Fourth-Order Automatic Differentiation for Robotic Kinematics via Nilpotent Hypercomplex Algebra

Daniel Condurache and Mihail Cojocari

Abstract— This paper presents an automatic differentiation framework for robotic kinematics based on a fourth-order nilpotent hypercomplex algebra. Derivative information is embedded directly into the hypercomplex numerical representation, enabling the exact computation of first-, second-, third-, and fourth-order motion quantities through algebraic operations, without finite-difference approximation, explicit time differentiation, or recursive construction of higher-order fields. Building on the Lie-group structure of rigid-body displacements, the rigid-body hyper-state is introduced as a hypercomplex homogeneous transformation over SE (3), in which pose, velocity, acceleration, jerk, and snap are encoded within a single nilpotent algebraic object. The formulation is extended to serial kinematic chains through the product-of-exponentials map, yielding a compact procedure for computing the end-effector hyper-state from joint variables and their higher-order time derivatives. The method is implemented for the KUKA LBR iiwa 7 R800 manipulator, where the end-effector hyper-state coefficients up to snap are obtained for a prescribed joint trajectory.

Keywords — Automatic differentiation, nilpotent hypercomplex algebra, rigid-body hyper-state, higher-order robotic kinematics, Lie group SE (3), product-of-exponentials.

I. Introduction

Higher-order robotic kinematics is no longer a purely theoretical extension of classical velocity and acceleration analysis. In smooth trajectory generation, high-precision motion control, and vibration-sensitive robotic tasks, the behavior of jerk and snap becomes part of the usable kinematic description of the mechanism. For serial manipulators, however, these quantities are difficult to compute in a form that remains compact, numerically stable, and structurally consistent with rigid-body motion. The difficulty is not only the order of differentiation, but also the preservation of the geometric coupling between rotation and translation throughout the kinematic chain.

Automatic differentiation (AD) provides a rigorous computational mechanism for propagating derivative information through elementary numerical operations. Unlike finite-difference schemes, it does not require a perturbation step, and unlike conventional symbolic differentiation, it avoids the repeated expansion of large closed-form expressions [1]–[3]. These properties make AD suitable for nonlinear computational models in which accuracy, stability, and implementation robustness are essential [4], [5]. In robotic kinematics, the same requirements arise naturally from the composition of rigid-body displacements and from the need to evaluate velocity, acceleration, jerk, snap, and higher-order motion quantities in a unified way [6]– [8].

Hypercomplex algebraic representations provide an effective way to carry this derivative information inside the kinematic variables themselves. Dual numbers, hyper-dual numbers, quaternions, and related constructions have been used to encode rigid-body motion and its derivatives while preserving the coupling between translational and rotational components [9]– [13]. Homogeneous transformation matrices remain particularly appropriate for serial-chain models because they are directly compatible with the product structure of rigid-body displacements [14]. When a nilpotent hypercomplex algebra is combined with this homogeneous representation, the derivative orders are propagated by the same algebraic operations that propagate the motion.

Higher-order rigid-body motion can be expressed through invariant tensor-vector fields associated with the Lie-group structure of rigid-body displacements. Recursive, Lie-algebraic, dual, multidual, and hyper-multidual formulations have been developed for rigid bodies, multibody systems, and lower-pair serial chains [15]– [23]. The present paper focuses on the fourth-order computational realization of this idea for serial robotic kinematics. The rigid-body hyper-state is represented as a hypercomplex homogeneous transformation over $SE(3)$, in which the pose and the velocity, acceleration, jerk, and snap fields are encoded within a single nilpotent algebraic object.

The main contributions of this paper are as follows. First, a fourth-order nilpotent hypercomplex algebra is used to compute first-, second-, third-, and fourth-order motion quantities exactly by algebraic coefficient extraction, without finite-difference approximation, repeated differentiation of the complete kinematic mapping, or recursive construction of higher-order fields. Second, the rigid-body hyper-state is formulated as a compact hypercomplex homogeneous transformation over $SE(3)$, providing a direct encoding of pose, velocity, acceleration, jerk, and snap. Third, the formulation is specialized to serial kinematic chains through the product-of-exponentials map, yielding a concise procedure for computing the end-effector hyper-state from joint variables and their higher-order time derivatives. The procedure is implemented for the KUKA LBR iiwa 7 R800 manipulator, where the nilpotent coefficients of the end-effector hyper-state, corresponding to velocity, acceleration, jerk, and snap fields, are obtained for a prescribed joint trajectory.

Daniel Condurache is with Technical University of Iasi, D. Mangeron Street no.59, 700050, Iasi, Romania and with Research Center for Industrial Robots Simulation and Testing, Technical University of Cluj-Napoca, Memorandumului 14, 400114, Cluj-Napoca, Romania (corresponding author; e-mail: daniel.condurache@tuiasi.ro).

Mihail Cojocari is with Technical University of Iasi, D. Mangeron Street no.59, 700050, Iasi, Romania (e-mail: mihail.cojocari@academic.tuiasi.ro).

II. HIGHER-ORDER RIGID-BODY MOTION: AN OVERVIEW

A rigid-body displacement is treated here as a time-parametrized curve on $SE(3)$. In the fourth-order setting considered in this paper, the motion is represented in three-dimensional Euclidean space $\mathbb{E}^3$ by the homogeneous transformation

$$\mathbf{g}(t) = \begin{bmatrix} \mathbf{R}(t) & \mathbf{r}_Q(t) \\ \mathbf{0} & 1 \end{bmatrix}, \quad (1)$$

where $\mathbf{R}(t) \in SO(3)$ is a proper orthogonal rotation tensor and $\mathbf{r}_Q(t) \in \mathbb{V}_3$ is the position vector of a reference point Q attached to the rigid body. The time parameter satisfies $t \in \mathbb{I} \subseteq \mathbb{R}$. Since the present paper addresses velocity, acceleration, jerk, and snap, the functions $\mathbf{R}(t)$ and $\mathbf{r}_Q(t)$ are assumed to be differentiable up to order four on $\mathbb{I}$.

The higher-order motion of the rigid body is described by affine vector fields. For an arbitrary body point characterized by the position vector $\mathbf{r} \in \mathbb{V}_3$, the value of the j-th order motion field is written as

$$\mathbf{a}_{\mathbf{r}}^{(j)} = \mathbf{a}_j + \mathbf{\Phi}_j \mathbf{r}, \;\; \mathbf{r} \in \mathbb{V}_3, \;\; j = 1, \dots, 4. \quad (2)$$

Here, $\mathbf{\Phi}_j$ is the j-th order invariant tensor and $\mathbf{a}_j$ is the corresponding invariant vector. These quantities are defined by

$$\mathbf{\Phi}_j = \mathbf{R}^{(j)} \mathbf{R}^T, \;\; j = 1, \dots, 4, \quad (3)$$

$$\mathbf{a}_j = \mathbf{r}_Q^{(j)} - \mathbf{\Phi}_j \mathbf{r}_Q, \;\; j = 1, \dots, 4. \quad (4)$$

Thus, the pair $(\mathbf{\Phi}_j, \mathbf{a}_j)$ gives the invariant description of the complete j-th order motion field over the rigid body. The vector $\mathbf{a}_j$ is not the direct j-th derivative of the reference-point position vector; it is obtained after subtracting the rotational contribution $\mathbf{\Phi}_j \mathbf{r}_Q$. These tensors and vectors generalize the classical velocity and acceleration tensors and the associated velocity and acceleration invariants [15], [16].

The invariant quantities satisfy the recursive differential relations [15]

$$\begin{cases} \mathbf{\Phi}_j = \dot{\mathbf{\Phi}}_{j-1} + \mathbf{\Phi}_{j-1} \mathbf{\Phi}_1, \\ \mathbf{a}_j = \dot{\mathbf{a}}_{j-1} + \mathbf{\Phi}_{j-1} \mathbf{a}_1, \end{cases} \;\; j = 2, 3, 4. \quad (5)$$

The first-order quantities are

$$\mathbf{\Phi}_1 = \dot{\mathbf{R}} \mathbf{R}^T \equiv (\boldsymbol{\omega} \times), \;\; \mathbf{a}_1 = \dot{\mathbf{r}}_Q - (\boldsymbol{\omega} \times) \mathbf{r}_Q \equiv \mathbf{v}. \quad (6)$$

In (6), $(\boldsymbol{\omega} \times)$ denotes the skew-symmetric tensor associated with the instantaneous angular velocity vector $\boldsymbol{\omega} = \mathrm{vect}(\dot{\mathbf{R}} \mathbf{R}^T)$. Relations (5) give the classical recursive characterization of the higher-order invariant fields. In the present paper, they are used as the reference kinematic structure; the quantities $(\mathbf{\Phi}_j, \mathbf{a}_j)$ are subsequently recovered directly from the nilpotent coefficients of a hypercomplex homogeneous transformation.

III. FOURTH-ORDER NILPOTENT HYPERCOMPLEX ALGEBRA

A. Hypercomplex Numbers

Following the nilpotent hypercomplex constructions used in [17]– [21], we consider the fourth-order scalar algebra

$$\widehat{\mathbb{R}} = \mathbb{R} \oplus \varepsilon \mathbb{R} \oplus \varepsilon^2 \mathbb{R} \oplus \varepsilon^3 \mathbb{R} \oplus \varepsilon^4 \mathbb{R}, \varepsilon \neq 0, \varepsilon^5 = 0. \quad (7)$$

Here, $\mathbb{R}$ denotes the field of real numbers, and ε is a nilpotent generator with nilpotency index five. A generic element $\hat{x} \in \widehat{\mathbb{R}}$ is written as

$$\begin{gathered} \hat{x} = x + x_1 \varepsilon + x_2 \varepsilon^2 + x_3 \varepsilon^3 + x_4 \varepsilon^4, \\ x, x_j \in \mathbb{R}, \; j = 1, \dots, 4. \end{gathered} \quad (8)$$

Similarly, for $\hat{y} \in \widehat{\mathbb{R}}$,

$$\begin{gathered} \hat{y} = y + y_1 \varepsilon + y_2 \varepsilon^2 + y_3 \varepsilon^3 + y_4 \varepsilon^4, \\ y, y_j \in \mathbb{R}, \; j = 1, \dots, 4. \end{gathered} \quad (9)$$

The real part of $\hat{x}$ is denoted by

$$\mathrm{Re}(\hat{x}) = x, \quad (10)$$

whereas its nilpotent part is

$$\mathrm{Mu}(\hat{x}) = \Delta \hat{x} = x_1 \varepsilon + x_2 \varepsilon^2 + x_3 \varepsilon^3 + x_4 \varepsilon^4. \quad (11)$$

The coefficients x_j represent the j -th order hypercomplex components of $\hat{x}$, and are defined as

$$x_j = [\varepsilon^j] \hat{x}, \quad j = 1, \dots, 4. \quad (12)$$

The addition and multiplication in $\widehat{\mathbb{R}}$ are defined by

$$\hat{x} + \hat{y} = \textstyle\sum_{j=0}^{4} (x_j + y_j) \varepsilon^j, \quad (13)$$

$$\hat{x} \hat{y} = \sum_{j=0}^{4} \left(\textstyle\sum_{p=0}^{j} x_p \, y_{j-p} \right) \varepsilon^j, \quad (14)$$

with the conventions $\varepsilon^0 = 1$, $x_0 = x$, and $y_0 = y$.

The powers of the nilpotent generator satisfy

$$\varepsilon^j \varepsilon^p = \varepsilon^p \varepsilon^j = \begin{cases} \varepsilon^{j+p}, & j + p \leq 4, \\ 0, & j + p \geq 5, \end{cases} \quad (15)$$

Therefore, $\widehat{\mathbb{R}}$ is a commutative and associative algebra with unity, with $\mathbb{R}$ embedded as a subalgebra. An element $\hat{x} \in \widehat{\mathbb{R}}$ is invertible if and only if Re $(\hat{x}) \neq 0$. Elements with zero real part are zero divisors.

Let $f \colon \mathbb{I} \subseteq \mathbb{R} \to \mathbb{R}$ be a real-valued function of class C^4. Its extension to a hypercomplex argument $\hat{x} \in \widehat{\mathbb{R}}$, with $x =$ Re $(\hat{x}) \in \mathbb{I}$, is defined by the finite nilpotent expansion

$$f(\hat{x}) = f(x) + \textstyle\sum_{j=1}^{4} \frac{(\Delta \hat{x})^j}{j!} f^{(j)}(x), \; \Delta \hat{x} = \hat{x} - x. \quad (16)$$

Since $(\Delta \hat{x})^5 = 0$, the expansion terminates exactly at order four. Hence,

$$f(\hat{x}) = f(x) + f_1(x, x_1)\varepsilon + f_2(x, x_1, x_2)\varepsilon^2 + \\ + f_3(x, x_1, x_2, x_3)\varepsilon^3 + f_4(x, x_1, x_2, x_3, x_4)\varepsilon^4. \quad (17)$$

The finite character of (16) follows directly from nilpotency; no convergence condition for an infinite Taylor series is involved.

Proposition 1. *Let f, g, and h be real-valued functions, four times differentiable on an interval* $\mathbb{I}$

$$f(x) = g(x)h(x),\ x \in \mathbb{I}. \quad (18)$$

Then, for any hypercomplex argument $\hat{x} \in \widehat{\mathbb{R}}$ *with real part* $\mathrm{Re}(\hat{x}) \in \mathbb{I}$, *the following identity holds*:

$$f(\hat{x}) = g(\hat{x})h(\hat{x}). \quad (19)$$

Proof: The result follows directly from the definition of function extension to hypercomplex variables, as given in (16), together with the Leibniz rule for higher-order derivatives. ■.

Proposition 1 provides a rigorous algebraic justification for extending differentiable real-valued functions to the fourth-order nilpotent hypercomplex setting. The construction uses only a finite nilpotent expansion; therefore, no infinite Taylor series and no convergence assumption are involved. It recovers the usual dual-number extension as a particular case and extends it to the present finite nilpotent algebra.

B. Hypercomplex Vectors

The space of fourth-order hypercomplex vectors is defined as the free $\widehat{\mathbb{R}}$-module

$$\widehat{\mathbb{V}}_3 = \mathbb{V}_3 \oplus \varepsilon\mathbb{V}_3 \oplus \varepsilon^2\mathbb{V}_3 \oplus \varepsilon^3\mathbb{V}_3 \oplus \varepsilon^4\mathbb{V}_3, \\ \varepsilon \neq 0, \varepsilon^5 = 0. \quad (20)$$

Here, $\mathbb{V}_3$ denotes the vector space of free vectors in three-dimensional Euclidean space. A generic hypercomplex vector $\hat{\mathbf{a}} \in \widehat{\mathbb{V}}_3$ is expressed as

$$\hat{\boldsymbol{a}} = \boldsymbol{a}_0 + \sum_{j=1}^{4} \boldsymbol{a}_j \varepsilon^j,\ \boldsymbol{a}_j \in \mathbb{V}_3,\ j = 1, \dots, 4. \quad (21)$$

Similarly,

$$\hat{\mathbf{b}} = \mathbf{b}_0 + \sum_{j=1}^{4} \mathbf{b}_j \varepsilon^j, \mathbf{b}_j \in \mathbb{V}_3,\ j = 1, \dots, 4. \quad (22)$$

Addition in $\widehat{\mathbb{V}}_3$ is defined component-wise by:

$$\hat{\mathbf{a}} + \hat{\mathbf{b}} = (\mathbf{a}_0 + \mathbf{b}_0) + \sum_{j=1}^{4} (\mathbf{a}_j + \mathbf{b}_j)\varepsilon^j. \quad (23)$$

Scalar multiplication of $\hat{\mathbf{a}} \in \widehat{\mathbb{V}}_3$ by $\hat{x} \in \widehat{\mathbb{R}}$ is defined as

$$\hat{x}\hat{\mathbf{a}} = \sum_{j=0}^{4} \left(\sum_{p=0}^{j} x_p\, \mathbf{a}_{j-p} \right) \varepsilon^j. \quad (24)$$

With these operations, $\widehat{\mathbb{V}}_3$ is a free module of rank three over the hypercomplex ring $\widehat{\mathbb{R}}$.

The scalar (dot) product $\hat{\mathbf{a}} \cdot \hat{\mathbf{b}}$, and the vector (cross) product $\hat{\mathbf{a}} \times \hat{\mathbf{b}}$ of two hypercomplex vectors are defined by equations (21), (22), and the multiplication rule (15). For three hypercomplex vectors $\hat{\mathbf{a}}, \hat{\mathbf{b}}, \hat{\mathbf{c}} \in \widehat{\mathbb{V}}_3$, the scalar triple product is defined as $\langle \hat{\mathbf{a}}, \hat{\mathbf{b}}, \hat{\mathbf{c}} \rangle = \hat{\mathbf{a}} \cdot (\hat{\mathbf{b}} \times \hat{\mathbf{c}})$. A set of hypercomplex vectors $\{\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \hat{\mathbf{e}}_3\}$ forms a basis of $\widehat{\mathbb{V}}_3$ if and only if

$$\mathrm{Re}(\langle \hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \hat{\mathbf{e}}_3 \rangle) \neq 0. \quad (25)$$

C. Hypercomplex Tensors

An $\widehat{\mathbb{R}}$-linear mapping from $\widehat{\mathbb{V}}_3$to itself is called a **Euclidean hypercomplex tensor**, i.e.,

$$\hat{\mathbf{T}}(\hat{\lambda}_1\hat{\mathbf{v}}_1 + \hat{\lambda}_2\hat{\mathbf{v}}_2) = \hat{\lambda}_1\hat{\mathbf{T}}(\hat{\mathbf{v}}_1) + \hat{\lambda}_2\hat{\mathbf{T}}(\hat{\mathbf{v}}_2), \\ \forall\hat{\lambda}_1, \hat{\lambda}_2 \in \widehat{\mathbb{R}}, \forall\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2 \in \widehat{\mathbb{V}}_3. \quad (26)$$

Let $\mathcal{L}(\widehat{\mathbb{V}}_3, \widehat{\mathbb{V}}_3)$ denote the set of hypercomplex tensors. For any $\hat{\mathbf{T}} \in \mathcal{L}(\widehat{\mathbb{V}}_3, \widehat{\mathbb{V}}_3)$, the transpose hypercomplex tensor $\hat{\mathbf{T}}^T$is defined by

$$\hat{\mathbf{v}}_1 \cdot (\hat{\mathbf{T}}\hat{\mathbf{v}}_2) = \hat{\mathbf{v}}_2 \cdot (\hat{\mathbf{T}}^T\hat{\mathbf{v}}_1), \forall\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2 \in \widehat{\mathbb{V}}_3. \quad (27)$$

The determinant of a hypercomplex tensor $\hat{\mathbf{T}}$ is defined through the relation

$$\langle \hat{\mathbf{T}}\hat{\mathbf{v}}_1, \hat{\mathbf{T}}\hat{\mathbf{v}}_2, \hat{\mathbf{T}}\hat{\mathbf{v}}_3 \rangle = \det(\hat{\mathbf{T}})\,\langle \hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \hat{\mathbf{v}}_3 \rangle, \quad (28)$$

for all $\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \hat{\mathbf{v}}_3 \in \widehat{\mathbb{V}}_3$ satisfying $\mathrm{Re}(\langle \hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \hat{\mathbf{v}}_3 \rangle) \neq 0$.

For any hypercomplex vector $\hat{\mathbf{a}} \in \widehat{\mathbb{V}}_3$, the associated skew-symmetric hypercomplex tensor, denoted by $(\hat{\mathbf{a}}\,\times)$, is defined by

$$(\hat{\mathbf{a}}\,\times)\hat{\mathbf{b}} = \hat{\mathbf{a}} \times \hat{\mathbf{b}}, \forall\hat{\mathbf{b}} \in \widehat{\mathbb{V}}_3. \quad (29)$$

The set of skew-symmetric hypercomplex tensors forms a free $\widehat{\mathbb{R}}$-module of rank three, naturally isomorphic to $\widehat{\mathbb{V}}_3$.

The group of proper orthogonal hypercomplex tensors is defined as

$$\widehat{SO}(3) = \{\hat{\mathbf{R}} \in \mathcal{L}(\widehat{\mathbb{V}}_3, \widehat{\mathbb{V}}_3), \hat{\mathbf{R}}\hat{\mathbf{R}}^T = \hat{\mathbf{I}}, \det(\hat{\mathbf{R}}) = 1\} \quad (30)$$

When a proper orthogonal hypercomplex tensor is represented by a hypercomplex axis-angle pair $(\hat{\mathbf{u}}, \hat{\alpha})$, with $\hat{\mathbf{u}} \cdot \hat{\mathbf{u}} = 1$, it admits the Rodrigues-type representation

$$\hat{\mathbf{R}} = \hat{\mathbf{I}} + \sin\hat{\alpha}\,(\hat{\mathbf{u}}\,\times) + (1 - \cos\hat{\alpha})(\hat{\mathbf{u}}\,\times)^2 \\ = \exp[\hat{\alpha}(\hat{\mathbf{u}}\,\times)] \quad (31)$$

Here, $\hat{\alpha}$ is a hypercomplex angle whose real part satisfies $\mathrm{Re}\,(\hat{\alpha}) \in [0, 2\pi)$.

D. Fourth-Order Hypercomplex Differential Transform

We now define the fourth-order hypercomplex differential transform used in the kinematic construction. Let $f: \mathbb{I} \subseteq \mathbb{R} \to \mathbb{R}$ be a real-valued function of class C^4. The associated hypercomplex-valued function $\breve{f}$ is

$$\breve{f}(t) = f(t) + \varepsilon\dot{f}(t) + \frac{\varepsilon^2}{2}\ddot{f}(t) + \frac{\varepsilon^3}{6}\dddot{f}(t) + \frac{\varepsilon^4}{24}f^{(4)}(t). \quad (32)$$

This construction stores the first-, second-, third-, and fourth-order time derivatives in a single nilpotent algebraic object. In the kinematic formulation below, the same algebraic operations that propagate the rigid-body motion also propagate derivative information up to snap.

Theorem 1. *Let $f, g \in C^4(\mathbb{I})$. The fourth-order hypercomplex differential transform satisfies the following identities whenever the expressions are well defined:*

$$\widetilde{f+g} = \breve{f} + \breve{g}, \tag{33}$$

$$\widetilde{fg} = \breve{f}\breve{g}, \tag{34}$$

$$\widetilde{\lambda f} = \lambda \breve{f}, \forall \lambda \in \mathbb{R}, \tag{35}$$

$$f\big(\widetilde{\alpha(t)}\big) = f\big(\breve{\alpha}(t)\big),\ \alpha \in C^4(\mathbb{I}), \tag{36}$$

Proof: The properties of the hypercomplex algebra and the relation are used:

$$\breve{f} = e^{\varepsilon \mathbb{D}} f, \tag{37}$$

where $e^{\varepsilon\mathbb{D}} = \mathbf{I} + \varepsilon\mathbb{D} + \frac{\varepsilon^2}{2}\mathbb{D}^2 + \frac{\varepsilon^3}{6}\mathbb{D}^3 + \frac{\varepsilon^4}{24}\mathbb{D}^4$ with $\mathbb{D} = \frac{d}{dt}$ the derivative operator with respect to time. ■

IV. Hyper-State of a Rigid Body with the Hypercomplex Differential Transform

For the rigid-body motion defined in (1), with $\mathbf{R}(t)$ and $\mathbf{r}_Q(t)$ differentiable up to order four on $\mathbb{I}$, the fourth-order hyper-state is constructed by applying the hypercomplex differential transform component-wise to the rotation tensor and to the reference-point position vector. This gives the hypercomplex homogeneous transformation

$$\breve{\mathbf{g}} = \begin{bmatrix} \breve{\mathbf{R}} & \breve{\mathbf{r}}_Q \\ \mathbf{0} & 1 \end{bmatrix}. \tag{38}$$

This representation combines the pose and its derivative information up to fourth order into a single algebraic object, enabling unified, efficient higher-order rigid-body kinematics.

Define the normalized homogeneous hypercomplex matrix:

$$\hat{\mathbf{h}} = \breve{\mathbf{g}}\mathbf{g}^{-1} = \begin{bmatrix} \breve{\mathbf{R}} & \breve{\mathbf{r}}_Q \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{r}_Q \\ \mathbf{0} & 1 \end{bmatrix}. \tag{39}$$

After direct multiplication, one obtains:

$$\hat{\mathbf{h}} = \begin{bmatrix} \widehat{\mathbf{\Phi}} & \hat{\mathbf{a}} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{40}$$

with

$$\widehat{\mathbf{\Phi}} = \breve{\mathbf{R}}\mathbf{R}^T \in \widehat{SO}(3)\,, \tag{41}$$

$$\hat{\mathbf{a}} = \left(\breve{\mathbf{r}}_Q - \breve{\mathbf{R}}\mathbf{R}^{\mathrm{T}}\mathbf{r}_Q\right) \in \widehat{\mathbb{V}}_3. \tag{42}$$

From (38), (39), (41), and (42), one obtains:

$$\widehat{\mathbf{\Phi}} = \mathbf{I} + \mathbf{\Phi}_1\varepsilon + \mathbf{\Phi}_2\frac{\varepsilon^2}{2} + \mathbf{\Phi}_3\frac{\varepsilon^3}{6} + \mathbf{\Phi}_4\frac{\varepsilon^4}{24}, \tag{43}$$

$$\hat{\mathbf{a}} = \mathbf{a}_1\varepsilon + \mathbf{a}_2\frac{\varepsilon^2}{2} + \mathbf{a}_3\frac{\varepsilon^3}{6} + \mathbf{a}_4\frac{\varepsilon^4}{24}. \tag{44}$$

The hypercomplex vector

$$\begin{bmatrix} \hat{\mathbf{a}}_{\mathbf{r}} \\ 1 \end{bmatrix} = \hat{\mathbf{h}}\begin{bmatrix} \mathbf{r} \\ 1 \end{bmatrix} \tag{45}$$

$$\hat{\mathbf{a}}_{\mathbf{r}} = \mathbf{r} + \mathbf{v}_{\mathbf{r}}\varepsilon + \tfrac{1}{2}\mathbf{a}_{\mathbf{r}}\varepsilon^2 + \tfrac{1}{6}\mathbf{j}_{\mathbf{r}}\varepsilon^3 + \tfrac{1}{24}\mathbf{s}_{\mathbf{r}}\varepsilon^4 \tag{46}$$

contains the position of the body point characterized by r and the corresponding values of the velocity, acceleration, jerk, and snap fields at that point.

Consequently, the homogeneous hypercomplex matrix $\hat{\mathbf{h}}$, defined by (39)–(40), with coefficients given in (43) and (44), provides a unified algebraic description of the rigid-body pose and its associated higher-order kinematic information.

V. Higher-Order Motions Of Serial Chains With Hypercomplex Algebra

Consider a spatial serial kinematic chain (Fig. 1), composed of rigid bodies $C_k, k = 0, \dots, m$. The motion of the rigid body $C_s, s = 1, \dots, m$, is described by the following POE (Product of Exponentials) kinematic mapping [6], [13]:

$$\begin{gathered} \mathbf{g}_s = f_s(\mathbf{q}_s) \\ = \exp({}^0\mathbf{S}_1 \mathrm{q}_1)\exp({}^1\mathbf{S}_2\mathrm{q}_2)\dots\exp({}^{s-1}\mathbf{S}_s\mathrm{q}_s)\,, \\ s = 1, \dots, m, m \in \mathbb{N}, \end{gathered} \tag{47}$$

where $\mathbf{q}_s = [q_1, q_2, \dots, q_s]^T$ denotes the vector of joint variables. In equation (47), ${}^{k-1}\mathbf{S}_k$, for $k = 1, \dots, s$, represent the screw coordinate vectors associated with body C_k, expressed in the reference frame attached to body C_{k-1}. The variables $q_k = q_k(t)$, $k = 1, \dots, s$, correspond to the associated joint coordinates [18], [22], [23].

The joint variables $q_k = q_k(t)$, defined for $t \in \mathbb{I} \subseteq \mathbb{R}$, are assumed to be four times continuously differentiable. Evaluating higher-order derivatives of the POE mapping using classical analytical methods involves complex recursive chain rules, nested Lie brackets, and rapidly growing expressions that are computationally inefficient. Numerical approximations are equally unsuited due to severe truncation errors. By transferring the joint variables into the proposed nilpotent hypercomplex domain, the exponential mappings algebraically absorb the derivative information up to snap, circumventing classical differentiation altogether. This result is stated formally in the following theorem.

Theorem 2. *The hyper-state of the rigid body C_s of the serial kinematic chain defined by the mapping (47) is derived from the hypercomplex homogeneous matrix*

$$\hat{\mathbf{h}}_s = \exp(\mathbf{J}_1^\wedge \Delta\breve{q}_1)\exp(\mathbf{J}_2^\wedge\Delta\breve{q}_2)\ \dots\ \exp(\mathbf{J}_s^\wedge\Delta\breve{q}_s), \tag{48}$$

where the matrices

$$\mathbf{J}_k^\wedge = \begin{bmatrix} (\mathbf{u}_k \times) & \mathbf{u}_{0k} \\ \mathbf{0}^T & 0 \end{bmatrix}. \tag{49}$$

$$\mathbf{u}_k = \mathbf{z}_{k-1}, \mathbf{u}_{0k} = \mathbf{p}_{k-1} \times \mathbf{z}_{k-1}, k = 1, \dots, s. \tag{50}$$

where $\mathbf{J}_k^\wedge$ is the homogeneous screw-generator matrix associated with the k -th joint axis, written in spatial screw-coordinate form. The vectors $\mathbf{u}_k$ and $\mathbf{u}_{0k}$denote, respectively, the direction and moment coordinates of the k-th joint axis. The vectors $\mathbf{z}_{k-1}$ and $\mathbf{p}_{k-1}$ are expressed in the fixed frame.

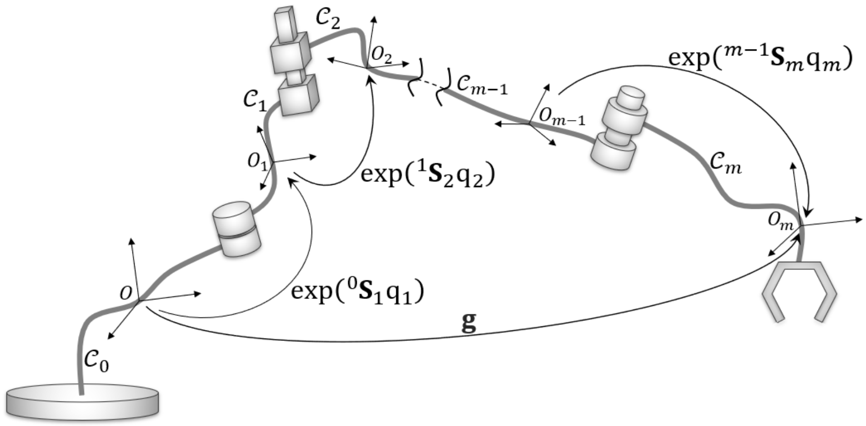

Figure 1. Spatial serial kinematic chain $\mathcal{C}_0, \dots, \mathcal{C}_m$ with attached reference frames and joint screw axes.

$$\Delta\breve{q}_k = \dot{q}_k\varepsilon + \frac{1}{2}\ddot{q}_k\varepsilon^2 + \frac{1}{6}\dddot{q}_k\varepsilon^3 + \frac{1}{24}q_k^{(4)}\varepsilon^4, k = 1, \dots, s, \quad (51)$$

denotes the hypercomplex increment associated with the k-th joint variable.

The hyper-state of the end-effector of the kinematic chain shown in Figure 1 is obtained by setting $s = m$ in Equation (48).

Proof: By applying the hypercomplex differential transform in Eq. 47, the following result is obtained:

$$\breve{\mathbf{g}}_s = \exp(^0\mathbf{S}_1\breve{q}_1)\exp(^1\mathbf{S}_2\breve{q}_2)\dots\exp(^{s-1}\mathbf{S}_s\breve{q}_s) \quad (52)$$

Moreover, the inverse of the homogeneous matrix $\mathbf{g}_s$ from Eq. (47) is given by

$$\mathbf{g}_s^{-1} = \exp(-^{s-1}\mathbf{S}_s q_s)\dots\exp(-^1\mathbf{S}_2 q_2)\exp(-^0\mathbf{S}_1 q_1) \quad (53)$$

Using the relation $\hat{\mathbf{h}}_s = \breve{\mathbf{g}}_s\mathbf{g}_s^{-1}$, Eq. (48) is obtained, where

$$\Delta\breve{q}_k = \breve{q}_k - q_k, \ k = 1, \dots, s. \quad (54)$$

In the derivation of this formula, the properties of the adjoint mapping in $SE(3)$ are employed [6], [7], [13], [22].

VI. Computational Algorithm and Numerical Example

In this section, the end-effector hyper-state of the KUKA LBR iiwa 7 R800 manipulator is evaluated using the algorithm below.

Algorithm:

Input: *spatial screw coordinates $\mathbf{u}_k, \mathbf{u}_{0k}, k = 1, \dots, 7$, and joint variables. The joint variables are chosen as the following periodic functions: $q_k(t) = \delta_k \sin t \cos t, k = 1, \dots, 7$, where $\delta_k = [(\pi/6), (\pi/4), \pi, \pi, (\pi/3), (\pi/3), (-\pi/3)]$,* [rad].

***For** $k = 1, \dots, 7$ **construct** $\Delta\breve{q}_k$ using (51), **construct** $\mathbf{J}_k^\wedge$ using (49)–(50), and **compute** $\exp(\mathbf{J}_k^\wedge\Delta\breve{q}_\mathbf{k})$.*

*Next **compute** $\hat{\mathbf{h}}_7$ from (48), with $s = 7$.*

Output: *extract $\widehat{\boldsymbol{\Phi}}$ and $\hat{\mathbf{a}}$ from $\hat{\mathbf{h}}_7$ using (40)–(44) and evaluate the coefficients at the selected time instant.*

The numerical computations were performed using MATLAB R2025b, and the results corresponding to $t = 0.7\pi$ are reported in Table I.

TABLE I. End-Effector Hyper-State Coefficients at $t = 0.7\pi$.

j	$\boldsymbol{\Phi}_j$ $[1/s^j]$	$\mathbf{a}_j[mm/s^j]$
1	$\begin{bmatrix} 0.0000 & 1.7673 & -0.1831 \\ -1.7673 & 0.0000 & 0.7012 \\ 0.1831 & -0.7012 & 0.0000 \end{bmatrix}$	$\begin{bmatrix} -104.7147 \\ -633.1116 \\ 42.0102 \end{bmatrix}$
2	$\begin{bmatrix} -3.1569 & -10.4905 & 3.1640 \\ 10.7473 & -3.6150 & -4.6457 \\ -0.6855 & 5.2929 & -0.5252 \end{bmatrix}$	$\begin{bmatrix} -1102.3410 \\ 4633.7610 \\ -12.8407 \end{bmatrix}$
3	$\begin{bmatrix} 57.3573 & -22.6716 & -37.8453 \\ 15.8928 & 66.7533 & -1.1788 \\ -10.8391 & -14.8594 & 11.5108 \end{bmatrix}$	$\begin{bmatrix} 34622.1900 \\ -7395.5170 \\ -6284.6980 \end{bmatrix}$
4	$\begin{bmatrix} -257.5302 & 393.9938 & 219.1441 \\ -290.9113 & -334.8075 & 153.9624 \\ 118.8654 & 57.0856 & -120.0204 \end{bmatrix}$	$\begin{bmatrix} -236158.9000 \\ -85392.0000 \\ 85789.6300 \end{bmatrix}$

As reported in Table I, the hyper-state formulation yields the exact coefficient pairs $(\boldsymbol{\Phi}_j, \mathbf{a}_j)$, $j = 1, \dots, 4$, reported here in rounded numerical form, for the end-effector of the KUKA LBR iiwa 7 R800 at $t = 0.7\pi$. These quantities are obtained from the normalized hypercomplex homogeneous matrix $\hat{\mathbf{h}}_s$ defined by (39)–(40), computed for the serial chain through the ordered product of local screw-generator exponentials in (48), with $\mathbf{J}_k^\wedge$ defined by (49)–(50) and $\Delta\breve{q}_k$ defined by (51). The coefficients are recovered from $\widehat{\boldsymbol{\Phi}}$ and $\hat{\mathbf{a}}$ by the factorial-scaled expansions (43)–(44), and represent, respectively, the invariant velocity, acceleration, jerk, and snap fields associated with the ε, ε^2, ε^3, and ε^4 terms. For the imposed trajectories $q_k(t) = \delta_k \sin t \cos t = (\delta_k/2)\sin 2t$, the j-th joint derivative has an amplitude proportional to $\delta_k 2^{j-1}$; therefore, the reported higher-order coefficients reflect both this analytical derivative scaling and the geometric coupling induced by the serial-chain screw generators. The archived MATLAB script [24] reproduces the complete computation by constructing the spatial screw coordinates $\mathbf{u}_k$ and $\mathbf{u}_{0k}$, forming the homogeneous screw-generator matrices $\mathbf{J}_k^\wedge$, propagating the product in (48), normalizing the resulting hypercomplex homogeneous matrix according to (39)–(40), and extracting the coefficient fields through (43)–(44). Thus, the numerical example does not approximate the end-effector derivatives by finite differences; it obtains them directly by nilpotent hypercomplex automatic differentiation, while preserving the coupled rotational–translational structure of the higher-order rigid-body motion.

VII. Conclusion

This paper introduced a fourth-order nilpotent hypercomplex automatic differentiation framework for robotic kinematics. By embedding derivative information directly into the algebraic variables, the method converts higher-order rigid-body kinematics into coefficient extraction, avoiding finite-difference approximation, explicit time differentiation, and recursive construction of higher-order fields.

The rigid-body hyper-state was formulated as a normalized hypercomplex homogeneous transformation on $SE(3)$, preserving the coupled rotational–translational structure of motion. For serial chains, the formulation leads to an ordered product of local screw-generator exponentials, from which the end-effector velocity, acceleration, jerk, and snap fields are obtained directly.

The KUKA LBR iiwa 7 R800 example confirms that the proposed algebraic construction produces the invariant coefficient pairs up to fourth order in a compact and structurally consistent computation. The result is an exact higher-order kinematic propagation within the nilpotent algebraic model, free from finite-difference truncation and step-size errors.

Future work will extend the framework to closed-loop and parallel robotic systems and exploit the fourth-order hyper-state in snap-bounded trajectory generation and vibration-sensitive motion planning.

Code Availability

The MATLAB source code used to generate the numerical results reported in this paper, together with the corresponding command-window output, is archived in Zenodo as a separate software record [24].

Acknowledgment

This research has been supported by the project new frontiers in adaptive modular robotics for patient-centred medical rehabilitation—ASKLEPIOS, funded by European Union—NextGenerationEU and Romanian Government, under National Recovery and Resilience Plan for Romania, contract no. 760071/23.05.2023, code CF 121/15.11.2022, with Romanian Ministry of Research, Innovation and Digitalization, within Component 9, investment I8.

References

[1] A. Griewank and A. Walther, Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, 2nd ed. Philadelphia, PA, USA: SIAM, 2008.

[2] U. Naumann, The Art of Differentiating Computer Programs: An Introduction to Algorithmic Differentiation. Philadelphia, PA, USA: SIAM, 2012.

[3] L. B. Rall, Ed., Automatic Differentiation: Techniques and Applications. Berlin, Germany: Springer, 1981.

[4] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, “Automatic differentiation in machine learning: A survey,” J. Mach. Learn. Res., vol. 18, no. 153, pp. 1–43, 2018. DOI: 10.48550/arXiv.1502.05767

[5] C. C. Margossian, “A review of automatic differentiation and its efficient implementation,” WIREs Data Mining Knowl. Discov., vol. 9, no. 4, Art. e1305, 2019. DOI: 10.1002/widm.1305

[6] K. M. Lynch and F. C. Park, Modern Robotics: Mechanics, Planning, and Control. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[7] L. Sciavicco and B. Siciliano, Modelling and Control of Robot Manipulators, 2nd ed. London, U.K.: Springer, 2012.

[8] R. Featherstone, Rigid Body Dynamics Algorithms. New York, NY, USA: Springer, 2008.

[9] J. B. Kuipers, Quaternions and Rotation Sequences. Princeton, NJ, USA: Princeton Univ. Press, 1999.

[10] L. Dorst, D. Fontijne, and S. Mann, Geometric Algebra for Computer Science. San Francisco, CA, USA: Morgan Kaufmann, 2007.

[11] R. D. Neidinger, “Introduction to automatic differentiation and MATLAB object-oriented programming,” SIAM Rev., vol. 52, no. 3, pp. 545–563, 2010. DOI: 10.1137/080743627

[12] J. A. Fike and J. J. Alonso, “The development of hyper-dual numbers for exact second-derivative calculations,” AIAA J., vol. 49, no. 2, pp. 387–399, 2011. DOI: 10.2514/1.J050977

[13] J. M. Selig, Geometric Fundamentals of Robotics. New York, NY, USA: Springer, 2005.

[14] I. L. Kantor and A. S. Solodovnikov, Hypercomplex Numbers: An Elementary Introduction to Algebras. New York, NY, USA: Springer, 1989.

[15] D. Condurache, "Higher-Order Relative Kinematics of Rigid Body and Multibody Systems. A Novel Approach with Real and Dual Lie Algebras", Mechanism and Machine Theory, vol. 176, 104999, 2022. DOI: 10.1016/j.mechmachtheory.2022.104999

[16] J. Angeles, "The application of dual algebra to kinematic analysis", Computational Methods, Mechanical Systems, vol. 161, pp. 3-31, 1998. DOI: 10.1007/978-3-662-03729-4_1

[17] J. A. Fike, J. J. Alonso, "Automatic Differentiation Through the Use of Hyper-Dual Numbers for Second Derivatives", in Recent Advances in Algorithmic Differentiation, edited by S. Forth, P. Hovland, E. Phipps, J. Utke, A. Walther, pp. 163-173, Berlin: Springer, 2012. DOI: 10.1007/978-3-642-30023-3_15

[18] D. Condurache, "Product of Exponential Formula of Multidual Quaternions and Higher-Order Kinematics", 2023 9th International Conference on Control, Decision, and Information Technologies (CoDIT), Rome, Italy, pp. 537-542, 2023. DOI: 10.1109/CoDIT58514.2023.10284240

[19] D. Condurache, "Higher-Order Kinematics of Lower-Pair Chains with Hyper-Multidual Algebra", Proceedings of the ASME 2022 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 7: 46th Mechanisms and Robotics Conference (MR). St. Louis, Missouri, USA. August 14–17, 2022. V007T07A073. ASME. DOI: 10.1115/DETC2022-88781

[20] A. Cohen, M. Shoham, "Hyper Dual Quaternions representation of rigid bodies kinematics", Mechanism and Machine Theory, vol. 150, 2020. DOI: 10.1016/j.mechmachtheory.2020.103861

[21] A. Cohen, M. Shoham, "Application of Hyper-Dual Numbers to Multibody Kinematics," ASME. J. Mechanisms Robotics, vol. 8, no. 1, 011015, 2016. DOI: 10.1115/1.4030588

[22] A. Müller, "An overview of formulae for the higher-order kinematics of lower-pair chains with applications in robotics and mechanism theory", Mechanism and Machine Theory, vol. 142, 103594, 2019. DOI: 10.1016/j.mechmachtheory.2019.103594

[23] D. Condurache, "An overview of higher-order kinematics of rigid body and multibody systems with nilpotent algebra ", Mechanism and Machine Theory, vol. 209, 105959, 2025. DOI: 10.1016/j.mechmachtheory.2025.105959.

[24] D. Condurache and M. Cojocari, "MATLAB Source Code for Fourth-Order MHM Hyper-State Computation of the KUKA LBR iiwa 7 R800", Zenodo, version 1.0.0, 2026. DOI: 10.5281/zenodo.20404299.