9th International Workshop on Research and Education in Mechatronics
September 18th-19th 2008, Bergamo, Italy

A NON-ABELIAN GROUP ALGEBRA FOR KINEMATIC COORDINATE TRANSFORMATION

Alessandro Tasora[1]

[1]*Università degli Studi di Parma, Dipartimento di Ingegneria Industriale, Parma, Italy*

Abstract: This paper proposes an algebra $\mathcal{T}(\mathbb{T}, \succ)$ which can be used to express kinematic transformations in chains of frames that move in threedimensional space. The algebraic structure of $\mathcal{T}$ will be discussed end the most relevant properties will be presented. This algebra can be translated into a set of algorithms that fit well into a compact formalism, by exploiting the operator-overloading feature of modern object-oriented programming languages. Implementation and application are discussed by means of examples.

Key words: Kinematics, algebra, coordinates, robotics

I INTRODUCTION

Coordinate transformations are extensively used in theoretical, applied and computational mechanics. In the simpliest case, these transformations arise from the need of computing the absolute position of points given their relative position respect to rigid frames which move in three-dimensional space. In this case, the transformation can be expressed as an affine map using matrix algebra, quaternion algebra or similar formalisms.

As an extension of the abovementioned problem, lets consider the case of a chain of moving frames, where the position of each frame is known respect to the previous one in the chain: the absolute position of the end of the chain can be expressed as a sequence of affine transformations based on the relative coordinates. In the robotic field, a compact way to represent this kind of nested coordinate transformations is the Denavit-Hartenberg approach [4], where 4x4 matrices are used to express rotations and translations with a single matrix multiplication on four-dimensional vectors; these matrices can be concatenated to express sequences of coordinate transformations as in robotic arms.

In this work we extend the problem even to speeds and accelerations. Lets consider a chain of frames where not only the relative position, but also the relative speed and relative acceleration of each frame is known respect to the previous one in the chain: a formalism can be developed to find the absolute position, speed and acceleration of the end of the chain.

To this end we define a $\succ$ operator such that a sequence of $\succ$ operations corresponds to a chain of coordinate transformations. For instance, say $\chi_{i,(j)}$ means the position, speed and acceleration (either linear and angular) of the coordinate i respect to coordinate j: one can write sequences of transformations as in the following example:

$$\chi_{3,(0)} = \chi_{3,(2)} \succ \chi_{2,(1)} \succ \chi_{1,(0)}. \qquad (1)$$

This correspond to the case of Fig.1, where for example one can use the previous transformation to compute $r_{3,(0)}$, i.e the absolute position of the origin of 3 respect to the absolute coordinate system 0, since

$$\chi_{3,(0)} = \{r_{3,(0)}, q_{3,(0)}, \dot{r}_{3,(0)}, \omega_{3,(0)}, \ddot{r}_{3,(0)}, \alpha_{3,(0)}\}.$$

A similar result could be obtained with linear algebra, for example using the Denavit-Hartenberg formalism, [5], however we remark that our approach is more general and also encompasses speed and acceleration transformations in a single operation (for instance, the absolute speed of 3 is also contained in $\chi_{3,(0)}$, and depends on angular velocities and speeds of all other systems in the chain).

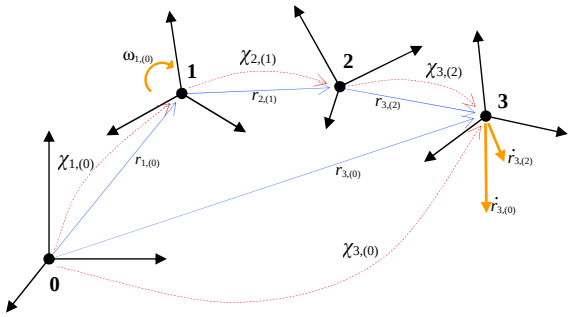

Figure 1: Example of chained transformation

II THE $\mathcal{T}(\mathbb{T}, \succ)$ ALGEBRA

We introduce a magma $\mathcal{T}(\mathbb{T}, \succ)$ whose non-abelian operator $\succ$ acts over a 19-dimensional Banach space

$\mathbb{T} = \{\mathbb{R}^3 \times \mathcal{P} \times \mathbb{R}^{12}\}$. The $\succ$ operator has arity $\bar{\alpha} = 2$.

Such algebra shows its usefulness in managing coordinate transformations: each element $\chi \in \mathbb{T}$ can represent a coordinate system in three-dimensional space, including position, rotation informations, as well as speed and acceleration informations.

In detail, position is expressed by $\mathbf{r} \in \mathbb{R}^3$, rotation is expressed by an unitary quaternion $\mathbf{q} \in \mathsf{S}^3$. Note that, alternatively, rotations could be parametrized also with a set of three angles, however it is well known that this could cause problems of map singularities because $\mathbb{R}^3$ is not omeomorphic to $\mathbb{SO}(3)$. The speed of the origin of the frame is denoted with $\dot{\mathbf{r}} \in \mathbb{R}^3$, its angular speed is denoted with $\omega \in \mathbb{R}^3$ (expressed in the base of the moving frame). Similarly we denote acceleration with $\ddot{\mathbf{r}}$ and angular acceleration with α, the latter being expressed in the local base of the moving frame. Thus, we define the vector $\chi \in \mathbb{T}$ as:

$$\chi = \{\mathbf{r}, \mathbf{q}, \dot{\mathbf{r}}, \omega, \ddot{\mathbf{r}}, \alpha\}, \mathbf{r} \in \mathbb{R}^3,\ \mathbf{q} \in \mathsf{S}^3,\ \dot{\mathbf{r}}, \omega, \ddot{\mathbf{r}}, \alpha \in \mathbb{R}^3.$$

Note that $\mathbb{T} = \{\mathbb{R}^3 \times \mathcal{P} \times \mathbb{R}^{12}\}$ is a topological space [7], a double covering of $\mathbb{T} = \{\mathbb{R}^3 \times \mathbb{SO}(3) \times \mathbb{R}^{12}\}$ because of the surjective local homeomorphism $r : \mathsf{S}^3 \to \mathbb{SO}(3, \mathbb{R})$ where two quaternions can represent the same rotation.

Also, $\mathbb{T}$ is a non-trivial bundle with fiber S^1 because of the π Hopf fibration $\mathsf{S}^1 \hookrightarrow \mathsf{S}^3 \xrightarrow{\pi} \mathsf{S}^2$.

Assuming that position, rotation, speed and acceleration of a frame a in χ_a are expressed relative to another frame b, we will use the notation $\chi_{a,(b)}$. Also, we will denote the frame b as the *parent frame* of a or, similarly, a as the *child frame* of b. The definition of the $\succ$ operation stems from the requirement that

$$\chi_{a,(c)} = \chi_{a,(b)} \succ \chi_{b,(c)} \quad \rightarrow$$
$$\begin{Bmatrix} \mathbf{r}_{a,(c)} \\ \mathbf{q}_{a,(c)} \\ \dot{\mathbf{r}}_{a,(c)} \\ \omega_{a,(c)} \\ \ddot{\mathbf{r}}_{a,(c)} \\ \alpha_{a,(c)} \end{Bmatrix} = \begin{Bmatrix} \mathbf{r}_{a,(b)} \\ \mathbf{q}_{a,(b)} \\ \dot{\mathbf{r}}_{a,(b)} \\ \omega_{a,(b)} \\ \ddot{\mathbf{r}}_{a,(b)} \\ \alpha_{a,(b)} \end{Bmatrix} \succ \begin{Bmatrix} \mathbf{r}_{b,(c)} \\ \mathbf{q}_{b,(c)} \\ \dot{\mathbf{r}}_{b,(c)} \\ \omega_{b,(c)} \\ \ddot{\mathbf{r}}_{b,(c)} \\ \alpha_{b,(c)} \end{Bmatrix} \quad (2)$$

In the following we develop the expressions for the terms in 2.

A. Position and rotation

Since $||\mathbf{q}_a||||\mathbf{q}_b|| = ||\mathbf{q}_a\mathbf{q}_b||$, the following endomorphism is $||\cdot||_2$-norm preserving and rotates a quaternion $\mathbf{q}_p \in \mathbb{Q}$ by means of an unimodular quaternion $\mathbf{q}_r \in \mathsf{S}^3$ and its conjugate $\mathbf{q}_r^*$:

$$\mathbf{q}_p^o = \mathbf{q}_r\mathbf{q}_p\mathbf{q}_r^*. \quad (3)$$

It is well known that 3 can be used to rotate points in space, with rotation expressed by $\mathbf{q}_r$ as Euler parameters, if $\mathbf{q}_p$ is a pure-quaternion with imaginary part as the cartesian coordinates of a point: $\mathbf{q}_p = \mathbf{q}_\Im(\mathbf{p}) = \{0, p_x, p_y, p_z\}$. Therefore, the affine transformation of the point $\mathbf{r}_{a,(b)}$ after rotation $\mathbf{q}_{b,(c)}$ and translation $\mathbf{r}_{b,(c)}$ can be expressed as:

$$\mathbf{q}_\Im(\mathbf{r}_{a,(c)}) = \mathbf{q}_\Im(\mathbf{r}_{b,(c)}) + \mathbf{q}_{b,(c)}\mathbf{q}_\Im(\mathbf{r}_{a,(b)})\mathbf{q}_{b,(c)}^*. \quad (4)$$

Alternatively, 4 can be expressed with linear algebra as

$$\mathbf{r}_{a,(c)} = \mathbf{r}_{b,(c)} + [A(\mathbf{q}_{b,(c)})]\mathbf{r}_{a,(b)}. \quad (5)$$

where we introduced the rotation matrix $[A(\mathbf{q}_{b,(c)})]$ function of the quaternion $\mathbf{q}_{b,(c)}$ as detailed in [cite].

The term $\mathbf{q}_{a,(c)}$, representing the rotation of the reference a respect to the reference c, can be easily obtained with a single quaternion product:

$$\mathbf{q}_{a,(c)} = \mathbf{q}_{b,(c)}\mathbf{q}_{a,(b)} \quad (6)$$

as can be seen applying the affine map 4 twice, for transforming a point p from reference a to reference b and from reference b to reference c:

$$\begin{aligned}\mathbf{q}_\Im(\mathbf{r}_{p,(c)}) =& \mathbf{q}_\Im(\mathbf{r}_{b,(c)}) + \mathbf{q}_{b,(c)}\left(\mathbf{q}_\Im(\mathbf{r}_{a,(b)}) + \right.\\ &\left. + \mathbf{q}_{a,(b)}\mathbf{q}_\Im(\mathbf{r}_{p,(a)})\mathbf{q}_{a,(b)}^*\right)\mathbf{q}_{b,(c)}^* \\ =& \mathbf{q}_\Im(\mathbf{r}_{b,(c)}) + \mathbf{q}_{b,(c)}\mathbf{q}_\Im(\mathbf{r}_{a,(b)})\mathbf{q}_{b,(c)}^* + \\ &+ \mathbf{q}_{b,(c)}\mathbf{q}_{a,(b)}\mathbf{q}_\Im(\mathbf{r}_{p,(a)})\mathbf{q}_{a,(b)}^*\mathbf{q}_{b,(c)}^* \\ =& \mathbf{q}_\Im(\mathbf{r}_{a,(c)}) + \mathbf{q}_{a,(c)}\mathbf{q}_\Im(\mathbf{r}_{p,(a)})\mathbf{q}_{a,(c)}^*.\end{aligned}$$

We recall that quaternion products are associative but not commutative [1].

B. Speeds

Speed terms $\dot{\mathbf{r}}_{a,(c)}$ and $\omega_{a,(c)}$ can be obtained from symbolic differentiation of Eq.6 and Eq.6. By applying the chain rule to the differentiation of the affine map of Eq.4:

$$\begin{aligned}\dot{\mathbf{q}}_\Im(\mathbf{r}_{a,(c)}) =& \dot{\mathbf{q}}_\Im(\mathbf{r}_{b,(c)}) + \dot{\mathbf{q}}_{b,(c)}\mathbf{q}_\Im(\mathbf{r}_{a,(b)})\mathbf{q}_{b,(c)}^* + \\ &+ \mathbf{q}_{b,(c)}\dot{\mathbf{q}}_\Im(\mathbf{r}_{a,(b)})\mathbf{q}_{b,(c)}^* + \\ &+ \mathbf{q}_{b,(c)}\mathbf{q}_\Im(\mathbf{r}_{a,(b)})\dot{\mathbf{q}}_{b,(c)}^* \\ \dot{\mathbf{q}}_\Im(\mathbf{r}_{a,(c)}) =& \dot{\mathbf{q}}_\Im(\mathbf{r}_{b,(c)}) + 2\dot{\mathbf{q}}_{b,(c)}\mathbf{q}_\Im(\mathbf{r}_{a,(b)})\mathbf{q}_{b,(c)}^* + \\ &+ \mathbf{q}_{b,(c)}\dot{\mathbf{q}}_\Im(\mathbf{r}_{a,(b)})\mathbf{q}_{b,(c)}^* \\ \mathbf{q}_\Im(\dot{\mathbf{r}}_{a,(c)}) =& \mathbf{q}_\Im(\dot{\mathbf{r}}_{b,(c)}) + 2\dot{\mathbf{q}}_{b,(c)}\mathbf{q}_\Im(\mathbf{r}_{a,(b)})\mathbf{q}_{b,(c)}^* + \\ &+ \mathbf{q}_{b,(c)}\mathbf{q}_\Im(\dot{\mathbf{r}}_{a,(b)})\mathbf{q}_{b,(c)}^*. \quad (7)\end{aligned}$$

[1] By Hurwitz theorem, there are no other associative skew fields in larger dimensions.

Similarly, one can perform the time derivative of the expression of Eq.6:

$$\dot{\mathbf{q}}_{a,(c)} = \dot{\mathbf{q}}_{b,(c)}\mathbf{q}_{a,(b)} + \mathbf{q}_{b,(c)}\dot{\mathbf{q}}_{a,(b)}. \tag{8}$$

The angular velocity $\omega_{a,(c)}$ of $\chi_{a,(c)}$, expressed in the coordinates of reference a, follows immediately the quaternion derivative $\dot{\mathbf{q}}_{a,(c)}$ obtained with Eq.8 using the following formula (discussed in [cite]):

$$\mathbf{q}_{\Im}(\omega_{a,(c)}) = 2\mathbf{q}^*_{a,(c)}\dot{\mathbf{q}}_{a,(c)}. \tag{9}$$

C. Accelerations

The last two parts of the $\chi_{a,(c)}$ vector are the acceleration $\ddot{\mathbf{r}}_{a,(c)}$ and the angular acceleration $\alpha_{a,(c)}$.

By differentiation of Eq.7:

$$\begin{aligned}
\ddot{\mathbf{q}}_{\Im}(\mathbf{r}_{a,(c)}) =& \ddot{\mathbf{q}}_{\Im}(\mathbf{r}_{b,(c)}) + 2\ddot{\mathbf{q}}_{b,(c)}\mathbf{q}_{\Im}(\mathbf{r}_{a,(b)})\mathbf{q}^*_{b,(c)} + \\
&+ 2\dot{\mathbf{q}}_{b,(c)}\dot{\mathbf{q}}_{\Im}(\mathbf{r}_{a,(b)})\mathbf{q}^*_{b,(c)} + \\
&+ 2\dot{\mathbf{q}}_{b,(c)}\mathbf{q}_{\Im}(\mathbf{r}_{a,(b)})\dot{\mathbf{q}}^*_{b,(c)} + \\
&+ \dot{\mathbf{q}}_{b,(c)}\dot{\mathbf{q}}_{\Im}(\mathbf{r}_{a,(b)})\mathbf{q}^*_{b,(c)} + \\
&+ \mathbf{q}_{b,(c)}\ddot{\mathbf{q}}_{\Im}(\mathbf{r}_{a,(b)})\mathbf{q}^*_{b,(c)} + \\
&+ \mathbf{q}_{b,(c)}\dot{\mathbf{q}}_{\Im}(\mathbf{r}_{a,(b)})\dot{\mathbf{q}}^*_{b,(c)} \\
=& \ddot{\mathbf{q}}_{\Im}(\mathbf{r}_{b,(c)}) + 2\ddot{\mathbf{q}}_{b,(c)}\mathbf{q}_{\Im}(\mathbf{r}_{a,(b)})\mathbf{q}^*_{b,(c)} + \\
&+ 4\dot{\mathbf{q}}_{b,(c)}\dot{\mathbf{q}}_{\Im}(\mathbf{r}_{a,(b)})\mathbf{q}^*_{b,(c)} + \\
&+ 2\dot{\mathbf{q}}_{b,(c)}\mathbf{q}_{\Im}(\mathbf{r}_{a,(b)})\dot{\mathbf{q}}^*_{b,(c)} + \\
&+ \mathbf{q}_{b,(c)}\ddot{\mathbf{q}}_{\Im}(\mathbf{r}_{a,(b)})\mathbf{q}^*_{b,(c)}
\end{aligned}$$

and finally, as $\ddot{\mathbf{q}}_{\Im}(\mathbf{r}_{a,(c)}) = \mathbf{q}_{\Im}(\ddot{\mathbf{r}}_{a,(c)})$, it is:

$$\begin{aligned}
\mathbf{q}_{\Im}(\ddot{\mathbf{r}}_{a,(c)}) =& \mathbf{q}_{\Im}(\ddot{\mathbf{r}}_{b,(c)}) + 2\ddot{\mathbf{q}}_{b,(c)}\mathbf{q}_{\Im}(\mathbf{r}_{a,(b)})\mathbf{q}^*_{b,(c)} + \\
&+ 4\dot{\mathbf{q}}_{b,(c)}\mathbf{q}_{\Im}(\dot{\mathbf{r}}_{a,(b)})\mathbf{q}^*_{b,(c)} + \\
&+ 2\dot{\mathbf{q}}_{b,(c)}\mathbf{q}_{\Im}(\mathbf{r}_{a,(b)})\dot{\mathbf{q}}^*_{b,(c)} + \\
&+ \mathbf{q}_{b,(c)}\mathbf{q}_{\Im}(\ddot{\mathbf{r}}_{a,(b)})\mathbf{q}^*_{b,(c)}
\end{aligned} \tag{10}$$

Angular accelerations follow from the differentiation of the epression of Eq.8:

$$\begin{aligned}
\ddot{\mathbf{q}}_{a,(c)} =& \ddot{\mathbf{q}}_{b,(c)}\mathbf{q}_{a,(b)} + \dot{\mathbf{q}}_{b,(c)}\dot{\mathbf{q}}_{a,(b)} + \\
&+ \dot{\mathbf{q}}_{b,(c)}\dot{\mathbf{q}}_{a,(b)} + \mathbf{q}_{b,(c)}\ddot{\mathbf{q}}_{a,(b)} \\
=& \ddot{\mathbf{q}}_{b,(c)}\mathbf{q}_{a,(b)} + 2\dot{\mathbf{q}}_{b,(c)}\dot{\mathbf{q}}_{a,(b)} + \mathbf{q}_{b,(c)}\ddot{\mathbf{q}}_{a,(b)}
\end{aligned} \tag{11}$$

The angular acceleration $\alpha_{a,(c)}$ of $\chi_{a,(c)}$, expressed in the coordinates of reference a, is obtained from the quaternion double derivative $\ddot{\mathbf{q}}_{a,(c)}$ of Eq.11 and from the differentiation of Eq.9:

$$\mathbf{q}_{\Im}(\alpha_{a,(c)}) = 2\dot{\mathbf{q}}^*_{a,(c)}\dot{\mathbf{q}}_{a,(c)} + 2\mathbf{q}^*_{a,(c)}\ddot{\mathbf{q}}_{a,(c)} \tag{12}$$

III PROPERTIES

In this section we will outline the most relevant properties of the $\mathcal{T}(\mathbb{T}, \succ)$ algebra, as defined in the previous paragraphs.

THEOREM 1

The $\mathcal{T}(\mathbb{T}, \succ)$ magma algebra is also a monoid featuring an identity element $\chi_I \in \mathbb{T}$.

Proof. Let consider an element $\chi_I \in \mathbb{T}$ as $\chi_I = \{\mathbf{0}, \mathbf{q}_I, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}\}$, where $\mathbf{q}_I$ is the unitary quaternion $1 + 0i + 0j + 0k$.

The product $\chi_a = \chi_b \succ \chi_I$ gives

$$\chi_a = \{\mathbf{r}_a, \mathbf{q}_a, \dot{\mathbf{r}}_a, \omega_a, \ddot{\mathbf{r}}, \alpha_a\}$$

. According to the definition 4, the position term can be made explicit as:

$$\mathbf{q}_{\Im}(\mathbf{r}_a) = \mathbf{q}_{\Im}(\mathbf{0}) + \mathbf{q}_I\mathbf{q}_{\Im}(\mathbf{r}_b)\mathbf{q}^*_I.$$

Since $\mathbf{q}_I\mathbf{q}_{\Im}(\mathbf{r}_b)\mathbf{q}^*_I = \mathbf{q}_{\Im}(\mathbf{r}_b)$ by property of quaternion multiplication, it is also

$$\mathbf{q}_{\Im}(\mathbf{r}_a) = \mathbf{q}_{\Im}(\mathbf{r}_b) \tag{13}$$

Also, Eq.6 becomes

$$\mathbf{q}_a = \mathbf{q}_I\mathbf{q}_b = \mathbf{q}_b. \tag{14}$$

Similarly, one can made explicit the speed and acceleration terms using definitions 7, 8, 10, 11: by exploining the aforementioned properties of quaternion multiplications, it is clear that most terms vanish:

$$\begin{aligned}
\mathbf{q}_{\Im}(\dot{\mathbf{r}}_a) &= \mathbf{q}_{\Im}(\mathbf{0}) + 2(\mathbf{0}\mathbf{q}_{\Im}(\mathbf{r}_b)\mathbf{q}^*_I) + \\
&\quad \mathbf{q}_I\mathbf{q}_{\Im}(\dot{\mathbf{r}}_b)\mathbf{q}^*_I = \mathbf{q}_{\Im}(\dot{\mathbf{r}}_b) && (15)\\
\dot{\mathbf{q}}_a &= \mathbf{0}\mathbf{q}_b + \mathbf{q}_I\dot{\mathbf{q}}_b = \dot{\mathbf{q}}_b && (16)\\
\mathbf{q}_{\Im}(\ddot{\mathbf{r}}_a) &= \mathbf{0} + 2(\mathbf{0}\mathbf{q}_{\Im}(\mathbf{r}_b)\mathbf{q}^*_I) + 4(\mathbf{0}\mathbf{q}_{\Im}(\dot{\mathbf{r}}_b)\mathbf{q}^*_I) + \\
&\quad + 2(\mathbf{0}\mathbf{q}_{\Im}(\mathbf{r}_b)\mathbf{0}^*) + \mathbf{q}_I\mathbf{q}_{\Im}(\ddot{\mathbf{r}}_b)\mathbf{q}^*_I = \mathbf{q}_{\Im}(\ddot{\mathbf{r}}_b) && (17)\\
\ddot{\mathbf{q}}_a &= \mathbf{0}\mathbf{q}_b + 2(\mathbf{0}\dot{\mathbf{q}}_b) + \mathbf{q}_I\ddot{\mathbf{q}}_b = \ddot{\mathbf{q}}_b. && (18)
\end{aligned}$$

Thus, from Eq.13-18 and properties 9,12, it follows that $\chi_a = \chi_b \succ \chi_I = \chi_b$, so $\succ \chi_I$ is the right-neutral element of the $\mathcal{T}(\mathbb{T}, \succ)$ algebra.

For reasons of space, we do not develop the product $\chi_a = \chi_b \succ \chi_I$; it would be easy to obtain again the same results of Eq.13-18 (although using other properties of quaternion algera for the cancellation of the terms). Hence it follows that $\succ \chi_I$ is also a left-neutral element in the $\mathcal{T}(\mathbb{T}, \succ)$ algebra.

Give that the magma has both right and left-neutral elements, $\mathcal{T}(\mathbb{T}, \succ)$ is also a monoid with $\chi_b \succ \chi_I = \chi_I \succ \chi_b = \chi_b$. QED.

THEOREM 2

The $\mathcal{T}(\mathbb{T}, \succ)$ algebra is a non-abelian group.

Proof.

For the $\mathcal{T}(\mathbb{T}, \succ)$ monoid algebra to be a group, each element χ must have an inverse element χ^{-1} such that $\chi^{-1} \succ \chi = \chi_I$, where χ_I is the neutral elment introduced in Theorem 1.

For semplicity, let's recall the notation of the product in Eq.2. If $\chi_{a,(b)} \succ \chi_{b,(c)} = \chi_I$, then $\chi_{a,(b)}$ is the left-inverse of $\chi_{b,(c)}$, and will be denoted as $\chi^{-1L}_{b,(c)}$. Also, $\chi_{b,(c)}$ is the right-inverse of $\chi_{a,(b)}$, and will be denoted as $\chi^{-1R}_{a,(b)}$. Thus, if right and left inverses exist,

$$\chi_{a,(b)} \succ \chi^{-1R}_{a,(b)} = \chi_I, \quad \chi^{-1L}_{b,(c)} \succ \chi_{b,(c)} = \chi_I$$

Imposing $\chi_{a,(c)} = \chi_I$ in the product of Eq.2, one can write $\chi_{a,(b)} \succ \chi_{b,(c)} = \chi_I$, then it will be possible to manipulate the definitions in Eq.4-11 to find the expression of the left-inverse by explicitating the terms belonging to $\chi_{a,(b)}$.

Let start from the transformation of positions. We rewrite Eq.4 as:

$$\mathbf{q}_{\Im}(\mathbf{r}_{b,(c)}) + \mathbf{q}_{b,(c)}\mathbf{q}_{\Im}(\mathbf{r}_{a,(b)})\mathbf{q}^*_{b,(c)} = \mathbf{q}_{\Im}(\mathbf{0}).$$

Let left-multiply all terms by quaternion $\mathbf{q}^{-1}_{b,(c)}$ and right-multiply all terms by $\mathbf{q}^{*-1}_{b,(c)}$. By remembering quaternion algebra properties $\mathbf{q}\mathbf{q}^{-1} = \{1,0,0,0\}$ and $\mathbf{q}\{1,0,0,0\} = \mathbf{q}$, $\mathbf{q} \in \mathbb{Q}$, it is easy to find:

$$\mathbf{q}_{\Im}(\mathbf{r}_{a,(b)}) = -\mathbf{q}^{-1}_{b,(c)}\mathbf{q}_{\Im}(\mathbf{r}_{b,(c)})\mathbf{q}^{*-1}_{b,(c)}. \tag{19}$$

This is the first element of the left-inverse vector, that in our proof is $\chi_{a,(b)} = \chi^{-1L}_{b,(c)}$.

Coming to rotations, remembering that the rotation part $\mathbf{q}$ in the neutral element χ_I is the unit quaternion $\{1,0,0,0\}$ as demonstrated in Theorem 1, we rewrite Eq.6 as follows:

$$\mathbf{q}_{b,(c)}\mathbf{q}_{a,(b)} = \{1,0,0,0\}.$$

Therefore, using quaternion inverses, premultiplying the terms by $\mathbf{q}^{-1}_{b,(c)}$ and simplifying, one gets the rotational part of the left-inverse:

$$\mathbf{q}_{a,(b)} = \mathbf{q}^{-1}_{b,(c)}. \tag{20}$$

For the speed part, one imposes that Eq.7 equals to zero:

$$\begin{aligned}\mathbf{q}_{\Im}(\dot{\mathbf{r}}_{b,(c)}) + 2\dot{\mathbf{q}}_{b,(c)}\mathbf{q}_{\Im}(\mathbf{r}_{a,(b)})\mathbf{q}^*_{b,(c)} + \\ + \mathbf{q}_{b,(c)}\mathbf{q}_{\Im}(\dot{\mathbf{r}}_{a,(b)})\mathbf{q}^*_{b,(c)} = \mathbf{q}_{\Im}(\mathbf{0})\end{aligned}$$

Here, few manipulations with quaternion algebra will produce the following result:

$$\begin{aligned}\mathbf{q}_{\Im}(\dot{\mathbf{r}}_{a,(b)}) =& \mathbf{q}^{-1}_{b,(c)}(-\mathbf{q}_{\Im}(\dot{\mathbf{r}}_{b,(c)}) + \\ &+ 2\dot{\mathbf{q}}_{b,(c)}\mathbf{q}^{-1}_{b,(c)}\mathbf{q}_{\Im}(\mathbf{r}_{b,(c)}))\mathbf{q}^{*-1}_{b,(c)}.\end{aligned} \tag{21}$$

Also, imposing that $\dot{\mathbf{q}}_{a,(c)}$ is null in Eq.8, it follows:

$$\dot{\mathbf{q}}_{a,(b)} = -\mathbf{q}^{-1}_{b,(c)}\dot{\mathbf{q}}_{b,(c)}\mathbf{q}^{-1}_{b,(c)}. \tag{22}$$

With similar algebraic manipulations, and remembering that $\mathbf{q}^{*-1}\mathbf{q}^* = \{1,0,0,0\}$, one can get the accleration part of the left inverse, obtaining:

$$\begin{aligned}\mathbf{q}_{\Im}(\ddot{\mathbf{r}}_{a,(b)}) =& \mathbf{q}^{-1}_{b,(c)}[-\mathbf{q}_{\Im}(\ddot{\mathbf{r}}_{b,(c)}) + \\ &+ 2\ddot{\mathbf{q}}_{b,(c)}\mathbf{q}^{-1}_{b,(c)}\mathbf{q}_{\Im}(\mathbf{r}_{b,(c)}) \\ &+ 4\dot{\mathbf{q}}_{b,(c)}\mathbf{q}^{-1}_{b,(c)}(\mathbf{q}_{\Im}(\dot{\mathbf{r}}_{b,(c)}) \\ &+ -2\dot{\mathbf{q}}_{b,(c)}\mathbf{q}^{-1}_{b,(c)}\mathbf{q}_{\Im}(\mathbf{r}_{b,(c)})) + \\ &+ 2\dot{\mathbf{q}}_{b,(c)}\mathbf{q}^{-1}_{b,(c)}\mathbf{q}_{\Im}(\mathbf{r}_{b,(c)})\mathbf{q}^{-1}_{b,(c)}\dot{\mathbf{q}}^*_{b,(c)}]\mathbf{q}^{*-1}_{b,(c)}\end{aligned} \tag{23}$$

and

$$\ddot{\mathbf{q}}_{a,(b)} = \mathbf{q}^{-1}_{b,(c)}(\ddot{\mathbf{q}}_{b,(c)} - 2\dot{\mathbf{q}}_{b,(c)}\mathbf{q}^{-1}_{b,(c)}\dot{\mathbf{q}}_{b,(c)})\mathbf{q}^{-1}_{b,(c)}. \tag{24}$$

Finally, using Eq.9 and 12, one can merge Eq.19-24 into $\chi_{a,(b)}$, that is the left-inverse $\chi^{-1L}_{b,(c)}$ which satisfies $\chi_{a,(b)} \succ \chi_{b,(c)} = \chi^{-1L}_{b,(c)} \succ \chi_{b,(c)} = \chi_I$.

Similarly, we could solve $\chi_{a,(b)} \succ \chi_{b,(c)} = \chi_I$ for $\chi_{b,(c)}$: after long symbolic manipulation (not reported in these pages in sake of compactness) we would obtain the same results of Eq.19-24, but with inverted subscripts, that is with $a,(b)$ swapped with $b,(c)$. Therefore, we built the right-inverse $\chi^{-1R}_{a,(b)}$ and we conclude that, for a generic element $\chi \in \mathbb{T}$, right- and left-inverses are the same, that is $\chi^{-1R} = \chi^{-1L} = \chi^{-1}$. Given the existence of the inverse, the algebra is a group.

Although associative, the group is non-abelian since the $\succ$ operation is non commutative: this follows from the fact that quaternion algebra is a skew field. QED.

IV SOFTWARE IMPLEMENTATION

In order to test the efficiency and the correctness ot the theoretical framework, we implemented a set of software libraries for coordinate transformation using the C++ language.

Our framework exploits the objects-oriented concepts, so $\chi \in \mathbb{T}$ elements are represented by objects inherited from a C++ class, named `ChMovingFrame`.

After extensive benchmarking, we noticed that it is more convenient to work with objects where angular speeds and angular accelerations are represented directly with quaternions $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ rather than with 3D vectors ω and α. Thank to the encapsulation paradigm of OOP, this design does not affect the way the programmer interacts with the data, because custom functions can provide ω and α only when requested, by

evaluating Eq.9 and Eq.12. Viceversa, the user can provide ω or α, and these are instantly converted to $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ using inverse formulas.

Since we used this software library in many engineering projects, we were able to make a statistical analysis and we found that, in most cases, objects of `ChMovingFrame` type are used simply to transform 3D points, and only in few cases there is some interest also in speeds and accelerations. This means that the most important function is the one in Eq.4, which we implemented in different flavours for optimal execution speed. For example, if a single `ChMovingFrame` object must transform many 3D vectors at once, we can use Eq.5, which is a bit faster than Eq.4 because rotation by matrix-vector multiplication (after the matrix has been initialized once, with the nine values) takes less time than computing the quaternion endomorphism. However this optimization implies that a 3x3 matrix is stored in the `ChMovingFrame` object, for easing the case of multiple point transformations; the nine values of the matrix are recomputed when the rotation of the frame changes. The improved performance is worth while the overhead of keeping such matrix updated, and the increased memory requirement.

So far, each `ChMovingFrame` object will contain three vectors, three quaternions and an auxiliary 3x3 matrix:

$$\chi_{c++} = \{\mathbf{r}, \mathbf{q}, [A(\mathbf{q})], \dot{\mathbf{r}}, \dot{\mathbf{q}}, \ddot{\mathbf{r}}, \ddot{\mathbf{q}}\}.$$

An useful feature of the C++ language is the *operator overloading*, wich allows a straightforward mapping of the $\mathcal{T}(\mathbb{T}, \succ)$ algebra into a new programming syntax where the $\succ$ operator can be represented as a binary operator between two `ChMovingFrame` objects. To avoid confusion with other default operators `*,+,-,/`, we decided to use the `>>` symbol to represent the $\succ$ operation.

Such operation is implemented in the header of the `ChMovingFrame` class, using the following binary operator overloading:

```
ChFrameMoving<Real>
operator >> (const ChFrameMoving<Real>& Fb)
const
...  etc ...
```

In the function above, the formulas in Eq.4-11 are evaluated, where the return value represents the resulting vector $\chi_{a,(c)}$, the object itself (the `this` pointer) is the left argument $\chi_{a,(b)}$ and parameter `Fb` is the right argument $\chi_{b,(c)}$.

Thus, the example of Eq.1 (see Fig.1) could be written with the following source code:

```
ChMovingFrame<> cs_30, cs_32, cs_21, cs_10;
cs_10.coord.pos = ChVector<>(2,4,1);
...  etc ...
cs_30 = cs_32 >> cs_21 >> cs_10;
```

We also implemented the $(\cdot)^{-1}$ operation, with arity $\bar{\alpha} = 1$, by overloading the `!` C++ unary operator. This requires the in-place evaluation of Eq.19-24. Thank to the implementation of the inversion, it is possible, again in example of Fig.1, to obtain `cs_32` if other frames are known: we multiply both sides by `!cs_10` and `!cs_21`, and remembering that `cs_ij >> !cs_ij` will cancel by Theorems 1 and 2, we simply write

```
cs_32 = cs_30 >> !cs_10 >> !cs_21;
```

Note that the previous statement would require two inversions and two coordinate transformations, but a more efficient approach can be developed. In fact we can implement an *inverse transformation* operator, named `<<`, which requires fewer CPU operations:

```
cs_32 = cs_30 << cs_10 << cs_21;
```

For reasons of space, details about the implementation of the `<<` operator are not given; suffices to say that formulas are not much different from the ones in Eq.19-24.

The `ChMovingFrame` class inherits the functionality of a parent class `ChFrame`, which features simplified functions for cases which do not need speed or acceleration data. All classes are templated and metaprogrammed [2], they can work with floating point in double or single precision.

We remark that we performed intense benchmarks and deep profiling of the code, to obtain the best tradeoff between computational efficiency, ease of use and exploitation of OOP features [3].

The libraries for the $\mathcal{T}$ algebra has been extensively used in our Chrono::Engine C++ library for multi body simulation [6] [1]. After testing and profiling we got satisfying results in terms of clean code, ease of development and fast computation. Such a library has been succesfully used in many engineering projects by hundreds of programmers and users.

V CONCLUSION

This paper introduced the $\mathcal{T}(\mathbb{T}, \succ)$ algebra as a compact formal method to represent kinematic transformations in chains of moving frames.

Two theorems about the algebraic structure have been demonstrated, showing that the $\mathcal{T}(\mathbb{T}, \succ)$ magma is a non-abelian group.

This formal framework maps well into a software implementation, thank to the operator-overloading of the C++ language. Some practical issues and examples have been discussed.

This algebra is implemented in our Chrono::Engine library for multibody simulation, hence it found application in many engineering fields and is currently adopted in many research centers around the world.

REFERENCES

[1] Paolo Righettini Alessandro Tasora, Marco Silvestri. Architecture of the chrono::engine physics simulation middleware. In *Proceedings of Multibody Dynamics 2007, ECCOMAS thematic conference*, Milano, Italy, June 2007.

[2] A. Alexandrescu. *Modern C++ Design: Generic Programming and Design Patterns Applied*. Addison-Wesley, New York, 2001.

[3] A. Alexandrescu H. Sutter. *C++ Coding Standards: 101 Rules, Guidelines, and Best Practices*. Addison-Wesley, New York, 2004.

[4] M. Vidyasagar M.W.Spong. *Robot Dynamics and Control*. Wiley, New York, 1989.

[5] A. Shabana. *Multibody Systems*. John Wiley and Sons, New York, 1989.

[6] A. Tasora. Chrono::engine project, web page, 2006.

[7] H. Weyl. *The Theory of Groups and Quantum Mechanics*. Dover Publications, New York, 1950.