris: Hermes, 1996

[16] G. Gogu, P. Coiffet, A. Barraco: Representation des deplacements des robots (in French). Paris: Hermes, 1997

[17] R. Murray, Z. Li, & S. Sastry: A Mathematical Introduction to Robotic Manipulation. CRC Press, 1994

[18] J. J. Craig: Introduction to Robotics: Mechanics and Control. Pearson Education, 2004

[19] B. Siciliano, L. Sciavicco: Robotics: Modelling, Planning and Control. Springer, 2008

[20] C. Pozna, R.-E. Precup: On a translated frame-based approach to geometric modeling of robots, Robotics and Autonomous Systems, Vol. 91, 2017, pp. 49-58

[21] W. R. Hamilton: Elements of Quaternions. Cambridge: Cambridge University Press, 1899

[22] C. Pozna, R.-E. Precup: Appendix of the paper C. Pozna and R.-E. Precup, "On the Use of Quaternions in the Translated Reference Frame Formalism", Acta Polytechnica Hungarica, 2023, posted at https://www.aut.upt.ro/~rprecup/Appendix_paper_APH_2023_Pozna_Precup.pdf and accessed at November 12, 2022

[23] G. Klančar, S. Blažič: Optimal constant acceleration motion primitives, IEEE Transactions on Vehicular Technology, Vol. 68, No. 9, 2019, pp. 8502-8511

[24] J. Vaščák, J. Hvizdoš, M. Puheim: Agent-based cloud computing systems for traffic management, in Proceedings of 2016 International Conference on Intelligent Networking and Collaborative Systems, Ostrava, Czech Republic, 2016, pp. 73-79

[25] Z. C. Johanyák: A modified particle swarm optimization algorithm for the optimization of a fuzzy classification subsystem in a series hybrid electric

vehicle, Technicki Vjesnik - Technical Gazette, Vol. 24, No. 2, 2017, pp. 295-301

[26] A. Lucchini, S. Formentin, M. Corno, D. Piga, S. M. Savaresi: Torque vectoring for high-performance electric vehicles: a data-driven MPC approach, IEEE Control Systems Letters, Vol. 4, No. 3, 2020, pp. 725-730

[27] N. Ando, P. Korondi, H. Hashimoto: Networked telemicromanipulation systems "Haptic Loupe", IEEE Transactions on Industrial Electronics, Vol. 51, No. 6, 2004, pp. 1259-1271

[28] R. E. Haber, J. R. Alique, A. Alique, J. Hernández, R. Uribe-Etxebarria: Embedded fuzzy-control system for machining processes: Results of a case study, Computers in Industry, Vol. 50, No. 3, 2003, pp. 353-366

[29] R.-E. Precup, R.-C. David, R.-C. Roman, E. M. Petriu, A.-I. Szedlak-Stinean: Slime mould algorithm-based tuning of cost-effective fuzzy controllers for servo systems, International Journal of Computational Intelligence Systems, Vol. 14, No. 1, 2021, pp. 1042-1052

[30] V. M. Nejković, M. S. Milićević, G. Janaćković, M. Grozdanović: Application of fuzzy analytic hierarchy process to inductive steel tube welding, Romanian Journal of Information Science and Technology, Vol. 25, No. 1, 2022, pp. 3-19

[31] E. Osaba, J. Del Ser, A. D. Martinez, J. L. Lobo, F. Herrera: AT-MFCGA: An adaptive transfer-guided multifactorial cellular genetic algorithm for evolutionary multitasking, Information Sciences, Vol. 570, 2021, pp. 577-598

[32] E. Osaba, E. Villar-Rodriguez, I. Oregi, A. Moreno-Fernandez-de-Leceta: Hybrid quantum computing-tabu search algorithm for partitioning problems: preliminary study on the traveling salesman problem, in Proceedings of 2021 IEEE Congress on Evolutionary Computation, Kraków, Poland, 2021, pp. 351-358

[33] S. I. Bejinariu, H. Costin: Medical image registration by means of a bio-inspired optimization strategy, Computer Science Journal of Moldova, Vol. 20, No. 2, 2012, pp. 178-202

[34] S. I. Bejinariu, H. Costin, F. Rotaru, C. Niţă, R. Luca, C. Lazăr: Parallel processing and bio-inspired computing for biomedical image registration, Computer Science Journal of Moldova, Vol. 22, No. 2, 2014, pp. 253-277

[35] I. A. Zamfirache, R.-E. Precup, R.-C. Roman, E. M. Petriu: Neural network-based control using actor-critic reinforcement learning and grey wolf optimizer with experimental servo system validation, Expert Systems with Applications, Vol. 225, 2023, paper 120112

[36] D. Sánchez, P. Melin, O. Castillo: Comparison of particle swarm optimization variants with fuzzy dynamic parameter adaptation for modular

granular neural networks for human recognition, Journal of Intelligent and Fuzzy Systems, Vol. 38, No. 3, 2020, pp. 3229-3252

[37] D. Singh, A. Shukla: Manifold optimization with MMSE hybrid precoder for Mm-Wave massive MIMO communication, Romanian Journal of Information Science and Technology, Vol. 25, No. 1, 2022, pp. 36-46

[38] Z. Preitl, R.-E. Precup, J. K. Tar, M. Takács: Use of multi-parametric quadratic programming in fuzzy control systems, Acta Polytechnica Hungarica, Vol. 3, No. 3, 2006, pp. 29-43

[39] O. Castillo, P. Melin, E. Ontiveros, C. Peraza, P. Ochoa, F. Valdez, J. Soria: A high-speed interval type 2 fuzzy system approach for dynamic parameter adaptation in metaheuristics, Engineering Applications of Artificial Intelligence, Vol. 85, 2019, pp. 666-680