Automatic Differentiation with Hypercomplex Numbers for Higher-Order Motion Analysis

Daniel Condurache and Mihail Cojocari

***Abstract*— This paper presents an automatic differentiation framework based on finite nilpotent hypercomplex algebra for computing higher-order rigid-body motion fields. Derivative information is embedded into hypercomplex joint increments and propagated through the product-of-exponentials representation of a serial chain. The formulation is stated for an arbitrary finite order n, while the symbolic example is reported for n=3, yielding velocity, acceleration, and jerk fields. The method is applied to a selected link of the KUKA LBR iiwa 7 R800 manipulator. For the reported case, direct symbolic differentiation confirms the result exactly, with residuals simplifying identically to zero.**

Keywords— Lie group SE(3), automatic differentiation, nilpotent hypercomplex algebra, higher-order motion fields, serial chains.

I. INTRODUCTION

Automatic Differentiation (AD) computes derivatives by propagating derivative information through elementary algebraic operations. It avoids the truncation and step-size errors of finite differences and reduces the need for repeated symbolic differentiation of increasingly complex expressions [1]– [5]. These properties are relevant in robotic kinematics, where rigid-body motion must often be described beyond pose and velocity, including acceleration, jerk, and higher-order motion fields [6]– [8].

Hypercomplex and nilpotent algebraic structures provide a compact way to embed derivative information directly into computational variables. Dual numbers, hyper-dual numbers, quaternions, and related tools have been used in rigid-body kinematics, automatic differentiation, and higher-order derivative computation [9]– [14]. Higher-order rigid-body kinematics can also be formulated on the Lie group $SE(3)$ through invariant motion fields, represented at each order by a tensor-vector pair [15], [16]. Related hyper-dual, multidual, and product-of-exponentials formulations have been developed for higher-order differentiation and rigid-body kinematics [17]– [22], while a general nilpotent-algebraic framework was introduced in [23].

This paper applies this framework to serial kinematic chains. The derivatives of the joint variables are encoded into nilpotent hypercomplex increments and propagated through the product-of-exponentials representation. The normalized hypercomplex homogeneous transformation then contains, in its coefficients, the invariant motion-field quantities of the selected rigid body. Hence, velocity, acceleration, jerk, and higher-order fields are obtained by coefficient extraction, without finite-difference approximation or separate differentiation of the complete forward-kinematics expression at each order.

The formulation is stated for an arbitrary finite order n, while the symbolic expressions reported here are limited to $n = 3$ for compactness. The MATLAB implementation is order-parametric and can generate higher-order quantities by increasing the multidual order, subject to symbolic complexity. The method is tailored for serial chains by utilizing screw generators transferred to the fixed frame and hypercomplex joint increments. It has been implemented on a specific link of the KUKA LBR iiwa 7 R800 manipulator. Direct symbolic differentiation confirms the reported results exactly, with residuals simplifying identically to zero.

II. HIGHER-ORDER ACCELERATION FIELDS OF RIGID-BODY MOTION

Consider a rigid body motion in three-dimensional Euclidean space $\mathbb{E}^3$, represented by a one-parameter curve in the Lie group $SE(3)$. The corresponding homogeneous transformation is written as

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

where $\mathbf{R}(t) \in SO(3)$ is the rotation tensor and $\mathbf{r}_Q(t) \in \mathbb{R}^3$ is the position vector of a reference point Q attached to the rigid body. Let $n \geq 1$ be a fixed finite differentiation order. In what follows, the functions $\mathbf{R}(t)$ and $\mathbf{r}_Q(t)$ are assumed to be sufficiently differentiable up to order n on the time interval $t \in \mathbb{I} \subseteq \mathbb{R}$.

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

$$\mathbf{a}_{\mathbf{r}}^{(j)} = \mathbf{a}_j + \mathbf{\Phi}_j \mathbf{r},\ j = 1, \ldots, n. \tag{2}$$

Here, $\mathbf{\Phi}_j$ is the corresponding j-th order invariant tensor, while $\mathbf{a}_j$ is the corresponding j-th order invariant vector associated with the translational component of the rigid-body motion field [15]. These quantities are defined as follows:

$$\mathbf{\Phi}_j = \mathbf{R}^{(j)} \mathbf{R}^T,\ j = 1, \ldots, n, \tag{3}$$

$$\mathbf{a}_j = \mathbf{r}_Q^{(j)} - \mathbf{\Phi}_j \mathbf{r}_Q,\ j = 1, \ldots, n. \tag{4}$$

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).

The tensor $\mathbf{\Phi}_j$ describes the rotational contribution of the j-th order motion field, whereas $\mathbf{a}_j$ represents the corresponding invariant vector contribution. The vector $\mathbf{a}_j$ should not be interpreted as the direct j-th derivative of the position vector of the reference point [15], [22], [23]. It is obtained from $\mathbf{r}_Q^{(j)}$ after subtracting the rotational contribution $\mathbf{\Phi}_j \mathbf{r}_Q$. Therefore, the pair $(\mathbf{\Phi}_j, \mathbf{a}_j)$ gives the invariant description of the complete j -th order vector field over the rigid body.

The complete set of higher-order motion fields up to order n is denoted by

$$\mathcal{H}_n(\mathbf{g}) = \{(\mathbf{\Phi}_j, \mathbf{a}_j) \mid j = 1,\ldots,n\}. \quad (5)$$

The recursive construction of these invariant quantities is discussed in [15], while their representation by nilpotent multidual algebra is developed in [23]. In the present paper, the fields are not computed recursively. Instead, they are recovered simultaneously from the coefficients of a hypercomplex homogeneous transformation.

The formulation above is kept for an arbitrary finite order n. In the symbolic application reported in this paper, the selected order is $n = 3$. Consequently, the explicitly reported fields are

$$\mathcal{H}_3(\mathbf{g}) = \{(\mathbf{\Phi}_1, \mathbf{a}_1), (\mathbf{\Phi}_2, \mathbf{a}_2), (\mathbf{\Phi}_3, \mathbf{a}_3)\}. \quad (6)$$

These fields correspond respectively to the velocity, acceleration, and jerk fields.

This formulation is stated at the level of an arbitrary rigid-body motion and an arbitrary finite order n. The subsequent sections specialize it to a serial manipulator by using a hypercomplex homogeneous representation and screw generators transported to the fixed frame. In the symbolic example, the computation is performed for a selected link of the KUKA LBR iiwa 7 R800 manipulator. However, the mathematical construction is not restricted to that link or to the third order; the third-order case is used because it provides compact symbolic expressions suitable for explicit reporting.

III. Hypercomplex Nilpotent Differential Transform

The complete construction of multidual numbers, multidual vectors, multidual tensors, and the associated differential transform is given in [23]. In the present paper, only the algebraic notation required for the hypercomplex homogeneous-matrix formulation is recalled. The notation is kept consistent with the previous section: n denotes a fixed finite differentiation order, while j denotes the extracted coefficient order.

Let $n \geq 1$ be fixed. The finite nilpotent hypercomplex algebra used in this work is defined by

$$\widehat{\mathbb{R}} = \mathbb{R} \oplus \varepsilon\mathbb{R} \oplus \varepsilon^2\mathbb{R} \oplus \ldots \oplus \varepsilon^n\mathbb{R}, \varepsilon \neq 0, \varepsilon^{n+1} = 0. \quad (7)$$

A generic element $\hat{x} \in \widehat{\mathbb{R}}$ can be written in the form

$$\hat{x} = \sum_{j=0}^{n} x_j \varepsilon^j, \ x_0 = x, \ x_j \in \mathbb{R}, \ j = 0,\ldots,n. \quad (8)$$

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

$$Re(\hat{x}) = x, \quad (9)$$

while its nilpotent part is

$$Mu(\hat{x}) = \Delta\hat{x} = \hat{x} - x = \sum_{j=1}^{n} x_j \varepsilon^j. \quad (10)$$

If $[\varepsilon^j]$ denotes the coefficient extraction operator, then

$$[\varepsilon^j]\hat{x} = x_j, j = 0,\ldots,n. \quad (11)$$

The multiplication of the nilpotent basis elements of $\widehat{\mathbb{R}}$ is governed by

$$\varepsilon^k \varepsilon^p = \varepsilon^p \varepsilon^k = \begin{cases} \varepsilon^{k+p}, & k+p \leq n, \\ 0, & k+p \geq n+1. \end{cases} \quad (12)$$

Thus, all algebraic expressions are automatically truncated at order n by the nilpotency condition. This finite-dimensional structure is the algebraic mechanism used to encode a finite number of derivatives in a single hypercomplex quantity.

Let $f: \mathbb{I} \subseteq \mathbb{R} \to \mathbb{R}, f = f(t)$, be sufficiently differentiable up to order n. The hypercomplex differential transform of f is defined by

$$\breve{f}(t) = f(t) + \varepsilon\dot{f}(t) + \frac{\varepsilon^2}{2!}\ddot{f}(t) + \ldots + \frac{\varepsilon^n}{n!} f^{(n)}(t). \quad (13)$$

Equivalently,

$$\breve{f}(t) = \sum_{j=0}^{n} \frac{\varepsilon^j}{j!} f^{(j)}(t). \quad (14)$$

Therefore, the derivative of order j is recovered from the coefficient of ε^j by

$$f^{(j)}(t) = j!\,[\varepsilon^j]\breve{f}(t), \ j = 0,\ldots,n. \quad (15)$$

The same transform is applied component-wise to vector and matrix functions. Thus, for a vector function $\mathbf{a}(t)$,

$$\breve{\mathbf{a}}(t) = \sum_{j=0}^{n} \frac{\varepsilon^j}{j!} \mathbf{a}^{(j)}(t), \quad (16)$$

and a matrix function $\mathbf{A}(t)$,

$$\breve{\mathbf{A}}(t) = \sum_{j=0}^{n} \frac{\varepsilon^j}{j!} \mathbf{A}^{(j)}(t). \quad (17)$$

The transform is compatible with the standard algebraic operations used in kinematic mappings. For functions that are sufficiently differentiable and allow these operations,

$$\breve{(f+g)} = \breve{f} + \breve{g}, \quad (18)$$

$$\breve{(fg)} = \breve{f}\breve{g}, \quad (19)$$

$$\breve{(\lambda f)} = \lambda\breve{f}, \ \forall\lambda \in \mathbb{R}, \quad (20)$$

$$\breve{(f \circ \alpha)} = f(\breve{\alpha}), \alpha \in C^n(\mathbb{I}). \quad (21)$$

Consequently, derivative information is propagated through algebraic operations in $\widehat{\mathbb{R}}$, without finite-difference approximations and without separately differentiating the final expressions.

For the symbolic application reported in this paper, the selected order is $n = 3$. In this case, Eq. (13) becomes

$$\breve{f}(t) = f(t) + \varepsilon\dot{f}(t) + \frac{\varepsilon^2}{2}\ddot{f}(t) + \frac{\varepsilon^3}{6}\dddot{f}(t), \varepsilon^4 = 0. \quad (22)$$

Thus, the third-order hypercomplex representation contains the value and the first three derivatives of each transformed quantity. This is the finite-order algebraic tool used in the subsequent homogeneous-matrix formulation.

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

Consider the rigid-body motion introduced in Eq. (1). The real pose of the rigid body is described by the homogeneous transformation $\mathbf{g}(t)$, while the hypercomplex differential transform introduced in Section III gives the corresponding hypercomplex homogeneous matrix

$$\breve{\mathbf{g}}(t) = \begin{bmatrix} \breve{\mathbf{R}}(t) & \breve{\mathbf{r}}_Q(t) \\ \mathbf{0} & 1 \end{bmatrix}. \tag{23}$$

Here, $\breve{\mathbf{R}}(t)$ and $\breve{\mathbf{r}}_Q(t)$ are obtained by applying the hypercomplex differential transform component-wise to $\mathbf{R}(t)$ and $\mathbf{r}_Q(t)$, respectively. The matrix $\breve{\mathbf{g}}(t)$ contains the pose and its time derivatives up to the selected finite order n. In the present formulation, the invariant motion-field quantities are obtained after normalization with respect to the real pose. For compactness, the explicit time argument is omitted in the following equations.

The normalized homogeneous hypercomplex matrix is defined as

$$\hat{\mathbf{h}} = \breve{\mathbf{g}}\mathbf{g}^{-1}. \tag{24}$$

The inverse homogeneous transformation is

$$\mathbf{g}^{-1} = \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{r}_Q \\ \mathbf{0} & 1 \end{bmatrix}. \tag{25}$$

Hence, one obtains

$$\hat{\mathbf{h}} = \begin{bmatrix} \widehat{\mathbf{\Phi}} & \hat{\mathbf{a}} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{26}$$

where

$$\widehat{\mathbf{\Phi}} = \breve{\mathbf{R}}\mathbf{R}^T, \tag{27}$$

and

$$\hat{\mathbf{a}} = \left(\breve{\mathbf{r}}_Q - \breve{\mathbf{R}}\mathbf{R}^{\mathrm{T}}\mathbf{r}_Q\right) = \breve{\mathbf{r}}_Q - \widehat{\mathbf{\Phi}}\mathbf{r}_Q. \tag{28}$$

Using the definitions introduced in Section II, the hypercomplex tensor $\widehat{\mathbf{\Phi}}$ and the hypercomplex vector $\hat{\mathbf{a}}$ have the coefficient expansions

$$\widehat{\mathbf{\Phi}} = \textstyle\sum_{j=0}^{n} \frac{\varepsilon^j}{j!}\mathbf{\Phi}_j\,,\ \ \mathbf{\Phi}_0 = \mathbf{I}, \tag{29}$$

$$\hat{\mathbf{a}} = \textstyle\sum_{j=0}^{n} \frac{\varepsilon^j}{j!}\mathbf{a}_j\,,\ \ \mathbf{a}_0 = \mathbf{0}. \tag{30}$$

Consequently, the invariant motion-field quantities are recovered by coefficient extraction as follows:

$$\mathbf{\Phi}_j = j!\,[\varepsilon^j]\widehat{\mathbf{\Phi}}\,,\ j = 1,\dots,n, \tag{31}$$

$$\mathbf{a}_j = j!\,[\varepsilon^j]\hat{\mathbf{a}},\ j = 1,\dots,n. \tag{32}$$

For an arbitrary point of the rigid body, characterized by the position vector $\mathbf{r}$, the associated hypercomplex point-field representation is obtained as

$$\begin{bmatrix} \hat{\mathbf{y}}_{\mathbf{r}} \\ 1 \end{bmatrix} = \hat{\mathbf{h}}\begin{bmatrix} \mathbf{r} \\ 1 \end{bmatrix}. \tag{33}$$

Therefore,

$$\hat{\mathbf{y}}_{\mathbf{r}} = \widehat{\mathbf{\Phi}}\,\mathbf{r} + \hat{\mathbf{a}}. \tag{34}$$

Using Eqs. (29) and (30), this becomes

$$\hat{\mathbf{y}}_{\mathbf{r}} = \mathbf{r} + \textstyle\sum_{j=1}^{n} \frac{\varepsilon^j}{j!}\left(\mathbf{a}_j + \mathbf{\Phi}_j\mathbf{r}\right). \tag{35}$$

From Eq. (2), the term between parentheses is precisely the value of the j-th order motion field at the point characterized by $\mathbf{r}$. Hence,

$$\hat{\mathbf{y}}_{\mathbf{r}} = \mathbf{r} + \textstyle\sum_{j=1}^{n} \frac{\varepsilon^j}{j!}\mathbf{a}_{\mathbf{r}}^{(j)}. \tag{36}$$

Thus, the normalized hypercomplex matrix $\hat{\mathbf{h}}$ contains the invariant motion fields of the rigid body up to the selected finite order n. The real pose is represented by $\mathbf{g}$, while $\hat{\mathbf{h}}$ contains the associated invariant field information. Therefore, the finite-order hyper-state may be represented by the pair $(\mathbf{g}, \hat{\mathbf{h}})$, or equivalently by the hypercomplex homogeneous transformation $\breve{\mathbf{g}}$ together with its normalized form.

For the symbolic application reported in this paper, the selected order is $n = 3$. In this case, Eqs. (29) and (30) become

$$\widehat{\mathbf{\Phi}} = \mathbf{I} + \varepsilon\mathbf{\Phi}_1 + \frac{\varepsilon^2}{2}\mathbf{\Phi}_2 + \frac{\varepsilon^3}{6}\mathbf{\Phi}_3, \tag{37}$$

$$\hat{\mathbf{a}} = \varepsilon\mathbf{a}_1 + \frac{\varepsilon^2}{2}\mathbf{a}_2 + \frac{\varepsilon^3}{6}\mathbf{a}_3. \tag{38}$$

Accordingly, the point-field representation is

$$\hat{\mathbf{y}}_{\mathbf{r}} = \mathbf{r} + \varepsilon\mathbf{a}_{\mathbf{r}}^{(1)} + \frac{\varepsilon^2}{2}\mathbf{a}_{\mathbf{r}}^{(2)} + \frac{\varepsilon^3}{6}\mathbf{a}_{\mathbf{r}}^{(3)}. \tag{39}$$

The terms multiplying ε, $\varepsilon^2/2$, and $\varepsilon^3/6$ correspond respectively to the velocity, acceleration, and jerk fields of the point characterized by $\mathbf{r}$. regardless of how the rigid-body motion is produced.

V. Higher-Order Motion Fields of Serial Chains Using Hypercomplex Algebra

Consider a spatial serial kinematic chain composed of rigid bodies $C_0, C_1, \dots, C_m$, where C_0 denotes the fixed base. Let C_s, $s = 1, \dots, m$, be the selected rigid body of the chain. Its pose with respect to the fixed frame is described by the product-of-exponentials mapping

$$\begin{gathered} \mathbf{g}_s = f_s(\mathbf{q}_s) \\ = \exp({}^0\mathbf{S}_1^{\wedge}q_1)\exp({}^1\mathbf{S}_2^{\wedge}q_2)\dots\exp({}^{s-1}\mathbf{S}_s^{\wedge}q_s)\,, \\ s = 1, \dots, m. \end{gathered} \tag{40}$$

Here,

$$\mathbf{q}_s = [q_1, q_2, \ldots, q_s]^T \tag{41}$$

is the vector of joint variables associated with the motion of C_s. The vector ${}^{k-1}\mathbf{S}_k$ denotes the screw coordinate vector of joint k, expressed in the reference frame attached to C_{k-1}, while ${}^{k-1}\mathbf{S}_k^{\wedge} \in se(3)$ denotes its corresponding matrix representation. The joint variables $q_k = q_k(t), k = 1, \ldots, s$, are assumed to be sufficiently differentiable up to the selected finite order n.

For compactness, we define the partial forward kinematic map

$$\begin{gathered}\mathbf{f}_0 = \mathbf{I}_4,\\ \mathbf{f}_{k-1} = \exp({}^0\mathbf{S}_1^{\wedge} q_1) \ldots \exp({}^{k-2}\mathbf{S}_{k-1}^{\wedge} q_{k-1}),\\ k = 2, \ldots, s.\end{gathered} \tag{42}$$

The spatial screw vector associated with joint k, expressed in the fixed frame, is obtained through the adjoint transformation

$$\mathbf{J}_k = \mathrm{Ad}_{\mathbf{f}_{k-1}}\, {}^{k-1}\mathbf{S}_k,\ k = 1, \ldots, s, \tag{43}$$

with $\mathbf{J}_1 = {}^0\mathbf{S}_1$. The quantities $\mathbf{J}_k$ are used here as screw generators transported to the fixed frame. Although they may be arranged column-wise as a screw Jacobian, the present construction does not rely on a velocity-level geometric Jacobian relation. If

$$\mathbf{J}_k = \begin{bmatrix}\mathbf{u}_k\\ \mathbf{u}_{0k}\end{bmatrix},\ k = 1, \ldots, s, \tag{44}$$

then the corresponding matrix representation in $se(3)$ is

$$\mathbf{J}_k^{\wedge} = \begin{bmatrix}\tilde{\mathbf{u}}_k & \mathbf{u}_{0k}\\ \mathbf{0}^T & 0\end{bmatrix},\ k = 1, \ldots, s, \tag{45}$$

where $\tilde{\mathbf{u}}_k$ is the skew-symmetric matrix associated with $\mathbf{u}_k$.

The hypercomplex joint increment associated with the k-th joint is defined as

$$\Delta\breve{q}_k = \breve{q}_k - q_k = \sum_{j=1}^{n} \frac{\varepsilon^j}{j!} q_k^{(j)},\ k = 1, \ldots, s,\ \varepsilon^{n+1} = 0. \tag{46}$$

The zero-order term is not included in $\Delta\breve{q}_k$, because the real configuration is already contained in the real pose $\mathbf{g}_s$. Therefore, only the derivative information of the joint variables is carried by the nilpotent part.

For the symbolic application reported in this paper, the selected order is $n = 3$. In this case,

$$\Delta\breve{q}_k = \dot{q}_k\varepsilon + \frac{1}{2}\ddot{q}_k\varepsilon^2 + \frac{1}{6}\dddot{q}_k\varepsilon^3,\ k = 1, \ldots, s,\ \varepsilon^4 = 0. \tag{47}$$

The normalized hypercomplex homogeneous matrix associated with the selected body C_s is then obtained as

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

Equation (48) is the serial-chain specialization of the normalized hypercomplex matrix introduced in Section IV. It combines the screw generators transported to the fixed frame with the hypercomplex joint increments. The end-effector case is obtained by setting $s = m$, while for an intermediate rigid body only the first s active joints are involved.

Applying the hypercomplex differential transform to the POE mapping in Eq. (40) gives

$$\breve{\mathbf{g}}_s = \exp({}^0\mathbf{S}_1^{\wedge}\breve{q}_1)\exp({}^1\mathbf{S}_2^{\wedge}\breve{q}_2) \ldots \exp({}^{s-1}\mathbf{S}_s^{\wedge}\breve{q}_s), \tag{49}$$

where

$$\breve{q}_k = q_k + \Delta\breve{q}_k,\ k = 1, \ldots, s. \tag{50}$$

The inverse of the real transformation $\mathbf{g}_s$ is

$$\mathbf{g}_s^{-1} = \exp(-{}^{s-1}\mathbf{S}_s^{\wedge} q_s) \ldots \exp(-{}^1\mathbf{S}_2^{\wedge} q_2)\exp(-{}^0\mathbf{S}_1^{\wedge} q_1), \tag{51}$$

Using

$$\hat{\mathbf{h}}_s = \breve{\mathbf{g}}_s \mathbf{g}_s^{-1}, \tag{52}$$

and the adjoint transformation properties in $SE(3)$, Eq. (48) follows. Thus, the higher-order derivative information is propagated algebraically through the ordered product of hypercomplex screw exponentials, without explicitly differentiating the complete POE mapping.

The matrix $\hat{\mathbf{h}}_s$ has the block form

$$\hat{\mathbf{h}}_s = \begin{bmatrix}\hat{\boldsymbol{\Phi}}_s & \hat{\mathbf{a}}_s\\ \mathbf{0} & 1\end{bmatrix}, \tag{53}$$

Consequently,

$$\hat{\boldsymbol{\Phi}}_s = \sum_{j=0}^{n} \frac{\varepsilon^j}{j!} \boldsymbol{\Phi}_{s,j},\ \boldsymbol{\Phi}_{s,0} = \mathbf{I}, \tag{54}$$

$$\hat{\mathbf{a}}_s = \sum_{j=0}^{n} \frac{\varepsilon^j}{j!} \mathbf{a}_{s,j},\ \mathbf{a}_{s,0} = \mathbf{0}. \tag{55}$$

The invariant motion-field quantities of body C_s are recovered by coefficient extraction:

$$\boldsymbol{\Phi}_{s,j} = j!\,[\varepsilon^j]\hat{\boldsymbol{\Phi}}_s,\ j = 1, \ldots, n, \tag{56}$$

$$\mathbf{a}_{s,j} = j!\,[\varepsilon^j]\hat{\mathbf{a}}_s,\ j = 1, \ldots, n. \tag{57}$$

For the selected rigid body C_s, the j-th order motion field is evaluated at an arbitrary point characterized by the position vector $\mathbf{r}$ as

$$\mathbf{a}_{s,\mathbf{r}}^{(j)} = \mathbf{a}_{s,j} + \boldsymbol{\Phi}_{s,j}\mathbf{r},\ j = 1, \ldots, n. \tag{58}$$

For $n = 3$, the normalized hypercomplex representation of this point becomes

$$\hat{\mathbf{y}}_{s,\mathbf{r}} = \mathbf{r} + \varepsilon\mathbf{a}_{s,\mathbf{r}}^{(1)} + \frac{\varepsilon^2}{2}\mathbf{a}_{s,\mathbf{r}}^{(2)} + \frac{\varepsilon^3}{6}\mathbf{a}_{s,\mathbf{r}}^{(3)}. \tag{59}$$

The terms multiplying ε, $\varepsilon^2/2$, and $\varepsilon^3/6$ are the values of the velocity, acceleration, and jerk fields at the point characterized by $\mathbf{r}$ on the selected rigid body C_s.

VI. Computational Algorithm and Symbolic Example

This section presents the symbolic computational procedure used to determine the higher-order motion fields of the KUKA LBR iiwa 7 R800 manipulator. The procedure follows directly from the hypercomplex formulation of the serial-chain motion introduced in Section V. The index s denotes the selected rigid body/link of the manipulator, while j denotes the derivative order. For the KUKA LBR iiwa 7 R800 manipulator, $s \in \{1, \ldots, 7\}$. The symbolic example reported below is given for $s = 2$ and $n = 3$, which yields the velocity, acceleration, and jerk fields.

The computation is symbolic. No numerical trajectory is prescribed for the joint variables. The joint coordinates q_k and their time derivatives $\dot{q}_k$, $\ddot{q}_k$ and $\dddot{q}_k$ are treated as independent symbolic variables at a generic time instant. Consequently, the obtained expressions are closed-form exact algebraic expressions and can be evaluated afterward for any admissible differentiable motion law. The MATLAB implementation is order-parametric; the restriction to $n = 3$ is used here only to keep the reported symbolic expressions compact.

The computational procedure is the following.

Algorithm 1. Symbolic computation of higher-order motion fields

Input: *selected link* s, *multidual order* n, *DH parameters of the KUKA LBR iiwa 7 R800, and symbolic joint variables* $q_k(t)$, $k = 1, \ldots, s$.

Step 1. *Construct the homogeneous transformation matrices of the serial chain using the standard DH model.*

Step 2. *Determine the screw generators of the active joints and transport them to the fixed frame according to Eqs. (43)–(45).*

Step 3. *For each active joint, construct the hypercomplex joint increment* $\Delta\breve{q}_k$ *using Eq. (46), or Eq. (47) for* $n = 3$.

Step 4. *Build the ordered product in Eq. (48).*

Step 5. *Extract the block components according to Eq. (53).*

Step 6. *Recover the invariant motion-field quantities using Eqs. (56) and (57).*

Step 7. *Validate the symbolic result by checking the structural identities of the screw axes and of the rotational hypercomplex matrix. An additional independent check is performed by differentiating the standard forward-kinematics map of the selected link and comparing the result with the fields obtained by the hypercomplex procedure.*

For the reported example, the active variables are q_1, q_2 and their derivatives up to order three. The geometric parameters are expressed in millimetres. The compact notation

$$\mathrm{c}_k = \cos q_k, \mathrm{s}_k = \sin q_k \tag{60}$$

is used in the symbolic expressions. For compactness, the selected-link subscript is omitted in the following coefficients, i.e. $\boldsymbol{\Phi}_j = \boldsymbol{\Phi}_{s,j}$ and $\mathbf{a}_j = \mathbf{a}_{s,j}$, with $s = 2$.

For $s = 2$ and $n = 3$, the first-order field coefficient is

$$\boldsymbol{\Phi}_1 = \begin{bmatrix} 0 & -\dot{q}_1 & \mathrm{c}_1\dot{q}_2 \\ \dot{q}_1 & 0 & \mathrm{s}_1\dot{q}_2 \\ -\mathrm{c}_1\dot{q}_2 & -\mathrm{s}_1\dot{q}_2 & 0 \end{bmatrix} \tag{61}$$

$$\mathbf{a}_1 = \begin{bmatrix} -380\mathrm{c}_1\dot{q}_2 \\ -380\mathrm{s}_1\dot{q}_2 \\ 0 \end{bmatrix} \tag{62}$$

The second-order field coefficient is

$$\boldsymbol{\Phi}_2 = \begin{bmatrix} -\mathrm{c}_1^2\dot{q}_2^2 - \dot{q}_1^2 & -\mathrm{c}_1\mathrm{s}_1\dot{q}_2^2 - \ddot{q}_1 & \mathrm{c}_1\ddot{q}_2 - 2\mathrm{s}_1\dot{q}_1\dot{q}_2 \\ -\mathrm{c}_1\mathrm{s}_1\dot{q}_2^2 + \ddot{q}_1 & -\dot{q}_1^2 - \mathrm{s}_1^2\dot{q}_2^2 & \mathrm{s}_1\ddot{q}_2 + 2\mathrm{c}_1\dot{q}_1\dot{q}_2 \\ -\mathrm{c}_1\ddot{q}_2 & -\mathrm{s}_1\ddot{q}_2 & -\dot{q}_2^2 \end{bmatrix} \tag{63}$$

$$\mathbf{a}_2 = \begin{bmatrix} 760\mathrm{s}_1\dot{q}_1\dot{q}_2 - 380\mathrm{c}_1\ddot{q}_2 \\ -380\mathrm{s}_1\ddot{q}_2 - 760\mathrm{c}_1\dot{q}_1\dot{q}_2 \\ 380\dot{q}_2^2 \end{bmatrix} \tag{64}$$

The third-order field coefficient is

$$\boldsymbol{\Phi}_3 = \begin{bmatrix} \boldsymbol{\Phi}_{3,11} & \boldsymbol{\Phi}_{3,12} & \boldsymbol{\Phi}_{3,13} \\ \boldsymbol{\Phi}_{3,21} & \boldsymbol{\Phi}_{3,22} & \boldsymbol{\Phi}_{3,23} \\ \boldsymbol{\Phi}_{3,31} & \boldsymbol{\Phi}_{3,32} & \boldsymbol{\Phi}_{3,33} \end{bmatrix} \tag{65}$$

where

$$\boldsymbol{\Phi}_{3,11} = -3\mathrm{c}_1^2\dot{q}_2\ddot{q}_2 + 3\mathrm{s}_1\mathrm{c}_1\dot{q}_1\dot{q}_2^2 - 3\dot{q}_1\ddot{q}_1 \tag{66}$$

$$\boldsymbol{\Phi}_{3,12} = \dot{q}_1^3 + 3\dot{q}_1\dot{q}_2^2\mathrm{s}_1^2 - 3\mathrm{c}_1\mathrm{s}_1\dot{q}_2\ddot{q}_2 - \dddot{q}_1 \tag{67}$$

$$\begin{aligned} \boldsymbol{\Phi}_{3,13} = {} & -3\mathrm{c}_1\dot{q}_1^2\dot{q}_2 - 3\mathrm{s}_1\dot{q}_1\ddot{q}_2 \\ & -\mathrm{c}_1\dot{q}_2^3 - 3\mathrm{s}_1\ddot{q}_1\dot{q}_2 + \mathrm{c}_1\dddot{q}_2 \end{aligned} \tag{68}$$

$$\boldsymbol{\Phi}_{3,21} = -3\mathrm{c}_1^2\dot{q}_1\dot{q}_2^2 - 3\mathrm{s}_1\mathrm{c}_1\dot{q}_2\ddot{q}_2 - \dot{q}_1^3 + \dddot{q}_1 \tag{69}$$

$$\boldsymbol{\Phi}_{3,22} = -3\mathrm{c}_1\mathrm{s}_1\dot{q}_1\dot{q}_2^2 - 3\mathrm{s}_1^2\dot{q}_2\ddot{q}_2 - 3\dot{q}_1\ddot{q}_1 \tag{70}$$

$$\begin{aligned} \boldsymbol{\Phi}_{3,23} = {} & -3\mathrm{s}_1\dot{q}_1^2\dot{q}_2 + 3\mathrm{c}_1\dot{q}_1\ddot{q}_2 \\ & -\mathrm{s}_1\dot{q}_2^3 + 3\mathrm{c}_1\ddot{q}_1\dot{q}_2 + \mathrm{s}_1\dddot{q}_2 \end{aligned} \tag{71}$$

$$\boldsymbol{\Phi}_{3,31} = \mathrm{c}_1\dot{q}_2^3 - \mathrm{c}_1\dddot{q}_2 \tag{72}$$

$$\boldsymbol{\Phi}_{3,32} = \mathrm{s}_1\dot{q}_2^3 - \mathrm{s}_1\dddot{q}_2 \tag{73}$$

$$\boldsymbol{\Phi}_{3,33} = -3\dot{q}_2\ddot{q}_2 \tag{74}$$

The corresponding third-order vector coefficient is

$$\mathbf{a}_3 = \begin{bmatrix} \mathbf{a}_{3,1} \\ \mathbf{a}_{3,2} \\ \mathbf{a}_{3,3} \end{bmatrix} \tag{75}$$

with

$$\begin{aligned} \mathbf{a}_{3,1} = {} & 1140\mathrm{c}_1\dot{q}_1^2\dot{q}_2 + 1140\mathrm{s}_1\dot{q}_1\ddot{q}_2 + 380\mathrm{c}_1\dot{q}_2^3 \\ & +1140\mathrm{s}_1\ddot{q}_1\dot{q}_2 - 380\mathrm{c}_1\dddot{q}_2 \end{aligned} \tag{76}$$

$$\begin{aligned} \mathbf{a}_{3,2} = {} & 1140\mathrm{s}_1\dot{q}_1^2\dot{q}_2 - 1140\mathrm{c}_1\dot{q}_1\ddot{q}_2 + 380\mathrm{s}_1\dot{q}_2^3 \\ & -1140\mathrm{c}_1\ddot{q}_1\dot{q}_2 - 380\mathrm{s}_1\dddot{q}_2 \end{aligned} \tag{77}$$

$$\mathbf{a}_{3,3} = 1140\dot{q}_2\ddot{q}_2 \tag{78}$$

These expressions show that the proposed hypercomplex automatic-differentiation procedure generates the complete higher-order motion fields directly in symbolic form. The result is not restricted to a prescribed numerical trajectory; the obtained fields remain valid for arbitrary differentiable functions $q_k(t)$.

The symbolic implementation also performs consistency checks. The screw-axis unit-norm residuals, the screw-axis orthogonality residuals, the first-order skew-symmetry residual of $\boldsymbol{\Phi}_1$, and the hypercomplex rotational orthogonality residuals are identically zero. In addition, the fields obtained from the hypercomplex computation are compared with those obtained by direct symbolic differentiation of the standard forward-kinematics map of the selected link. The direct reference fields are obtained from

$$\mathbf{\Phi}_j^{dir} = \mathbf{R}^{(j)}\mathbf{R}^T, \quad \mathbf{a}_j^{dir} = \mathbf{r}_Q^{(j)} - \mathbf{\Phi}_j^{dir}\mathbf{r}_Q. \qquad (79)$$

For $j = 1,2,3,$ the residuals $\mathbf{\Phi}_j - \mathbf{\Phi}_j^{dir}$ and $\mathbf{a}_j - \mathbf{a}_j^{dir}$ simplify identically to zero. This confirms that the hypercomplex formulation produces the same higher-order motion fields as direct symbolic differentiation, while obtaining all derivative orders through a single algebraic computation.

VII. Conclusion

This paper presented a nilpotent hypercomplex automatic-differentiation formulation for higher-order rigid-body motion fields. The method represents the finite-order hyper-state by a normalized hypercomplex homogeneous transformation, whose coefficients directly provide the invariant tensor-vector quantities at each differentiation order. The formulation was stated for an arbitrary finite order n and specialized to serial kinematic chains by using screw generators transported to the fixed frame and hypercomplex joint increments.

The symbolic case study was carried out for a selected link of the KUKA LBR iiwa 7 R800 manipulator, with n=3. The obtained closed-form expressions give the velocity, acceleration, and jerk fields. The third-order case was selected only for compact reporting; the mathematical model and MATLAB implementation remain order-parametric.

The results were verified by independent direct symbolic differentiation of the standard forward-kinematics map. For the reported orders, all residuals simplify identically to zero. This confirms that the proposed procedure gives exact algebraic higher-order motion fields, without finite-difference approximation and without separately differentiating the complete product-of-exponentials expression for each order.

Future work will address numerical implementations for snap and higher-order derivatives, focusing on computational efficiency, structural consistency, and extension to broader serial multibody and robotic systems.

Code Availability

The MATLAB source code and command-window output supporting the reported symbolic results are archived in Zenodo [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.

[5] C. C. Margossian, “A review of automatic differentiation and its efficient implementation,” WIREs Data Mining Knowl. Discov., vol. 9, no. 4, Art. no. 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.

[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 in 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", in Proc. ASME 2022 Int. Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE), St. Louis, MO, USA, Aug. 14–17, 2022, Art. no. V007T07A073, 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, 01101, 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, M. Cojocari, "MATLAB Source Code for Symbolic Hyper-State Computation of Higher-Order Motion Fields of the KUKA LBR iiwa 7 R800", Zenodo, version 1.0.0, 2026, DOI: 10.5281/zenodo.20402025.