A Coordinate-Invariant Local Representation of Motion and Force Trajectories for Analysis and Generalization Across Coordinate Systems

Arno Verduyn[1,2,3,⋆][0000−0002−5073−3881],
Erwin Aertbeliën[1,2,3][0000−0002−4514−0934],
Maxim Vochten[4][0000−0001−5070−846X], and
Joris De Schutter[2][0000−0001−9619−5815]

[1] These authors contributed equally to this work.
[2] Department of Mechanical Engineering, KU Leuven, Leuven, Belgium
[3] Flanders Make at KU Leuven, Leuven, Belgium.
[4] Department of Mechanics, Royal Military Academy, Brussels, Belgium.

Abstract. Analyzing the trajectories of rigid bodies and of interaction forces is essential for a wide range of tasks in robotics, including trajectory segmentation, recognition, and prediction. A key challenge in these tasks is that the results often depend on the choice of coordinate system in which the trajectory is expressed, thereby complicating trajectory analysis and comparison across different coordinate systems. This challenge motivates the use of coordinate-invariant trajectory representations. However, coordinate-invariant representations often suffer from sensitivity to measurement noise near singularities in the representation, where the representation is not uniquely defined. To address this limitation, this paper provides a focused and formalized treatment of a novel invariant trajectory representation recently introduced in preprint form. Specifically, we present it here as the Dual-Upper-Triangular Invariant Representation (DUTIR), formally prove its coordinate invariance, and numerically demonstrate its reduced sensitivity to noise near singularities. In addition, this paper formulates the representation at a level of abstraction to show that it seamlessly applies to both rigid-body motion trajectories and interaction-force trajectories, thereby presenting it as a versatile tool for robotics, biomechanics, and related domains.

Keywords: Trajectory · Invariance · Screw Theory · Twist · Wrench.

1 Introduction

Analyzing the trajectories of rigid bodies and of interaction forces is essential for a wide range of tasks in robotics [23], biomechanics [1], and related fields. These tasks include trajectory segmentation [19], recognition [7, 9, 21], and prediction [13]. Typically, these tasks are performed using algorithms that operate

⋆ Corresponding author: arno.verduyn@kuleuven.be

on trajectory coordinates expressed in a specific coordinate system, such as a body-fixed coordinate system attached to a mobile robot, or a coordinate system attached to a sensor. A key challenge, however, is to achieve consistent results across different choices of coordinate system [22]. Different choices of coordinate system can arise, for example, from changes in robot configuration or from variations in motion-capture setups. A common way to avoid this challenge is to enforce consistent coordinate systems during data collection through coordinate system calibration procedures. However, these calibration procedures are often labour-intensive. Furthermore, an open problem remains: how to handle existing datasets that contain trajectories expressed in different coordinate systems and where the relationships between the different coordinate systems are unknown. A more rigorous way to address this challenge is to learn trajectory models that generalize across coordinate systems. In the literature, methods exist to learn such models from data [7,8,21,24]. A differentiator among these methods is the amount of data that is required. In this paper, we focus on methods that require only one demonstration. Specifically, we focus on methods based on linear algebra and differential geometry that transform the demonstrated trajectory into a representation that remains unchanged when the coordinate system changes, referred to as a *coordinate-invariant representation*.

Invariant representations have been introduced in the literature. For example, the work in [7] obtains an invariant representation of the trajectory by embedding it into a coordinate-invariant subspace computed via a singular value decomposition. However, this approach requires data of the complete trajectory to be available a priori. Other works [5, 9, 21, 23] aim to eliminate this requirement by proposing representations based on *local* trajectory features. Here, a local feature is a feature defined exclusively from trajectory data within a small neighborhood along the trajectory. For example, the works of [5, 9, 21] compute such local, differential-geometry-based trajectory features analytically in closed form. The approach in [23] proposes a more robust computation of local trajectory features using iterative, optimization-based methods.

However, obtaining both locality and coordinate invariance has its cost, since local invariant representations often show high sensitivity to measurement noise near singularities in the representation, where the representation is not uniquely defined. Furthermore, the iterative optimization-based method proposed in [23] requires relatively long computation times compared to analytical approaches, which limits its suitability for real-time use. These limitations motivate the research question addressed in this paper:

How can rigid-body trajectories and interaction-force trajectories be represented in a coordinate-invariant manner while ensuring (1) locality of the representation, (2) low computational complexity, and (3) low noise sensitivity?

To address this question, this paper provides a focused and formalized treatment of a novel invariant trajectory representation recently introduced in preprint form in [17]. For clarity and further reference, we introduce the representation here as the Dual-Upper-Triangular Invariant Representation (DUTIR).

This paper proves the coordinate invariance of DUTIR and introduces regularization to reduce the DUTIR's sensitivity to measurement noise near singularities. In addition, this paper formulates the representation at a level of abstraction to show that it seamlessly applies to both rigid-body motion trajectories and interaction-force trajectories, thereby presenting it as a versatile tool for robotics, biomechanics, and related domains.

2 Preliminaries

This section explains the concepts of *twist*, *wrench* and the general *screw*, which are mathematical entities that can be used to represent rigid-body motions, interaction forces, and their *trajectories*.

2.1 Rigid-Body Motion

A *body* can be interpreted as a collection of points. If the relative distances between these points remain constant over time, the body is considered non-deformable or *rigid*. To describe its motion, we commonly define a coordinate system and rigidly attach it to the body. The orientation of this coordinate system then represents the orientation of the body, while the position of its origin represents the position of a chosen point of the body, referred to as a *body reference point*. The motion of the rigid body can then be completely described by the kinematics of this coordinate system attached to the body. The first-order kinematics consist of the rotational velocity $\boldsymbol{\omega}$ of the coordinate system and the translational velocity $\boldsymbol{v}$ of its origin.

The two three-dimensional vectors, $\boldsymbol{\omega}$ and $\boldsymbol{v}$, are commonly combined into a single six-dimensional vector, $\boldsymbol{t} = (\boldsymbol{\omega}^\top, \boldsymbol{v}^\top)^\top$, referred to as the *twist*. Numerically, the twist can be computed from the position and orientation of the body-fixed coordinate system at two successive time instances using the logarithmic map [10]. In this paper, we only consider twists where the coordinates of both $\boldsymbol{\omega}$ and $\boldsymbol{v}$ are expressed in one and the same coordinate system, and $\boldsymbol{v}$ represents the velocity of the origin of that coordinate system. Such a twist is also referred to as a *screw* [11], as further explained in Section 2.3.

We indicate the choice of coordinate system for the twist with a subscript. For example, ${}_a\boldsymbol{t}$ is read as "the twist expressed in coordinate system $\{a\}$." Changing between coordinate systems, for example from $\{a\}$ to $\{b\}$ can be done using the 6×6 *transformation matrix* ${}^a_b\boldsymbol{S}$:

$$ {}_b\boldsymbol{t} = {}^a_b\boldsymbol{S}\ {}_a\boldsymbol{t} \quad \text{with} \quad {}^a_b\boldsymbol{S} = \begin{bmatrix} {}^a_b\boldsymbol{R} & \boldsymbol{0} \\ [{}^a_b\boldsymbol{p}]_\times\ {}^a_b\boldsymbol{R} & {}^a_b\boldsymbol{R} \end{bmatrix}, \tag{1} $$

and where ${}^a_b\boldsymbol{R}$ represents the relative orientation of $\{a\}$ with respect to $\{b\}$, encoded as a 3×3 orthogonal rotation matrix, and where ${}^a_b\boldsymbol{p}$ represents the position vector from the origin of $\{b\}$ to the origin of $\{a\}$, with its coordinates expressed in coordinate system $\{b\}$. The operator $[\,\cdot\,]_\times$ in (1) transforms the

three-dimensional vector ${}^a_b\boldsymbol{p}$ into a 3×3 skew-symmetric matrix, which is commonly used to represent cross-product operations in matrix form [11]. In the remainder of this paper, the sub- and superscripts of ${}^a_b\boldsymbol{S}$ are omitted unless their explicit inclusion is required for clarity.

2.2 Rigid-Body Wrench

A rigid body can be subjected to forces and torques. In that case, all the forces and torques acting on the body can be reduced to a single resultant force $\boldsymbol{f}$ and a single resultant torque $\boldsymbol{\tau}$ about a specified reference point without loss of generality. The two three-dimensional vectors $\boldsymbol{f}$ and $\boldsymbol{\tau}$ can be combined into a six-dimensional vector $\boldsymbol{w} = (\boldsymbol{f}^\top, \boldsymbol{\tau}^\top)^\top$, referred to as the *wrench*. Similarly to twists, we again only consider wrenches with their components, $\boldsymbol{f}$ and $\boldsymbol{\tau}$, corresponding to a single coordinate system. Changing between coordinate systems can be done similarly as for twists using the same 6×6 transformation matrix in (1), thanks to the chosen ordering of the components $\boldsymbol{f}$ and $\boldsymbol{\tau}$ in the wrench.

2.3 Abstraction to General Screws

Both the twist and the wrench belong to the category of *screws*. A screw is a six-dimensional vector that can be associated with a physically meaningful axis in space, referred to as the underlying *screw axis*. The interpretation of screw axes for twists and wrenches originates from classical mechanics, with Chasles [3] showing that a rigid-body motion can always be described as a rotation and translation along an axis in space, and with Poinsot [12] showing that force systems can always be reduced to a single force and torque along an axis.

Formally, a screw, denoted as $\boldsymbol{\xi}$, is a six-dimensional vector consisting of two vectors, $\boldsymbol{\alpha} \in \mathbb{R}^{3\times1}$ and $\boldsymbol{\beta} \in \mathbb{R}^{3\times1}$:

$$\boldsymbol{\xi} = \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix}. \tag{2}$$

For twists $\boldsymbol{\alpha} = \boldsymbol{\omega}$ and $\boldsymbol{\beta} = \boldsymbol{v}$, while for wrenches $\boldsymbol{\alpha} = \boldsymbol{f}$ and $\boldsymbol{\beta} = \boldsymbol{\tau}$. The direction of the screw axis of $\boldsymbol{\xi}$, denoted as $\boldsymbol{e}$, can be found by:

$$\boldsymbol{e} = \frac{\boldsymbol{\alpha}}{\|\boldsymbol{\alpha}\|}, \tag{3}$$

while the position of the point $\boldsymbol{p}_\perp$ on the screw axis closest to the origin of the coordinate system can be found by:

$$\boldsymbol{p}_\perp = \frac{\boldsymbol{\alpha} \times \boldsymbol{\beta}}{\|\boldsymbol{\alpha}\|^2}. \tag{4}$$

Note that all screws $\boldsymbol{\xi}$ transform between coordinate systems with the transformation matrix $\boldsymbol{S}$ as defined in (1). For this reason, the matrix $\boldsymbol{S}$ is also referred to as the *screw-transformation matrix*.

In the remainder of this paper, we reason at the level of general screws. As a result, the introduced concepts and methods apply seamlessly to representations of rigid-body motion and of interaction forces.

2.4 Screw Trajectories

Screws are often evaluated at consecutive time instances to analyse how they evolve during a certain time duration. In that case, time is used as the *progress variable*. Apart from time, other choices for the progress variable exist. For example, when time-invariant analysis of rigid-body motion trajectories is desired, a *geometric* progress variable is often used. Typical geometric progress variables include the angle traversed by the body during rotation [14], and the arc length traced by a reference point attached to the body during translation [25]. In the remainder of this paper, we abstract away from the specific choice of progress variable and reason at the level of a generic progress variable denoted by x.

We indicate the progress instance at which the screw $\boldsymbol{\xi}$ is evaluated explicitly by adding the argument $[x_i]$, yielding $\boldsymbol{\xi}[x_i]$. We refer to the ordered sequence of screws: $\boldsymbol{\xi}[x_i]$ for $i \in [1, M]$ as a *discrete screw trajectory*, where M represents the total number of trajectory samples.

The computation of the invariant trajectory representation proposed in this paper takes such discrete screw trajectories as input. Hence, it depends only on the resulting sequence $\boldsymbol{\xi}[x_i]$, not on the specific physical interpretation of the progress variable x. Hence, the proposed invariant trajectory representation applies seamlessly to any choice of progress variable.

3 Proposed Method

The coordinates of screw trajectories depend on the choice of coordinate system in which they are expressed. The aim of this section is to introduce a novel representation of screw trajectories that is invariant to the choice of coordinate system. The proposed representation is obtained by first encoding the local evolution of the screw trajectory (see Sec. 3.1) and then describing this local trajectory evolution in a coordinate-invariant manner (see Sec. 3.2).

3.1 Encoding of Local Trajectory Evolution Using Screw Triplets

The magnitude, direction, and local evolution of the screw $\boldsymbol{\xi}$ at progress instance x_i are encoded in the triplet of consecutive screws: $\boldsymbol{\xi}[x_{i-1}]$, $\boldsymbol{\xi}[x_i]$, and $\boldsymbol{\xi}[x_{i+1}]$. We denote this triplet of consecutive screws by the 6×3 matrix $\boldsymbol{\Xi}[x_i]$:

$$\boldsymbol{\Xi}[x_i] = \begin{bmatrix} \boldsymbol{\xi}[x_{i-1}] & \boldsymbol{\xi}[x_i] & \boldsymbol{\xi}[x_{i+1}] \end{bmatrix} = \begin{bmatrix} \boldsymbol{\alpha}[x_{i-1}] & \boldsymbol{\alpha}[x_i] & \boldsymbol{\alpha}[x_{i+1}] \\ \boldsymbol{\beta}[x_{i-1}] & \boldsymbol{\beta}[x_i] & \boldsymbol{\beta}[x_{i+1}] \end{bmatrix}. \tag{5}$$

Specifically, the local evolution of $\boldsymbol{\xi}[x_i]$ up to *second order* is encoded in $\boldsymbol{\Xi}[x_i]$. This property is illustrated by considering the standard central finite-difference approximation of the second-order derivative of the screw $\boldsymbol{\xi}[x_i]$:

$$\frac{d^2\boldsymbol{\xi}[x_i]}{dx_i^2} = \frac{\boldsymbol{\xi}[x_{i+1}] - 2\boldsymbol{\xi}[x_i] + \boldsymbol{\xi}[x_{i-1}]}{(\Delta x)^2} + \mathcal{O}(\Delta x), \tag{6}$$

which relies on the three twists in $\boldsymbol{\Xi}[x_i]$. Further on, we omit the argument $[x_i]$ when clear from the context.

3.2 Coordinate-Invariant Representation of $\boldsymbol{\Xi}$

We will show that the matrix $\boldsymbol{\Xi}$ can be mathematically decomposed into the product of a screw-transformation matrix $\boldsymbol{S}$ and a coordinate-invariant representation $\boldsymbol{U}$. We refer to this novel decomposition as the *SU-decomposition*[5] and to the corresponding representation as the *Dual-Upper-Triangular Invariant Representation* (DUTIR). Theorem 1 introduces these concepts, while Property 1 details the coordinate invariance of the corresponding representation.

Theorem 1 (Existence of an *SU*-Decomposition). *For any $\boldsymbol{\Xi}[x_i]$ in $\mathbb{R}^{6\times 3}$ that satisfies the two regularity conditions:*

$$\|\boldsymbol{\alpha}[x_{i-1}]\| \neq 0, \quad \textit{and} \quad \boldsymbol{\alpha}[x_{i-1}] \times \boldsymbol{\alpha}[x_i] \neq \boldsymbol{0}, \tag{7}$$

there exists a unique 6×6 screw-transformation matrix $\boldsymbol{S}$ and 6×3 twice upper triangular matrix $\boldsymbol{U}$, such that:

$$\boldsymbol{\Xi} = \boldsymbol{S}\,\boldsymbol{U}, \tag{8}$$

where $\boldsymbol{U}$ is obtained by stacking two 3×3 upper-triangular blocks $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$:

$$\boldsymbol{U} = \begin{bmatrix}\boldsymbol{U}_1\\ \boldsymbol{U}_2\end{bmatrix} = \begin{bmatrix} u_{11} & u_{12} & u_{13}\\ 0 & u_{22} & u_{23}\\ 0 & 0 & u_{33}\\ u_{41} & u_{42} & u_{43}\\ 0 & u_{52} & u_{53}\\ 0 & 0 & u_{63}\end{bmatrix}, \tag{9}$$

and where the convention $u_{11} > 0$ and $u_{22} > 0$ is imposed to ensure uniqueness.

Proof. Decomposition (8) can be expanded by writing $\boldsymbol{\Xi}$ as a vertical stack of two 3×3 matrices, $\boldsymbol{A}$ and $\boldsymbol{B}$, and by using the definition of $\boldsymbol{S}$ in (1):

$$\boldsymbol{\Xi} = \begin{bmatrix}\boldsymbol{A}\\ \boldsymbol{B}\end{bmatrix} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{0}\\ [\boldsymbol{p}]_\times \boldsymbol{R} & \boldsymbol{R}\end{bmatrix}\begin{bmatrix}\boldsymbol{U}_1\\ \boldsymbol{U}_2\end{bmatrix}. \tag{10}$$

The first block row of (10) can be expressed as follows:

$$\boldsymbol{A} = \boldsymbol{R}\,\boldsymbol{U}_1 = \boldsymbol{R}\begin{bmatrix} u_{11} & u_{12} & u_{13}\\ 0 & u_{22} & u_{23}\\ 0 & 0 & u_{33}\end{bmatrix}, \tag{11}$$

where the first two columns of $\boldsymbol{R}$ and $\boldsymbol{U}_1$ can be found using a standard QR-decomposition algorithm [6], which is well-defined if the first two diagonal elements of $\boldsymbol{U}_1$, u_{11} and u_{22}, both differ from zero. Afterwards, the sign convention, $u_{11} > 0$ and $u_{22} > 0$, can be imposed by flipping the sign of the corresponding row of $\boldsymbol{U}_1$ and of the corresponding column of $\boldsymbol{R}$ whenever necessary. The third column of $\boldsymbol{R}$ and the sign of the third diagonal element u_{33} is then determined

[5] The proposed decomposition is inspired by the QR-decomposition of a 3×3 matrix.

by imposing that $\boldsymbol{R}$ is a right-handed orthogonal matrix. This procedure results in a unique solution for the rotation matrix $\boldsymbol{R}$ and upper-triangular matrix $\boldsymbol{U}_1$.

The conditions for non-zero diagonal elements, $u_{11} \neq 0$ and $u_{22} \neq 0$, result in the regularity conditions shown in (7). This follows from the definition of the QR-decomposition. Specifically, the first two columns of $\boldsymbol{R}$, i.e. $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$, are related to $\boldsymbol{\alpha}[x_{i-1}]$ and $\boldsymbol{\alpha}[x_i]$ as follows:

$$\boldsymbol{r}_1 = \frac{\boldsymbol{\alpha}[x_{i-1}]}{\|\boldsymbol{\alpha}[x_{i-1}]\|}, \quad \text{and} \quad \boldsymbol{r}_2 = \frac{(\boldsymbol{\alpha}[x_{i-1}] \times \boldsymbol{\alpha}[x_i]) \times \boldsymbol{\alpha}[x_{i-1}]}{\|(\boldsymbol{\alpha}[x_{i-1}] \times \boldsymbol{\alpha}[x_i]) \times \boldsymbol{\alpha}[x_{i-1}]\|}. \tag{12}$$

Using (11), the scalars u_{11}, u_{12} and u_{22} can be written as:

$$u_{11} = \boldsymbol{r}_1 \cdot \boldsymbol{\alpha}[x_{i-1}] \quad , \quad u_{12} = \boldsymbol{r}_1 \cdot \boldsymbol{\alpha}[x_i] \quad , \quad \text{and} \quad u_{22} = \boldsymbol{r}_2 \cdot \boldsymbol{\alpha}[x_i]. \tag{13}$$

From (12)-(13), it follows that u_{11} is well-defined and non-zero if $\|\boldsymbol{\alpha}[x_{i-1}]\| \neq 0$, while u_{22} is well-defined and non-zero if additionally $\boldsymbol{\alpha}[x_{i-1}] \times \boldsymbol{\alpha}[x_i] \neq \boldsymbol{0}$.

Thus far, matrices $\boldsymbol{R}$ and $\boldsymbol{U}_1$ are completely and uniquely determined. The position vector $\boldsymbol{p}$ in (10) can then be determined as follows. After pre-multiplication with $\boldsymbol{R}^\top$, the lower block row of (10) can be written as:

$$\boldsymbol{R}^\top \boldsymbol{B} = \left(\boldsymbol{R}^\top [\boldsymbol{p}]_\times \boldsymbol{R}\right) \boldsymbol{U}_1 + \boldsymbol{U}_2. \tag{14}$$

Equation (14) can be rewritten using the orthogonal basis transformation property of a skew-symmetric matrix [11]: $\boldsymbol{R}^\top [\boldsymbol{p}]_\times \boldsymbol{R} = [\boldsymbol{R}^\top \boldsymbol{p}]_\times$. This results in:

$$\boldsymbol{R}^\top \boldsymbol{B} = \left([\boldsymbol{R}^\top \boldsymbol{p}]_\times\right) \boldsymbol{U}_1 + \boldsymbol{U}_2. \tag{15}$$

Introducing $\boldsymbol{p}^* \coloneqq \boldsymbol{R}^\top \boldsymbol{p}$ results in:

$$\boldsymbol{R}^\top \boldsymbol{B} = [\boldsymbol{p}^*]_\times \boldsymbol{U}_1 + \boldsymbol{U}_2. \tag{16}$$

From the lower-diagonal entries in (16), three scalar equations are obtained that do not rely on the unknown upper-triangular elements in $\boldsymbol{U}_2$.

$$(\boldsymbol{R}^\top \boldsymbol{B})_{21} = u_{11} p_z^*, \tag{17}$$

$$(\boldsymbol{R}^\top \boldsymbol{B})_{31} = -u_{11} p_y^*, \tag{18}$$

$$(\boldsymbol{R}^\top \boldsymbol{B})_{32} = u_{22} p_x^* - u_{12} p_y^*, \tag{19}$$

where the notation $(\boldsymbol{R}^\top \boldsymbol{B})_{ij}$ in the left-hand side of equations (17)-(19) denotes the element in the i^{th} row and j^{th} column of $\boldsymbol{R}^\top \boldsymbol{B}$.

The coordinates p_x^*, p_y^*, p_z^* of $\boldsymbol{p}^*$ can be uniquely determined from (17)-(19), again under the same conditions: $u_{11} \neq 0$ and $u_{22} \neq 0$. Now that $\boldsymbol{R}$, $\boldsymbol{U}_1$, and $\boldsymbol{p}^*$ are completely determined, $\boldsymbol{p}$ can be found as $\boldsymbol{p} = \boldsymbol{R}\, \boldsymbol{p}^*$, while $\boldsymbol{U}_2$ can be solved from (16):

$$\boldsymbol{U}_2 = \boldsymbol{R}^\top \boldsymbol{B} - [\boldsymbol{p}^*]_\times \boldsymbol{U}_1. \tag{20}$$

The upper-triangular matrices $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$, together with the orientation $\boldsymbol{R}$ and position $\boldsymbol{p}$ are now completely and uniquely determined. The derivation of this SU-decomposition algorithm proves that a unique SU-decomposition exists under the derived regularity conditions. ■

Property 1. The matrix $\boldsymbol{U}$ is invariant to changes in coordinate system.

Proof. Consider $\boldsymbol{\Xi}$ expressed in coordinate system $\{1\}$, denoted as ${}_1\boldsymbol{\Xi}$. The coordinate system of ${}_1\boldsymbol{\Xi}$ can be changed to a different coordinate system $\{2\}$ by left multiplication with the screw-transformation matrix ${}^1_2\boldsymbol{S}$, such that:

$$ {}_2\boldsymbol{\Xi} = {}^1_2\boldsymbol{S}\ {}_1\boldsymbol{\Xi}. \tag{21} $$

Substituting the SU-decomposition (8) of ${}_1\boldsymbol{\Xi}$ in (21) results in:

$$ {}_2\boldsymbol{\Xi} = {}^1_2\boldsymbol{S}\ \left({}_1\boldsymbol{S}\ \boldsymbol{U}\right). \tag{22} $$

Since the matrix multiplication of the two screw-transformation matrices, ${}^1_2\boldsymbol{S}$ and ${}_1\boldsymbol{S}$, yields another screw-transformation matrix ${}_2\boldsymbol{S}$, this results in:

$$ {}_2\boldsymbol{\Xi} = {}_2\boldsymbol{S}\ \boldsymbol{U}. \tag{23} $$

This proves that ${}_2\boldsymbol{S}\ \boldsymbol{U}$ is a valid SU-decomposition of ${}_2\boldsymbol{\Xi}$. Moreover, since the SU-decomposition is unique, the decomposition ${}_2\boldsymbol{S}\ \boldsymbol{U}$ is the only possible SU-decomposition of ${}_2\boldsymbol{\Xi}$. Hence, the SU-decomposition of ${}_2\boldsymbol{\Xi}$ necessarily results in the same $\boldsymbol{U}$ matrix as the one for ${}_1\boldsymbol{\Xi}$. This proves that $\boldsymbol{U}$ is invariant to changes in the coordinate system of $\boldsymbol{\Xi}$. ■

Remark 1. The matrix $\boldsymbol{\Xi}$ is a 6×3 matrix and hence contains 18 degrees of freedom (DoFs). In the SU-decomposition (8), these 18 DoFs are redistributed between $\boldsymbol{S}$ and $\boldsymbol{U}$. Matrix $\boldsymbol{S}$ contains 6 DoFs: three arising from the position vector $\boldsymbol{p}$ and three arising from the rotation matrix $\boldsymbol{R}$. The remaining 12 DoFs are contained in $\boldsymbol{U}$. The variability of the entries of $\boldsymbol{\Xi}$ can hence be separated into variability due to (1) a change of coordinate system, captured entirely by $\boldsymbol{S}$, and (2) a change in the intrinsic evolution of the screw $\boldsymbol{\xi}[x_i]$, captured in an invariant manner up to second order by $\boldsymbol{U}$. Furthermore, since both $\boldsymbol{S}$ and $\boldsymbol{U}$ are unique in the regular case, the SU-decomposition is a *bijective map* between $\boldsymbol{\Xi}$ and the pair of matrices $(\boldsymbol{S}, \boldsymbol{U})$.

3.3 Geometric Interpretation of the SU-decomposition

The matrix $\boldsymbol{R}$ and vector $\boldsymbol{p}$ can be interpreted as the orientation and position coordinates of an orthogonal coordinate system, functionally defined by the screw axis and its motion (see Fig. 1). The first column $\boldsymbol{r}_1$ of $\boldsymbol{R}$ defines the x-axis of this functional coordinate system and is aligned with the screw axis of $\boldsymbol{\xi}[x_{i-1}]$. The third column $\boldsymbol{r}_3$ of $\boldsymbol{R}$ defines the z-axis of this functional coordinate system and is aligned with the common normal of the screw axes of $\boldsymbol{\xi}[x_{i-1}]$ and $\boldsymbol{\xi}[x_i]$. The y-axis follows from right-handed orthogonality. The origin $\boldsymbol{p}$ coincides with the intersection point of the screw axis of $\boldsymbol{\xi}[x_{i-1}]$ and the common normal. Although this geometric interpretation can be intuitively inferred from the structure of $\boldsymbol{U}$, particularly its zero entries, proofs of this interpretation are provided in Appendix A for completeness.

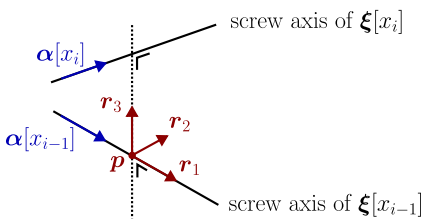

Fig. 1. Geometric interpretation of $\boldsymbol{p}$ and $\boldsymbol{R} = [\boldsymbol{r}_1\ \boldsymbol{r}_2\ \boldsymbol{r}_3]$, which are the components of the screw-transformation matrix $\boldsymbol{S}$ in the SU-decomposition of $\boldsymbol{\Xi} = [\boldsymbol{\xi}[x_{i-1}]\ \boldsymbol{\xi}[x_i]\ \boldsymbol{\xi}[x_{i+1}]]$. The screw axes of $\boldsymbol{\xi}[x_{i-1}]$ and $\boldsymbol{\xi}[x_i]$ are shown as solid black lines. The common normal of the two screw axes is shown as a dotted black line.

Remark 2. This geometric interpretation highlights that the SU-decomposition not only allows to compute the DUTIR of $\boldsymbol{\Xi}$, but it also enables an efficient, closed-form computation of key geometric entities. In particular, by computing $\boldsymbol{p}$ and $\boldsymbol{R}$, we compute the position and orientation of the screw axis of $\boldsymbol{\xi}[x_{i-1}]$ as well as the common normal of the screw axes of $\boldsymbol{\xi}[x_{i-1}]$ and $\boldsymbol{\xi}[x_i]$.

Remark 3. The position $\boldsymbol{p}$ and matrix $\boldsymbol{R}$ are geometric entities anchored to the screw trajectory. As a result, they transform consistently under changes of coordinate system. Consequently, the computation of $\boldsymbol{R}$ and $\boldsymbol{p}$ from $\boldsymbol{\Xi}$ constitutes an *equivariant map* [4] with respect to coordinate system transformations.

3.4 Regularization of the SU-decomposition

When the regularity conditions (7) are not met, the QR-decomposition of $\boldsymbol{A}$ is ill-defined and the system of equations (17)-(19) becomes underdetermined. Consequently, an infinite number of possible solutions for $\boldsymbol{R}$, $\boldsymbol{p}^*$, and $\boldsymbol{p}$ exist. An intuitive example is the case of a pure translational motion (where $\boldsymbol{\alpha} = \boldsymbol{\omega} = \boldsymbol{0}$ and $\boldsymbol{\beta} = \boldsymbol{v} \neq \boldsymbol{0}$). In this case, the screw axes of $\boldsymbol{\xi}[x_{i-1}]$ and $\boldsymbol{\xi}[x_i]$ are ill-defined, since a pure translation can be represented by (1) a screw axis at a finite distance from the moving body and parallel to the translational velocity $\boldsymbol{v}$, or by (2) a screw axis at an infinite distance from the body and perpendicular to the translational velocity $\boldsymbol{v}$ [16] . In practice, the second solution is rarely meaningful. Inspired by the first, more intuitive case, we introduce a regularization procedure for the SU-decomposition in this subsection.

1) Regularization of $\boldsymbol{p}$: To regularize $\boldsymbol{p}$, we first predefine a maximum allowed distance between $\boldsymbol{p}$ and the origin of the coordinate system and denote it as L. This value for L can be interpreted as a *geometric scale*, defining a spatial region in which the computed value of $\boldsymbol{p}$ is considered relevant. Then, after solving (17) - (19) to obtain $\boldsymbol{p}^*$, we evaluate whether regularization is required, which is the case when $\|\boldsymbol{p}^*\| > L$. In that case, we search for a regularized version of $\boldsymbol{p}^*$, denoted as $\hat{\boldsymbol{p}}^*$, that satisfies $\|\hat{\boldsymbol{p}}^*\| = L$. After performing the regularization action (which is detailed later in this subsection), the regularized matrix $\hat{\boldsymbol{U}}_2$ can be found by solving (20) using $\hat{\boldsymbol{p}}^*$. An important consequence of

this regularization is that, when active, the regularized matrix $\hat{\boldsymbol{U}}_2$ is no longer guaranteed to be upper-triangular. Instead, the lower-diagonal elements of $\hat{\boldsymbol{U}}_2$ can be non-zero, with their values depending on the location of the origin of the coordinate system. Ideally, these lower-diagonal elements remain small such that a high resemblance to the original $\boldsymbol{U}_2$ is obtained. To this end, we design the regularization such that the sum of squares of the lower-diagonal terms in the first column of $\hat{\boldsymbol{U}}_2$ are minimized. Expressions for these terms, denoted as ε_{51} and ε_{61}, follow from (17) and (18):

$$\varepsilon_{51} = (\boldsymbol{R}^\top \boldsymbol{B})_{21} - u_{11}\hat{p}^*_z \quad \text{and} \quad \varepsilon_{61} = (\boldsymbol{R}^\top \boldsymbol{B})_{31} + u_{11}\hat{p}^*_y, \tag{24}$$

where ε_{ij} denotes the lower-diagonal entry at the i-th row and j-th column of $\boldsymbol{U}$. Minimizing the sum of squares of ε_{51} and ε_{61} subject to the equality constraint $\|\hat{\boldsymbol{p}}^*\| = L$ can be solved analytically using the method of Lagrange multipliers [2]. Specifically, we search for the stationary points of the Lagrangian $\mathscr{L}(\hat{\boldsymbol{p}}^*, \lambda)$ defined in terms of the variable $\hat{\boldsymbol{p}}^*$ and the Lagrange multiplier λ:

$$\mathscr{L}(\hat{\boldsymbol{p}}^*, \lambda) = \varepsilon_{51}^2 + \varepsilon_{61}^2 + \lambda\left(L^2 - \|\hat{\boldsymbol{p}}^*\|^2\right). \tag{25}$$

The following two solutions for $\hat{\boldsymbol{p}}^*$ exist:

$$\hat{\boldsymbol{p}}^* = \begin{pmatrix} \text{sign}(p^*_x)\sqrt{L^2 - (p^*_y)^2 - (p^*_z)^2} \\ p^*_y \\ p^*_z \end{pmatrix} \quad \text{or} \quad \hat{\boldsymbol{p}}^* = \begin{pmatrix} 0 \\ L\frac{p^*_y}{\sqrt{(p^*_y)^2+(p^*_z)^2}} \\ L\frac{p^*_z}{\sqrt{(p^*_y)^2+(p^*_z)^2}} \end{pmatrix}. \tag{26}$$

The first solution in (26) consists of retaining p^*_y and p^*_z, while projecting p^*_x along the screw axis of $\boldsymbol{\xi}[x_{i-1}]$ onto the spherical manifold defined by L. This solution is a valid solution when $(p^*_y)^2 + (p^*_z)^2 < L^2$. The second solution in (26) consists of setting $\hat{p}^*_x = 0$, while isotropically projecting p^*_y and p^*_z onto the spherical manifold defined by L. The two solutions for $\hat{\boldsymbol{p}}^*$ are visualized in Figure 2.

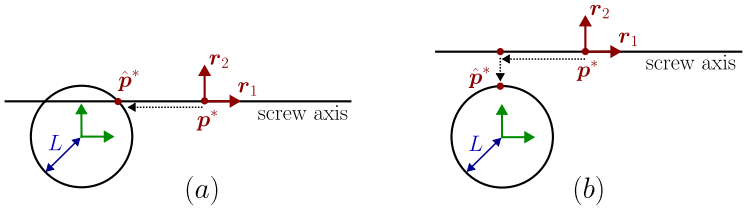

Fig. 2. Two-dimensional illustration of the regularization of $\boldsymbol{p}^*$, showing the coordinate system of $\boldsymbol{\Xi}$ (green), the spherical manifold defined by L (black circle), the screw axis of $\boldsymbol{\xi}[x_{i-1}]$ (black line), and the functional coordinate system defined by $\boldsymbol{p}^*$, $\boldsymbol{r}_1$, and $\boldsymbol{r}_2$ (red). The solution for $\hat{\boldsymbol{p}}^*$ is illustrated for (a) the case where the screw axis intersects the spherical manifold, i.e., when $(p^*_y)^2 + (p^*_z)^2 < L^2$, and (b) the case where the screw axis does not intersect the spherical manifold, i.e., when $(p^*_y)^2 + (p^*_z)^2 > L^2$.

Based on these two solutions, the regularization of $\boldsymbol{p}$ is implemented as follows. We first compute $\boldsymbol{p}^*$ and test whether regularization is required, which is the

case when $\|\boldsymbol{p}^*\| > L$. If so, we test whether $(p_y^*)^2 + (p_z^*)^2 < L^2$. If this condition holds, it means that the screw axis intersects the spherical manifold (see Fig. 2a) and the first solution for $\hat{\boldsymbol{p}}^*$ should be selected. If this condition does not hold, we use the second solution. Finally, $\hat{\boldsymbol{p}}$ is found as $\hat{\boldsymbol{p}} = \boldsymbol{R}\,\hat{\boldsymbol{p}}^*$, and the matrix $\hat{\boldsymbol{U}}_2$ can be found by solving (20) using $\hat{\boldsymbol{p}}^*$.

2) Regularization of $\boldsymbol{R}$: We propose to additionally regularize $\boldsymbol{R}$, since the regularization of $\boldsymbol{p}$ breaks the exact upper-triangular structure of $\boldsymbol{U}_2$, and this structure can be approximately restored by regularizing $\boldsymbol{R}$.

Conceptually, we search for the regularized matrix $\hat{\boldsymbol{R}}$ that yields matrices $\hat{\boldsymbol{U}}_1$ and $\hat{\boldsymbol{U}}_2$ that are both as close as possible to upper-triangular matrices. Importantly, exact upper-triangular structure of both $\hat{\boldsymbol{U}}_1$ and $\hat{\boldsymbol{U}}_2$ cannot be achieved. Such exact structure corresponds to the unique solution of the unregularized SU-decomposition, from which we deviate due to the regularization of $\boldsymbol{p}$.

Specifically, we propose to regularize $\boldsymbol{R}$ as $\hat{\boldsymbol{R}} = \boldsymbol{R}\boldsymbol{R}_c^\top$, where $\boldsymbol{R}_c$ is a corrective rotation matrix determined via solving an orthogonal Procrustes problem:

$$\underset{\boldsymbol{R}_c}{\text{minimize}} \quad w^2 \left\| \boldsymbol{U}_1 - \boldsymbol{R}_c \boldsymbol{U}_1 \right\|_F^2 + \left\| \hat{\boldsymbol{U}}_{2,\triangle} - \boldsymbol{R}_c \hat{\boldsymbol{U}}_2 \right\|_F^2, \tag{27}$$

$$\text{subject to} \quad \boldsymbol{R}_c^\top \boldsymbol{R}_c = \boldsymbol{I}_{3\times 3} \quad \text{and} \quad \det(\boldsymbol{R}_c) = 1, \tag{28}$$

which admits a closed-form solution [15]. In (27), $\|\cdot\|_F$ denotes the Frobenius norm, w is a weighting parameter expressed in meters, and $\hat{\boldsymbol{U}}_{2,\triangle}$ is the exact upper-triangular version of $\hat{\boldsymbol{U}}_2$. The matrix $\hat{\boldsymbol{U}}_{2,\triangle}$ is obtained from $\hat{\boldsymbol{U}}_2$ using the same procedure to transform $\boldsymbol{A}$ into the upper-triangular matrix $\boldsymbol{U}_1$ (11).

Introducing the weight w is necessary because the entries of $\boldsymbol{U}_1$ and $\hat{\boldsymbol{U}}_2$ have different physical units: for twists, $\boldsymbol{U}_1$ contains angular components in *rad/progress*, while $\hat{\boldsymbol{U}}_2$ contains translational components in *m/progress*. For wrenches, the corresponding units are *N* and *Nm*, respectively.

Given the computed matrix $\boldsymbol{R}_c$, the regularized matrix $\hat{\boldsymbol{U}}_1$ is obtained as $\hat{\boldsymbol{U}}_1 = \boldsymbol{R}_c \boldsymbol{U}_1$, while $\hat{\boldsymbol{U}}_2$ is updated as $\boldsymbol{R}_c \hat{\boldsymbol{U}}_2$.

Remark 4. The proposed regularization introduces two distinct solution cases for the regularized SU-decomposition. When the regularization is inactive, the solutions for $\hat{\boldsymbol{p}}$ and $\hat{\boldsymbol{R}}$ remain *purely coordinate-invariant*. When the regularization is active, the solutions for $\hat{\boldsymbol{p}}$ and $\hat{\boldsymbol{R}}$ become sensitive to the choice of the origin of the coordinate system.

Remark 5. We propose to set $w = L$ such that the regularization is completely determined by a single parameter. Choosing $w = L$ is allowed since w and L share the same physical unit (meters). Furthermore, this choice is conceptually well motivated: both L and w govern the extent to which the regularized SU-decomposition depends on the origin of the coordinate system. Specifically, decreasing either L or w increases the sensitivity to the location of this origin. In practice, we propose to treat L as a hyperparameter. For instance, in recognition tasks, a value for L can be selected via hyperparameter validation procedures, such as maximizing recognition performance on a held-out validation set.

3.5 Numerical Example

In this subsection, we present a numerical example demonstrating the application of the SU-decomposition to rigid-body motion trajectory data. We use real trajectory data from [19], corresponding to a human-demonstrated pouring task performed with a kettle. Note that working with real recordings of human demonstrations is particularly challenging due to both the measurement noise during motion capture and the substantially lower precision of human motion compared to robotic execution. The trajectory is parametrized with respect to time t, and consists of three distinct motion segments: sliding the kettle along the table ($t \in [0, 0.7]$), lifting the kettle ($t \in [0.7, 1.6]$), and pouring ($t \in [1.6, 2.8]$).

To highlight coordinate invariance, the SU-decomposition is applied twice using different choices of the coordinate system attached to the body (see Fig. 3a). From the successive positions and orientations of the coordinate system, twist coordinates are computed via numerical differentiation using the logarithmic map [10] (see Fig. 3b). As expected, the resulting twist coordinates differ under changes in coordinate system. Subsequently, $\boldsymbol{U}$ is evaluated along the trajectory at successive time instances. To assess the effect of regularization, we computed the SU-decomposition both without regularization (see Fig. 3c) and with regularization when choosing $L = 0.3$m (see Fig. 3d).

In the unregularized case (see Fig. 3c), the computed $\boldsymbol{U}$ is *strictly* twice upper-triangular. Also, the results in the two coordinate systems are *exactly* the same, confirming the coordinate invariance of $\boldsymbol{U}$. However, u_{53} and u_{63} exhibit a high noise sensitivity during the sliding segment ($t \in [0, 0.7]$). This high noise sensitivity arises since the sliding motion corresponds to a near-pure translation, and pure translation corresponds to a singularity in the SU-decomposition.

In the regularized case (see Fig. 3d), the computed $\boldsymbol{U}$ is *approximately* twice upper-triangular. Also, the results in the two coordinate systems are *approximately* the same. This hence demonstrates high robustness to coordinate-system changes, albeit without exact invariance. Figure 3d also shows that the noise sensitivity of u_{53} and u_{63} is substantially reduced, and that more pronounced signals appear in the profiles of u_{41}, u_{42}, and u_{43} during the sliding segment. This is a desirable effect since the magnitude of these signals corresponds to the magnitude of the translational velocity during sliding. These results hence showcase that the regularized approach yields a representation of near-pure translational motion that is more interpretable and less sensitive to noise.

Additional examples and the implementation of the SU-decomposition are provided at https://github.com/arnoverduyn/SU_decomposition.

4 Discussion and Conclusion

The goal of this paper was to present a coordinate-invariant representation of screw trajectories that ensures (1) locality of the representation, (2) low computational complexity, and (3) low noise sensitivity. This goal is achieved through the contributions presented in this work. Specifically, we presented the SU-decomposition and the corresponding DUTIR of $\boldsymbol{\Xi}$. The matrix $\boldsymbol{\Xi}$ captures

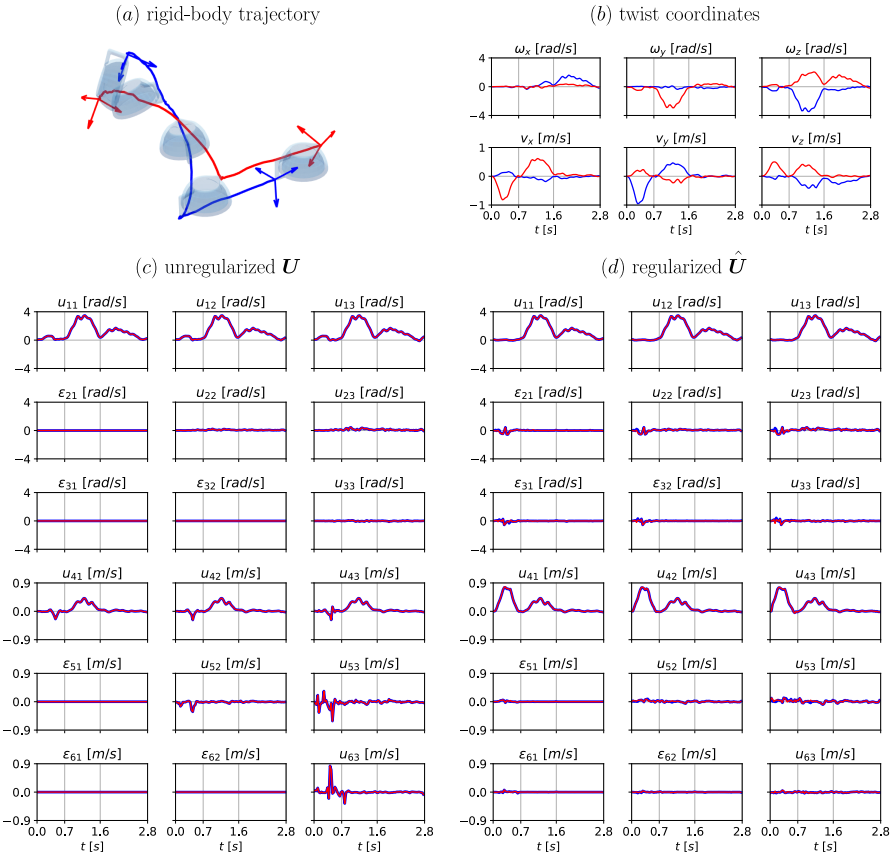

Fig. 3. Numerical example of the SU-decomposition applied to rigid-body trajectory data. (a) Visualization of the trajectory for two different choices (red and blue) of the coordinate system attached to the body; (b) Twist coordinates extracted from the trajectory via numerical differentiation using the logarithmic map [10]; (c) Components of the invariant representation $\boldsymbol{U}$ obtained without regularization. In this case, the lower-diagonal entries ε_{ij} of $\boldsymbol{U}$ are exactly zero; (d) Components of $\hat{\boldsymbol{U}}$ obtained with regularization. In this case, the lower-diagonal entries ε_{ij} of $\boldsymbol{U}$ are approximately zero.

the local evolution of the screw trajectory up to second order, while the DU-TIR, $\boldsymbol{U}$, provides a coordinate-invariant representation of this evolution. The proposed SU-decomposition is computationally efficient, since it involves the computation of analytical, closed-form solutions without iterative procedures. Furthermore, we presented regularization to reduce the sensitivity to noise of the SU-decomposition near irregularity.

The SU-decomposition and its regularization have been experimentally validated in prior work. Specifically, the work in [18] builds further on the SU-decomposition, derives a coordinate-invariant measure of similarity between screw trajectories, and validates this similarity measure for rigid-body trajectory recog-

nition. Across multiple rigid-body motion datasets and across different choices for the progress variable, this method has demonstrated high robustness to coordinate-system variations, outperforming similarity measures based on existing invariant trajectory representations [5, 7, 9, 21, 23, 25]. These results also showed that the proposed regularization in the SU-decomposition significantly reduced the sensitivity to noise when dealing with singular motions, such as pure translations. Although the proposed regularization reintroduces a dependency on the choice of the origin of the coordinate system, the sensitivity to this choice has shown to remain minor. This is because, by design, the regularization is only activated near singular cases and furthermore, the sensitivity of the invariant representation to variations in coordinate-system-origin is limited by minimizing the lower-diagonal terms in the first column of $\hat{\boldsymbol{U}}_2$.

In other prior work [20], a real-time recognition system was developed inspired by the proposed SU-decomposition. This system robustly recognized gestures (i.e. motion signatures traced by the palm of the human hand) across varying coordinate systems in real time, achieving a solid F_1-score of 92.3%.

The discussed prior works focused on the motion trajectories of rigid bodies. Importantly, the SU-decomposition and DUTIR are also directly applicable to the trajectories of interaction forces (i.e. through the wrench). The application to wrenches is for example relevant for recognizing robot actions in contact with the environment. Incorporating wrench information is then beneficial if the aim is to discriminate robot actions that exhibit similar motion profiles but differ significantly in their contact wrench profiles. Experimental validation of the proposed representation for this type of task is part of future work.

However, invariant representations have their limitations and hence should not be employed in every practical scenario. For example, in scenarios where coordinate system calibration is straightforward, reliable, and the trajectories are consistently performed relative to the calibrated coordinate system (e.g. performed at the same location, in the same direction, etc.), then traditional coordinate-dependent methods may outperform invariant methods. This is because invariant representations, by design, discard information tied to the coordinate system. Nevertheless, invariant representations are particularly valuable for analysing trajectories when coordinate systems are misaligned with their exact relationships unknown, when the relation between motion direction and coordinate axes is non-deterministic, or when aiming to eliminate the need for coordinate system calibration when such calibration is labour-intensive.

In conclusion, the proposed DUTIR and SU-decomposition offer a promising way for future algorithms in robotics that need to compare trajectories in an invariant manner without compromising robustness or computational efficiency.

Acknowledgments. This work was supported by a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 Research and Innovation Programme (Grant agreement No. 788298).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

A Interpretation of the *SU*-decomposition: Proofs

In Section 3.3, it is stated that:

1. The first column $\boldsymbol{r}_1$ of $\boldsymbol{R}$ is parallel to the screw axis of $\boldsymbol{\xi}[x_{i-1}]$;
2. The third column $\boldsymbol{r}_3$ of $\boldsymbol{R}$ is parallel to the common normal of the screw axes of $\boldsymbol{\xi}[x_{i-1}]$ and $\boldsymbol{\xi}[x_i]$;
3. The vector $\boldsymbol{p}$ represents a point on the screw axis of $\boldsymbol{\xi}[x_{i-1}]$;
4. The vector $\boldsymbol{p}$ represents the intersection point between the screw axis of $\boldsymbol{\xi}[x_{i-1}]$ and the common normal of the screw axes of $\boldsymbol{\xi}[x_{i-1}]$ and $\boldsymbol{\xi}[x_i]$.

In this appendix, we present proofs of these statements. Before presenting the proofs, we first summarize a set of relations involving $\boldsymbol{\alpha}[x_{i-1}]$, $\boldsymbol{\alpha}[x_i]$, $\boldsymbol{r}_1$, $\boldsymbol{r}_2$, u_{11}, u_{12}, and u_{22}. The vectors $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are obtained by orthonormalizing $\boldsymbol{\alpha}[x_{i-1}]$ and $\boldsymbol{\alpha}[x_i]$. The scalars u_{11}, u_{12}, and u_{22} represent the components of $\boldsymbol{\alpha}[x_{i-1}]$ and $\boldsymbol{\alpha}[x_i]$ along $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ (see Fig. 4). By applying trigonometric identities, the

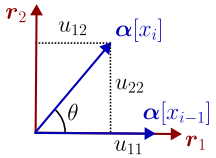

Fig. 4. Geometric relations between $\boldsymbol{\alpha}[x_{i-1}]$, $\boldsymbol{\alpha}[x_i]$, u_{11}, u_{12}, and u_{22}.

following relations can be derived:

$$\|\boldsymbol{\alpha}[x_{i-1}]\| = u_{11}, \tag{29}$$

$$\boldsymbol{\alpha}[x_{i-1}] \cdot \boldsymbol{\alpha}[x_i] = \|\boldsymbol{\alpha}[x_{i-1}]\|\|\boldsymbol{\alpha}[x_i]\| \cos\theta = u_{11}u_{12}, \tag{30}$$

$$\boldsymbol{\alpha}[x_{i-1}] \times \boldsymbol{\alpha}[x_i] = \|\boldsymbol{\alpha}[x_{i-1}]\|\|\boldsymbol{\alpha}[x_i]\| \sin\theta\ \boldsymbol{r}_3 = u_{11}u_{22}\boldsymbol{r}_3, \tag{31}$$

and:

$$\|\boldsymbol{\alpha}[x_{i-1}]\|^2\|\boldsymbol{\alpha}[x_i]\|^2 - (\boldsymbol{\alpha}[x_{i-1}] \cdot \boldsymbol{\alpha}[x_i])^2 = u_{11}^2\|\boldsymbol{\alpha}[x_i]\|^2\left(1 - \cos^2\theta\right) = u_{11}^2u_{22}^2. \tag{32}$$

Next, we present the proofs of the four enumerated statements.

Proof of 1: The proof follows from (3) and (12). Equation (3) shows that the direction of the screw axis of $\boldsymbol{\xi}[x_{i-1}]$ is obtained by normalizing the vector $\boldsymbol{\alpha}[x_{i-1}]$, while (12) confirms that $\boldsymbol{r}_1$ represents this normalized vector. ■

Proof of 2: The proof follows from (31), which shows that $\boldsymbol{r}_3$ lies parallel to the vector $\boldsymbol{\alpha}[x_{i-1}] \times \boldsymbol{\alpha}[x_i]$. This vector is orthogonal to the screw axes of $\boldsymbol{\xi}[x_{i-1}]$ and $\boldsymbol{\xi}[x_i]$, and thus defines the direction of their common normal. ■

Proof of 3: The point $\boldsymbol{p}$ lies on the screw axis of $\boldsymbol{\xi}[x_{i-1}]$ if and only if the following parametric line equation is satisfied:

$$\boldsymbol{p} = \boldsymbol{p}_\perp + p_\parallel \boldsymbol{e}. \tag{33}$$

The first term in (33) represents the point on the screw axis of $\boldsymbol{\xi}[x_{i-1}]$ closest to the origin of the coordinate system (see (4)). The second term in (33) represents a displacement along the screw axis of $\boldsymbol{\xi}[x_{i-1}]$, with $p_{\parallel} \in \mathbb{R}$ a free parameter.

By substituting (3) and (4) in (33), then left multiplying the result with $\boldsymbol{R}^{\top}$, and finally using $\boldsymbol{p}^* = \boldsymbol{R}^{\top}\boldsymbol{p}$ and (29), it can be shown that (33) reduces to:

$$\boldsymbol{p}^* = \frac{1}{u_{11}} \begin{pmatrix} 0 \\ -(\boldsymbol{R}^{\top}\boldsymbol{B})_{31} \\ (\boldsymbol{R}^{\top}\boldsymbol{B})_{21} \end{pmatrix} + p_{\parallel} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \tag{34}$$

which leads to the following three conditions:

$$p_x^* = p_{\parallel}, \quad u_{11}p_y^* = -(\boldsymbol{R}^{\top}\boldsymbol{B})_{31} \quad \text{and} \quad u_{11}p_z^* = (\boldsymbol{R}^{\top}\boldsymbol{B})_{21}. \tag{35}$$

The first condition is satisfied, since $p_{\parallel}$ is a free parameter. The other two conditions are also satisfied, since they correspond to the expressions for computing p_z^* and p_y^* shown in (17) and (18), respectively. This proves that the line equation in (33) is satisfied, and hence that $\boldsymbol{p}$ lies on the screw axis of $\boldsymbol{\xi}[x_{i-1}]$. ■

Proof of 4: The common normal of two axes is the line that perpendicularly intersects both axes. The distance between the two intersection points is the shortest distance between the two axes. These two intersection points can hence be found by identifying the points on each axis with minimum mutual distance.

Similarly to (33), we can define a line equation for the screw axis of the second screw $\boldsymbol{\xi}[x_i]$. We denote the entities corresponding to the first and second screw, $\boldsymbol{\xi}[x_{i-1}]$ and $\boldsymbol{\xi}[x_i]$, with the subscripts 1 and 2, respectively, resulting in:

$$\boldsymbol{p}_1 = \boldsymbol{p}_{\perp 1} + p_{\parallel 1}\, \boldsymbol{e}_1 \quad \text{and} \quad \boldsymbol{p}_2 = \boldsymbol{p}_{\perp 2} + p_{\parallel 2}\, \boldsymbol{e}_2. \tag{36}$$

The points on the two screw axes with minimal mutual distance can be found by minimizing the squared distance, $d^2 = \|\boldsymbol{p}_1 - \boldsymbol{p}_2\|^2$. The minimum can be found by searching for the points where the gradient, $\boldsymbol{\nabla} d^2 = (\frac{\partial d^2}{\partial p_{\parallel 1}}, \frac{\partial d^2}{\partial p_{\parallel 2}})$, is equal to the zero vector. This results in a system of two equations in the variables $p_{\parallel 1}$ and $p_{\parallel 2}$. Following this approach, the following expression for $p_{\parallel 1}$ can be found:

$$p_{\parallel 1} = \frac{\|\boldsymbol{\alpha}[x_{i-1}]\| \left[\boldsymbol{\alpha}[x_{i-1}] \cdot (\boldsymbol{\alpha}[x_i] \times \boldsymbol{\beta}[x_i])\right] + \frac{\boldsymbol{\alpha}[x_{i-1}] \cdot \boldsymbol{\alpha}[x_i]}{\|\boldsymbol{\alpha}[x_{i-1}]\|} \left[\boldsymbol{\alpha}[x_i] \cdot (\boldsymbol{\alpha}[x_{i-1}] \times \boldsymbol{\beta}[x_{i-1}])\right]}{\|\boldsymbol{\alpha}[x_{i-1}]\|^2 \|\boldsymbol{\alpha}[x_i]\|^2 - (\boldsymbol{\alpha}[x_{i-1}] \cdot \boldsymbol{\alpha}[x_i])^2}. \tag{37}$$

Using (29), (30), and (32), expression (37) can be rewritten as:

$$p_{\parallel 1} = \frac{u_{11}\boldsymbol{\alpha}[x_{i-1}] \cdot (\boldsymbol{\alpha}[x_i] \times \boldsymbol{\beta}[x_i]) + u_{12}\boldsymbol{\alpha}[x_i] \cdot (\boldsymbol{\alpha}[x_{i-1}] \times \boldsymbol{\beta}[x_{i-1}])}{u_{11}^2 u_{22}^2}. \tag{38}$$

Using the property: $\boldsymbol{a} \cdot (\boldsymbol{b} \times \boldsymbol{c}) = \boldsymbol{c} \cdot (\boldsymbol{a} \times \boldsymbol{b})$, and relation (31), (38) reduces to:

$$p_{\parallel 1} = \frac{u_{11}^2 u_{22} \boldsymbol{\beta}[x_i] \cdot \boldsymbol{r}_3 - u_{11}u_{12}u_{22}\boldsymbol{\beta}[x_{i-1}] \cdot \boldsymbol{r}_3}{u_{11}^2 u_{22}^2}, \tag{39}$$

which finally can be rewritten as:

$$p_{\parallel 1} = \frac{1}{u_{22}} \left[(\boldsymbol{R}^{\top}\boldsymbol{B})_{32} - \frac{u_{11}}{u_{12}} (\boldsymbol{R}^{\top}\boldsymbol{B})_{31} \right]. \tag{40}$$

From (18) and (19), it follows that $p_{\parallel 1} = p_x^*$, which proves the argument. ■

References

1. Ancillao, A., Vochten, M., Verduyn, A., De Schutter, J., Aertbeliën, E.: An optimal method for calculating an average screw axis for a joint, with improved sensitivity to noise and providing an analysis of the dispersion of the instantaneous axes. Plos one **17**(10), e0275218 (2022)
2. Bertsekas, D.P.: Constrained optimization and Lagrange multiplier methods. Academic press (2014)
3. Chasles, M.: Note sur les propriétés génerales du système de deux corps semblables entr'eux et placés d'une manière quelconque dans l'espace; et sur le déplacement fini ou infiniment petit d'un corps solide libre. Bulletin des Sciences Mathématiques, Férussac **14**, 321–326 (1830)
4. Cohen, T., Welling, M.: Group equivariant convolutional networks. In: Proceedings of The 33rd International Conference on Machine Learning. vol. 48, pp. 2990–2999. PMLR, New York, New York, USA (20–22 Jun 2016), https://proceedings.mlr.press/v48/cohenc16.html
5. De Schutter, J.: Invariant description of rigid body motion trajectories. ASME Journal of Mechanisms and Robotics **2**(1) (2010). https://doi.org/10.1115/1.4000524
6. Golub, G.H., Van Loan, C.F.: Matrix computations. The Johns Hopkins University Press, Baltimore, MD (2013)
7. Guo, Y., Li, Y., Shao, Z.: Rrv: A spatiotemporal descriptor for rigid body motion recognition. IEEE Transactions on Cybernetics **48**(5), 1513–1525 (2018). https://doi.org/10.1109/TCYB.2017.2705227
8. Iosifidis, A., Tefas, A., Pitas, I.: View-invariant action recognition based on artificial neural networks. IEEE Transactions on Neural Networks and Learning Systems **23**(3), 412–424 (2012). https://doi.org/10.1109/TNNLS.2011.2181865
9. Lee, D., Soloperto, R., Saveriano, M.: Bidirectional invariant representation of rigid body motions and its application to gesture recognition and reproduction. Autonomous Robots **42**, 1–21 (2018). https://doi.org/10.1007/s10514-017-9645-x
10. Lynch, K.M., Park, F.C.: Modern robotics. Cambridge University Press, Cambridge (2017)
11. Murray, R.M., Sastry, S.S., Zexiang, L.: A Mathematical Introduction to Robotic Manipulation. CRC Press, Inc., Boca Raton, FL, USA, 1st edn. (1994)
12. Poinsot, L.: Sur la composition des moments et la composition des aires. Journal de l'Ecole Polytechnique **6**(13), 182–205 (1806)
13. Pöppelbaum, J., Schwung, A.: Predicting rigid body dynamics using dual quaternion recurrent neural networks with quaternion attention. IEEE Access **10**, 82923–82943 (2022). https://doi.org/10.1109/ACCESS.2022.3196340
14. Roth, B.: Finding geometric invariants from time-based invariants for spherical and spatial motions. Journal of Mechanical Design **127**(2), 227–231 (2005). https://doi.org/10.1115/1.1828462
15. Schönemann, P.H.: A generalized solution of the orthogonal procrustes problem. Psychometrika **31**(1), 1–10 (1966)
16. Stramigioli, S., Bruyninckx, H.: Geometry and screw theory for robotics. Tutorial during ICRA **2001**, 75 (2001)
17. Verduyn, A., Aertbeliën, E., Maes, G., De Schutter, J., Vochten, M.: Bilts: A bi-invariant similarity measure for robust object trajectory recognition under reference frame variations (2025), https://arxiv.org/abs/2405.04392

18. Verduyn, A., Bruyninckx, H., Vochten, M., De Schutter, J.: Invariant Motion Trajectory Similarity Measurement: Resolving Singularity Issues for Robust Invariant Rigid-Body Motion Recognition. Doctoral dissertation, Arenberg Doctoral School, KU Leuven, Leuven, Belgium (2025), https://lirias.kuleuven.be/4238129?&lang=en
19. Verduyn, A., Vochten, M., De Schutter, J.: Enhancing motion trajectory segmentation of rigid bodies using a novel screw-based trajectory-shape representation. 2024 IEEE International Conference on Robotics and Automation (ICRA) pp. 7179–7185 (2024). https://doi.org/10.1109/ICRA57147.2024.10610030
20. Verduyn, A., Vochten, M., De Schutter, J.: Enhancing hand palm motion gesture recognition by eliminating reference frame bias via frame-invariant similarity measures. 2025 IEEE 21st International Conference on Automation Science and Engineering (CASE) pp. 866–873 (2025). https://doi.org/10.1109/CASE58245.2025.11163910
21. Vochten, M., De Laet, T., De Schutter, J.: Comparison of rigid body motion trajectory descriptors for motion representation and recognition. 2015 IEEE International Conference on Robotics and Automation (ICRA) pp. 3010–3017 (2015). https://doi.org/10.1109/ICRA.2015.7139612
22. Vochten, M., De Laet, T., De Schutter, J.: Generalizing demonstrated motion trajectories using coordinate-free shape descriptors. Robotics and Autonomous Systems **122**, 103291 (2019). https://doi.org/10.1016/j.robot.2019.103291
23. Vochten, M., Mohammadi, A.M., Verduyn, A., De Laet, T., Aertbeliën, E., De Schutter, J.: Invariant descriptors of motion and force trajectories for interpreting object manipulation tasks in contact. IEEE Transactions on Robotics **39**(6), 4892–4912 (2023). https://doi.org/10.1109/tro.2023.3309230
24. Wang, P., Li, W., Gao, Z., Zhang, J., Tang, C., Ogunbona, P.O.: Action recognition from depth maps using deep convolutional neural networks. IEEE Transactions on Human-Machine Systems **46**(4), 498–509 (2016). https://doi.org/10.1109/THMS.2015.2504550
25. Yao, G., Youfu, L., Zhanpeng, S.: Dsrf: A flexible trajectory descriptor for articulated human action recognition. Pattern Recognition **76**, 137–148 (2018). https://doi.org/https://doi.org/10.1016/j.patcog.2017.10.034