

Invariant Descriptors of Motion and Force Trajectories for Interpreting Object Manipulation Tasks in Contact

Maxim Vochten^{1,3,4}, Ali Mousavi Mohammadi^{1,3}, Arno Verduyn^{1,3}, Tinne De Laet², Erwin Aertbeliën¹, and Joris De Schutter¹

Abstract—Invariant descriptors of point and rigid-body motion trajectories have been proposed in the past as representative task models for motion recognition and generalization. Currently, no invariant descriptor exists for representing force trajectories, which appear in contact tasks. This paper introduces invariant descriptors for force trajectories by exploiting the duality between motion and force. Two types of invariant descriptors are presented depending on whether the trajectories consist of screw or vector coordinates. Methods and software are provided for robustly calculating the invariant descriptors from noisy measurements using optimal control. Using experimental human demonstrations of 3D contour following and peg-on-hole alignment tasks, invariant descriptors are shown to result in task representations that do not depend on the calibration of reference frames or sensor locations. The tuning process for the optimal control problems is shown to be fast and intuitive. Similar to motions in free space, the proposed invariant descriptors for motion and force trajectories may prove useful for the recognition and generalization of constrained motions, such as during object manipulation in contact.

Index Terms—invariance, screw theory, trajectory representation, optimal control, contact tasks

I. INTRODUCTION

IN *human-robot interaction* (HRI) there is a need for generalizable task models that allow robots to reactively adapt the robot’s execution and to robustly recognize human actions in continuously changing environments. For execution purposes, these models should be easily adaptable to deal with obstacles, different starting/ending/via points, robot platform constraints, and human-robot interaction. For recognition purposes, these models should be sufficiently general to recognize the human’s action at different locations in space, from different viewpoints, and with different execution styles (e.g. varying scale and motion profile). The focus of this paper is on extracting task representations that involve the manipulation of objects in contact, where not only the relative motion between objects but also interaction forces are relevant as input for the task modeling process. Typical applications include surface following tasks and assembly tasks.

This result is part of 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).

¹ Department of Mechanical Engineering and Flanders Make at KU Leuven, 3001 Leuven, Belgium.

² Faculty of Engineering Science, KU Leuven, 3001 Leuven, Belgium.

³ These authors contributed equally to this work.

⁴ Corresponding author (maxim.vochten@kuleuven.be).

Task models in HRI are commonly learned from one or multiple human demonstrations of the task using *Learning by Demonstration* [1]. The challenge is to achieve a representative task model without the need for a high number of demonstrations.

One approach to achieve representative task models from few demonstrations is to exploit invariance in the task. Here, *invariance* refers to properties of the task that remain unchanged under certain transformations such as changes in reference frame. Invariance has proven useful in various fields of robotics. Typically, the objectives were: (1) to ensure that exactly the same task execution or behavior of a robot system was obtained within its workspace, and (2) to ensure that the quantitative characterization of such robot action using a metric resulted in exactly the same number(s) regardless of where and how the robot action was recorded or measured. Below, some examples of the importance and use of invariant solutions in different robotics contexts are briefly reported.

For the *control* of free-space robot motion such as mobile robots and UAVs, invariance was achieved by representing the trajectory tracking error in such a way that the same closed-loop behavior for trajectory tracking and disturbance rejection was obtained regardless of the direction in which the robot was moving [2]. For constrained motion, i.e., motion in contact, which is the subject of this paper, the risk of obtaining non-invariant control solutions is more pronounced since both motion (translation and rotation) and interaction wrenches (i.e. forces and moments) have to be controlled simultaneously. For example, as pointed out in [3], it does not make sense to define *orthogonality* between two twists (i.e. translational and rotational velocities), or between two contact wrenches, because of (1) dimensional inconsistency, (2) dependence on the choice of units, and (3) dependence on the choice of the origin of the coordinates. In contrast, *reciprocity* between a twist and a wrench is an invariant concept. In [4], [5] a firm geometric foundation for kinestatics and invariant hybrid force/position control was laid based on screw theory. Taking this into consideration, invariant solutions were proposed, for example to obtain dynamic models of redundant and constrained robot manipulators [6], or in applications of hybrid force/position control [7].

In *estimation*, invariance refers to obtaining the same values for the estimated quantities regardless of the reference frame in which the estimator’s states are represented. For example, the Invariant Extended Kalman Filter [8] formulated the state

correction term in an invariant way using Lie group theory and showed a better convergence of the estimation compared to a standard Extended Kalman filter with a linear correction term. The Invariant Extended Kalman Filter was applied, for instance, to attitude estimation for UAVs [9].

In *motion generation*, invariance refers to obtaining exactly the same physical desired motion trajectory regardless of the reference frame in which the trajectory coordinates are expressed. Human reaching motions, represented as point trajectories, were found to correspond well to a minimum-jerk profile, an invariant property that was shown to be equivalent to having a constant equi-affine curvature [10]. Based on this property, new trajectories can be generated or adapted from reference trajectories in an invariant way [11]. This and other invariant properties such as shape preservation were considered for the deformation of trajectories using global transformations [12]. For rigid-body motion, approaches were proposed where the trajectories are either generated in a left-invariant way, meaning independent of the world reference frame, or in a right-invariant way, meaning independent of the body reference frame [13]. This was also taken into account for the adaptation of reference or demonstrated rigid-body trajectories [14], [15].

Invariance in *action recognition* refers to recognizing actions regardless of where and how they are recorded. This can be done by comparing the recorded action instance and its model using invariant distance metrics. For example, local invariant signatures were proposed for describing and recognizing object outlines [16] and trajectories [17]. These invariant representations were shown to be invariant to occlusion, rotation, translation, and scaling.

These previous works showed that invariant representations offer many advantages that are relevant for different robotics applications. In the remainder of the paper we focus on invariant descriptors for trajectories. These descriptors can be used to build representative invariant models of a demonstrated task in a human-robot interaction context with fewer demonstrations compared to conventional methods using coordinate-based descriptors. In addition, thanks to their invariant properties, these models enable robust action recognition in different recording contexts and exhibit good generalization properties in motion generation.

A. Invariant trajectory descriptors for motion

Invariant trajectory descriptors are features of the trajectory that remain unchanged under certain transformations of the original trajectory coordinates, such as changes in reference frame, scale, and time profile. Invariant descriptors can be categorized in many types [18] depending on the features that are extracted from the curve. Many have been proposed, both for point trajectories [16], [19]–[22] and for rigid-body trajectories [23]–[27].

This paper focuses on a particular type of invariant descriptor, referred to as *differential invariants* [16], [18]. These invariants correspond to differential-geometric properties of the trajectory and provide a local invariant representation that is robust to occlusion and segmentation errors. Furthermore, they

allow to reconstruct and generate new trajectories from the invariant descriptor by integrating corresponding differential equations. The definition of such local differential-geometric invariants can be seen as an application of *Cartan's method of moving frames* [28], which consists of defining a *natural moving frame* in the considered manifold, after which the derivative of the frame is shown to correspond to a set of geometric invariants.

For point motion in Euclidean space, the differential invariants correspond to curvature and torsion defined in a moving frame referred to as the *Frenet-Serret frame*. Originally introduced in vision for characterizing and recognizing objects by their outlines [16], [29], the same concepts of curvature and torsion were applied for describing and recognizing point motion trajectories [17], [20].

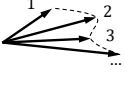
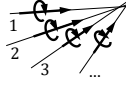
For rigid-body motion, differential invariants were introduced based on the concept of the screw axis in screw theory [25]. A moving frame was defined on the screw axis of which the motion was described by a set of invariants referred to as *screw invariants*. These invariants are *bi-invariant*, meaning independent of both the world reference frame and the reference frame attached to the body. An alternative invariant descriptor was proposed by defining two separate Frenet-Serret frames for the position and orientation of the rigid body [26], [27], resulting in a simpler set of formulas. However, invariance for the choice of reference point for translation (the body frame's origin) was lost.

Invariant trajectory descriptors for motion have, for the large majority, been applied in motion recognition [20], [26], [27]. Trajectory generation using invariant descriptors has also been explored. The invariant descriptors are capable of both reconstructing the original trajectory [25], as well as reproducing the trajectory at new locations and in different directions by applying the correct initial values when integrating the differential equations [30]. Building further on that, trajectory adaptation methods based on optimal control and invariant descriptors were proposed [15]. The goal was to adapt a reference trajectory to comply with new constraints while minimizing deviation from the invariant descriptor of the reference trajectory. This was shown to result in excellent extrapolation capabilities [15].

Calculating differential invariants from measurement data is challenging due to noise-sensitivity and singularities. *Singularities* are defined as instances along the trajectory where components of the descriptor are arbitrarily defined. Near these singularities, small variations in the trajectory (e.g., due to measurement noise or irrelevant human variations) will result in large variations in the ill-defined components of the descriptor. To remedy this, efforts have been spent to develop robust approaches to calculate invariant descriptors, ranging from discrete approximations of the descriptor [16], [17], [20], [26] to averaging the descriptor over segments of the curve [31]. More recently, approaches based on optimal control were introduced for calculating point trajectory descriptors [32] and rigid-body trajectory descriptors [33]. Using optimal control, the local descriptor is calculated in a window of measurements while regularization terms deal with singularities and noise by smoothing the invariants in a local neighborhood. A

TABLE I

OVERVIEW OF TRAJECTORY TYPES OF WHICH INVARIANT DESCRIPTORS CAN BE CALCULATED.

Trajectory	Applied to motion	Applied to force
vectors c $\in \mathbb{R}^3$ 	rot. velocity ω	force f
	transl. velocity v	moment m
screws s $\in \mathbb{R}^6$ 	<i>twist:</i>	<i>wrench:</i>
	$t = \begin{pmatrix} \omega \\ v \end{pmatrix}$	$w = \begin{pmatrix} f \\ m \end{pmatrix}$

disadvantage of current optimal control-based approaches is their dependency on many tuning parameters for weighting the trajectory accuracy cost and regularization costs, and their long calculation times.

B. Paper objective and contributions

Up till now, invariant trajectory descriptors for interaction forces with similar benefits as the invariant descriptors for motion have not been proposed. This paper's main objective is to extend the existing concepts and methodology for motion invariants towards interaction forces by exploiting duality between motion and force. The resulting invariant descriptors can be used to model motion and force trajectories that occur in motion-in-contact tasks. Table I summarizes this duality for the two cases where motion and force trajectories are either represented with vector coordinates or with screw coordinates.

The main contribution of this paper is the introduction of two types of local differential invariant descriptors for force and moment trajectories. The first type is referred to as the *vector invariant descriptor* for vector trajectories and is applied to force and moment vectors separately. The second type is referred to as the *screw axis invariant descriptor* for screw trajectories and is applied to a screw wrench, combining force and moment in one entity. The duality is shown between the new invariant descriptors for force and moment and the existing invariant descriptors for motion: the screw invariants for motion [25] and the vector invariants for translation and rotation [26], [27]. Furthermore, we explain how to calculate the invariant descriptors for motion and force using numerical optimal control schemes, which filter out the effects of measurement noise and improve the behavior of the descriptor near singularities. Triggered by the higher noise levels of measured force and moment trajectories compared to measured motion trajectories, we improved the formulation of these numerical optimal control schemes compared to [33], resulting in much more intuitive and faster tuning as well as increased robustness of the calculation. The invariant properties of the proposed descriptors were experimentally verified using a human-demonstrated 3D contour following task and a peg-on-hole alignment task. All software used in the validation experiment, including data, has been made publicly available [34].

The remainder of the paper is structured as follows. Section II provides the necessary background on rigid-body kinematics and statics. Sections III and IV introduce local invariant

TABLE II

OVERVIEW OF NOTATION USED THROUGHOUT THE PAPER. (IN THIS TABLE a AND b ARE PLACEHOLDERS FOR TWO FRAMES OF INTEREST.)

Variable	Description
$\{a\}$	reference frame named "a" attached to a rigid body or the world
${}_b^a\mathbf{R}$	3×3 rotation matrix representing the orientation of frame $\{b\}$ with respect to frame $\{a\}$
${}_a^b\mathbf{T}$	4×4 homogeneous transformation matrix representing the orientation and position of frame $\{b\}$ with respect to frame $\{a\}$
${}_a\omega$	3×1 rotational velocity vector of the rigid body expressed in $\{a\}$
${}_a v^b$	3×1 translational velocity vector of the rigid body at the origin of $\{b\}$ and expressed in $\{a\}$
${}_a f$	3×1 force vector acting on the rigid body expressed in $\{a\}$
${}_a m^b$	3×1 moment vector acting on the rigid body at the origin of $\{b\}$ and expressed in $\{a\}$
${}_a t^a$	6×1 screw twist of rigid body with the first three elements the rotational velocity vector ${}_a\omega$ and the last three elements the translational velocity vector ${}_a v^a$
${}_a w^a$	6×1 screw wrench acting on the rigid body with the first three elements the force vector ${}_a f$ and the last three elements the moment vector ${}_a m^a$
${}_a^b\mathbf{S}$	6×6 screw transformation matrix that, when multiplied with a screw twist or wrench, transforms the screw from $\{b\}$ to $\{a\}$

descriptors in a general form, resulting in existing descriptors when applied to motion trajectories and new descriptors when applied to force and moment trajectories. Sections III and IV deal with vector and screw invariant descriptors, respectively. Section V highlights a practical 3D contour following task, while Section VI examines a peg-on-hole alignment task. Section VII provides a discussion about the benefits and limitations of the proposed concepts and methodology, it points at future work and presents a conclusion.

II. PRELIMINARIES

This section reviews essential background and introduces notation as summarized in Table II.

A. Rigid-body kinematics

The motion trajectory of a rigid body is commonly represented by attaching a reference frame $\{b\}$ to the *body* and expressing the position and orientation of this frame as a function of time t with respect to a fixed reference frame $\{w\}$, also referred to as the *world* reference frame. For all variables introduced below, the explicit dependency on time t is omitted to simplify the expressions.

The position is represented by the 3-dimensional *displacement vector* ${}_w^b p$ from the origin of $\{w\}$ to the origin of $\{b\}$. The orientation is represented by the *rotation matrix* ${}_w^b \mathbf{R}$, which expresses the coordinates of the unit vectors of the moving body frame $\{b\}$ with respect to the world frame $\{w\}$:

$${}_w^b \mathbf{R} = [e_x \quad e_y \quad e_z]. \quad (1)$$

Position and orientation can be combined into the *homogeneous transformation matrix* or *pose matrix* ${}_w^b \mathbf{T}$:

$${}_w^b \mathbf{T} = \begin{bmatrix} {}_w^b \mathbf{R} & {}_w^b p \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}, \quad (2)$$

which is part of the $SE(3)$ group according to Lie theory. The coordinates of the pose matrix ${}_w^b \mathbf{T}$ depend on the choice of

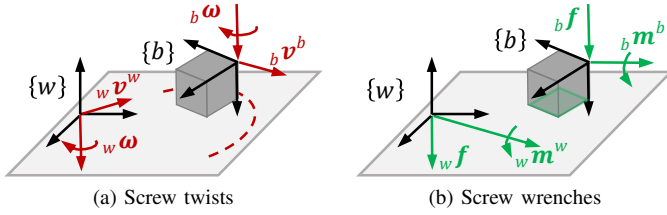


Fig. 1. Visualization of screw twist $\mathfrak{t} = (\omega^T \ v^T)^T$ and screw wrench $\mathfrak{w} = (f^T \ m^T)^T$ in the world frame $\{w\}$ and in the body frame $\{b\}$.

the fixed reference frame $\{w\}$ in the world and the reference frame $\{b\}$ on the rigid body.

The velocity of a rigid body can be fully characterized by two 3-dimensional vectors: the rotational velocity vector ω and the translational velocity vector v of the origin of the considered reference frame. These vectors are summarized with a 6-dimensional screw [35], the *twist* $\mathfrak{t} = \begin{pmatrix} \omega \\ v \end{pmatrix}$.

To define the time-derivative of the homogeneous transformation matrix, $\dot{T} = \frac{dT}{dt}$, we first introduce the operator $[\cdot]_{\times}$. When applied to the vector $\omega = [\omega_x \ \omega_y \ \omega_z]^T$, it results in a 3×3 skew-symmetric matrix:

$$[\omega]_{\times} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}, \quad (3)$$

and when applied to a screw \mathfrak{t} , it results in the following 4×4 *twist matrix* $[\mathfrak{t}]_{\times}$, which is an element of the Lie algebra $se(3)$:

$$[\mathfrak{t}]_{\times} = \begin{bmatrix} [\omega]_{\times} & v \\ \mathbf{0}_{1 \times 3} & 0 \end{bmatrix}. \quad (4)$$

The definition of the time-derivative of the homogeneous transformation matrix \dot{T} depends on the choice of reference frame in which the twist \mathfrak{t} is expressed. Choosing the body frame $\{b\}$, the *body twist* ${}_b\mathfrak{t}^b$ contains the rotational velocity vector ${}_b\omega$ expressed in $\{b\}$ and the translational velocity vector ${}_b v^b$ of the origin of $\{b\}$ and expressed in $\{b\}$ (see Fig. 1a). The derivative of the pose matrix ${}_w\dot{T}$ is then defined as:

$${}_w\dot{T} = {}_wT [{}_b\mathfrak{t}^b]_{\times}, \quad \text{with} \quad [{}_b\mathfrak{t}^b]_{\times} = \begin{bmatrix} [{}_b\omega]_{\times} & {}_b v^b \\ \mathbf{0}_{1 \times 3} & 0 \end{bmatrix}. \quad (5)$$

The twist ${}_b\mathfrak{t}^b$ has the property of being invariant for a change of $\{w\}$ (*left-invariance*).

Alternatively, choosing the world frame $\{w\}$, the spatial twist ${}_w\mathfrak{t}^w$ consists of the rotational velocity vector ${}_w\omega$ expressed in $\{w\}$, and the translational velocity vector ${}_w v^w$ of the point on the rigid body that instantaneously coincides with the origin of $\{w\}$ expressed in $\{w\}$. The pose derivative ${}_w\dot{T}$ is then defined as:

$${}_w\dot{T} = [{}_w\mathfrak{t}^w]_{\times} {}_wT, \quad \text{with} \quad [{}_w\mathfrak{t}^w]_{\times} = \begin{bmatrix} [{}_w\omega]_{\times} & {}_w v^w \\ \mathbf{0}_{1 \times 3} & 0 \end{bmatrix}. \quad (6)$$

where $[{}_w\mathfrak{t}^w]_{\times}$ is now multiplied on the left of ${}_wT$ since the twist is expressed in $\{w\}$. The twist ${}_w\mathfrak{t}^w$ has the property of being invariant for a change of $\{b\}$ (*right-invariance*).

The relation between twists ${}_w\mathfrak{t}^w$ and ${}_b\mathfrak{t}^b$ is provided by the 6×6 *screw transformation matrix* $S({}_wT)$ (also referred to as

the *adjoint matrix* [36]), which is constructed from elements of the pose matrix ${}_wT$:

$${}_w\mathfrak{t}^w = S({}_wT) {}_b\mathfrak{t}^b = \begin{bmatrix} {}_wR & \mathbf{0}_{3 \times 3} \\ [{}_w p]_{\times} {}_wR & {}_wR \end{bmatrix} {}_b\mathfrak{t}^b. \quad (7)$$

Twists ${}_b\mathfrak{t}^b$ and ${}_w\mathfrak{t}^w$ represent first-order kinematics of the trajectory and are invariant to the choice of world frame and body frame, respectively. But neither ${}_w\mathfrak{t}^w$ nor ${}_b\mathfrak{t}^b$ is invariant for the choice of *both* reference frames $\{w\}$ and $\{b\}$. Trajectory descriptors that are invariant to *both* reference frames (also referred to as coordinate-invariance) will be discussed in Sections III and IV.

B. Duality between twist and wrench as general screws

All forces and moments acting on a rigid body can be reduced to a single resultant force vector and a single resultant moment vector. One option is to express these vectors in the body frame: ${}_b f$ represents the resultant force expressed in $\{b\}$, and ${}_b m^b$ represents the moment with respect to the origin of $\{b\}$ and with coordinates expressed in $\{b\}$ (see Fig. 1b). Similar to the *twist*, a compact notation for the resulting force and moment is the 6-dimensional *screw wrench*:

$${}_b\mathfrak{w}^b = \begin{pmatrix} {}_b f \\ {}_b m^b \end{pmatrix}. \quad (8)$$

Transforming the wrench from the body frame $\{b\}$ to the world frame $\{w\}$ is done using the same 6×6 *screw transformation matrix* $S({}_wT)$ as for the twist:

$${}_w\mathfrak{w}^w = S({}_wT) {}_b\mathfrak{w}^b. \quad (9)$$

This analogy between twists and wrenches is also referred to as the *duality between velocity and force*. Both twist and wrench can be interpreted as a 6-dimensional *screw* [35], meaning they can always be represented by a vector lying along a line in space and the moment of that vector around the line.

C. Time-invariance using a geometric progress variable

In the previous subsections, motion and force trajectories were assumed to be a function of time t . To obtain a *time-invariant trajectory representation*, i.e., a trajectory representation that is invariant with respect to the motion profile, a geometric progress variable ξ must be chosen.

For tasks where translation is dominant, such as in contour following, the arc length is a natural choice. The arc length is calculated by integrating the norm of the translational velocity vector v over time. For tasks where rotation is dominant, such as in opening a door, a natural progress variable would be the integral of the norm of the rotational velocity vector ω . If translation and rotation are both important, then a progress parameter can be defined combining the translation and rotation, for example, using a linear combination [25]:

$$\xi(t) = \int_0^t w \|\omega\| + (1-w) \|v\| dt, \quad (10)$$

where w weights the relative contribution. In [25], further details are provided for obtaining a progress value that is

invariant for the choice of reference point on the body for the translation \mathbf{v} .

For the pose \mathbf{T} and wrench \mathbf{w} , the reparameterization from time t to progress ξ is straightforward:

$$\mathbf{T}(\xi) = \mathbf{T}(t(\xi)) \quad \text{and} \quad \mathbf{w}(\xi) = \mathbf{w}(t(\xi)). \quad (11)$$

In practice, since measurement data is often only available in a discretely sampled form, the reparameterization is achieved through numerical interpolation.

For the velocity, the reparameterization from time t to progress ξ requires an additional step since velocities represent derivatives of the trajectory. As an example, the reparameterization of the translational velocity vector \mathbf{v} , as the derivative of the position \mathbf{p} , can be worked out as:

$$\mathbf{v}(\xi) = \frac{d\mathbf{p}(\xi)}{d\xi} = \frac{d\mathbf{p}(t(\xi))}{dt(\xi)} \cdot \frac{dt(\xi)}{d\xi} = \frac{\mathbf{v}(t(\xi))}{\dot{\xi}(t(\xi))}, \quad (12)$$

where the chain rule of differentiation was applied in the second equality to obtain the relation with time t , and the *progress rate* $\dot{\xi} = \frac{d\xi}{dt}$ is the derivative of the progress with respect to time. Similar derivations can be made for the rotational velocity vector $\boldsymbol{\omega}$ and the twist $\boldsymbol{\mathcal{t}}$:

$$\boldsymbol{\omega}(\xi) = \frac{\boldsymbol{\omega}(t(\xi))}{\dot{\xi}(t(\xi))} \quad \text{and} \quad \boldsymbol{\mathcal{t}}(\xi) = \frac{\boldsymbol{\mathcal{t}}(t(\xi))}{\dot{\xi}(t(\xi))}. \quad (13)$$

Equations (11)-(13) can be used in the other direction to apply a motion profile $\xi(t)$ and obtain the pose, wrench, translational velocity, rotational velocity or twist as a function of time. This is relevant, for example, when a trajectory was planned in the progress domain ξ and needs to be executed with a specific timing $\xi(t)$. A special choice for the progress variable is one where $\xi = 0$ at the beginning of the trajectory and $\xi = 1$ at the end of the trajectory. This results in a reparameterization that is independent of the scale of the trajectory [25].

III. INVARIANT DESCRIPTOR FOR VECTOR TRAJECTORIES

This section derives a local invariant descriptor for a general *vector trajectory* $\mathbf{c}(\xi)$. The trajectory $\mathbf{c}(\xi)$ is assumed to be time-invariant, i.e. parameterized as a function of a geometric progress ξ as discussed in Section II-C. To keep the formulas concise, the explicit dependency on ξ is omitted in the following formulas. The derivative of \mathbf{c} with respect to the progress ξ is represented using the prime notation: $\mathbf{c}' = \frac{d\mathbf{c}}{d\xi}$.

A. Definition of invariants

The vector invariants are based on a generalization of the *Frenet-Serret equations* for translation [37], where \mathbf{c} corresponded to the translational velocity vector. While in [26], [27] the Frenet-Serret equations were extended to rotational velocity trajectories, the aim here is to exploit the duality between velocity and force so that the invariants can be presented in a general form, applicable to rotational and translational velocity trajectories, as well as to force and moment trajectories.

The vector invariants are defined and expressed in a local moving frame, referred to as the *Frenet-Serret frame* (FS frame), which is constructed as follows. Its first axis, the

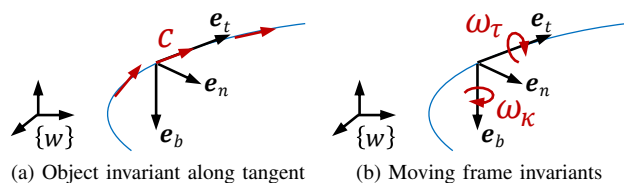


Fig. 2. Invariant descriptor for a vector trajectory: (a) object invariant c defined in the moving FS frame given by the unit vectors \mathbf{e}_t , \mathbf{e}_n and \mathbf{e}_b ; (b) moving frame invariants ω_τ and ω_κ . The figure depicts the special case where \mathbf{c} corresponds to a translational velocity vector with respect to reference frame $\{\mathcal{W}\}$ and where, for visualization purposes, the moving frame is depicted with its origin chosen at the corresponding position along the point trajectory (indicated in blue).

tangent \mathbf{e}_t , corresponds to the unit vector in the direction of \mathbf{c} . The second axis corresponds to the normal \mathbf{e}_n of the trajectory. The third axis, the binormal \mathbf{e}_b , follows directly from the cross-product of \mathbf{e}_t and \mathbf{e}_n . These unit vectors are defined as follows from \mathbf{c} and \mathbf{c}' ¹:

$$\mathbf{e}_t = \frac{\mathbf{c}}{\|\mathbf{c}\|}, \quad \mathbf{e}_n = \frac{(\mathbf{c} \times \mathbf{c}') \times \mathbf{c}}{\|(\mathbf{c} \times \mathbf{c}') \times \mathbf{c}\|}, \quad \mathbf{e}_b = \mathbf{e}_t \times \mathbf{e}_n. \quad (14)$$

In differential geometry, the linear span of \mathbf{e}_t and \mathbf{e}_n is referred to as the *osculating plane*, i.e. the plane in which the vector \mathbf{c} rotates instantaneously, while \mathbf{e}_b represents the axis about which this rotation occurs. The complete orientation of the FS frame is expressed using a rotation matrix $\tilde{\mathbf{R}}$:

$$\tilde{\mathbf{R}} = [\mathbf{e}_t \quad \mathbf{e}_n \quad \mathbf{e}_b], \quad (15)$$

where the tilde \sim indicates it is a local moving frame, in this case the FS frame. See Fig. 2 for a visualization of the frame.

The invariants are then defined as follows. The first invariant c is the magnitude of the vector \mathbf{c} along the tangent:

$$c = \mathbf{c} \cdot \mathbf{e}_t. \quad (16)$$

This invariant is referred to as the *object invariant* since it relates to the object. For example, c can represent the magnitude of the translational or rotational velocity vector of the object, or the magnitude of the force or moment vector applied to the object. The vector \mathbf{c} can be retrieved from the object invariant c using:

$$\mathbf{c} = \tilde{\mathbf{R}} \tilde{\mathbf{c}}, \quad \text{with} \quad \tilde{\mathbf{c}} = (c \quad 0 \quad 0)^T. \quad (17)$$

The second and third invariants ω_κ and ω_τ govern the first-order kinematics of the local FS frame²:

$$\tilde{\mathbf{R}}' = \tilde{\mathbf{R}} \begin{bmatrix} 0 & -\omega_\kappa & 0 \\ \omega_\kappa & 0 & -\omega_\tau \\ 0 & \omega_\tau & 0 \end{bmatrix}, \quad (18)$$

where ω_κ is known as the curvature rate and ω_τ as the torsion rate, with the following explicit formulas [37]:

$$\omega_\kappa = \frac{\|\mathbf{c} \times \mathbf{c}'\|}{\|\mathbf{c}\|^2}, \quad \omega_\tau = \frac{((\mathbf{c} \times \mathbf{c}') \times (\mathbf{c} \times \mathbf{c}'')) \cdot \mathbf{c}}{\|\mathbf{c} \times \mathbf{c}'\|^2 \|\mathbf{c}\|}. \quad (19)$$

¹To be consistent with the common notation of the FS frame and equations, the authors adapted the order of the columns \mathbf{e}_t , \mathbf{e}_n , and \mathbf{e}_b of $\tilde{\mathbf{R}}$ in their previous papers such as [15], [25], [33]. Therefore, this paper has a slightly different notation than the authors' previous works. This adaptation affects only the appearance of the equations, not the concept.

²The relation between (18) and the Frenet-Serret equation as usually presented in literature is derived in the appendix.

For a naturally parameterized point trajectory, with \mathbf{c} the point's velocity, ω_κ and ω_τ correspond to the local curvature κ and torsion τ of the point's trajectory.

Equation (18) can be written compactly using (3):

$$\tilde{\mathbf{R}}' = \tilde{\mathbf{R}} [\mathbf{i}]_\times, \quad \text{with} \quad \mathbf{i} = \begin{pmatrix} \omega_\tau \\ 0 \\ \omega_\kappa \end{pmatrix}, \quad (20)$$

where \mathbf{i} is referred to as the *moving frame invariant* since it models the first-order kinematics of the moving frame.

Only the orientation is relevant for the FS frame, not its position. Therefore it is sometimes referred to as an *orientation frame* [38]. For visualization purposes, its evolution is often shown by representing it at the point on the motion trajectory associated with the progress variable ξ . This was done in Fig. 2, which shows the definition of the moving frame and the corresponding invariants for a point trajectory.

Singularities: Singularities are defined as instances where axes of the moving frame and the corresponding invariants are not uniquely defined anymore. A first type of singularity is when $\mathbf{c} = \mathbf{0}$. The object invariant c is then zero, while the moving frame invariants ω_κ and ω_τ are undefined. In other words, the complete moving frame is arbitrary.

A second type of singularity is when the vector \mathbf{c} has an unchanging orientation at some instance ($\mathbf{c} \times \mathbf{c}' = \mathbf{0}$), remaining parallel to itself. The first moving frame invariant ω_κ is then zero while the second moving frame invariant ω_τ is undefined. In other words, the tangent of the moving frame is well-defined, but the normal and binormal axes are arbitrary.

Invariant properties: If the trajectory $\mathbf{c}(\xi)$ is expressed as a function of a geometric progress variable ξ as explained in Section II-C, then the descriptor (c, \mathbf{i}) is also *time-invariant*. Furthermore, the descriptor is *invariant with respect to the choice of world frame* $\{w\}$ (both origin and orientation) in which the coordinates of the given vector trajectory $\mathbf{c}(\xi)$ are expressed. It is evident that a change of $\{w\}$ merely results in the same change of orientation for the vectors \mathbf{c} , \mathbf{c}' , and \mathbf{c}'' , resulting in the same object invariant c , the same FS frame for all ξ , and hence also the same moving frame invariant \mathbf{i} . If the coordinates of \mathbf{c} are instead expressed in $\{b\}$, the descriptor is still *invariant for changes in orientation of the body frame* $\{b\}$. However, not all considered instances of the descriptor are invariant with respect to the origin of body frame $\{b\}$. As pointed out in [27], the translational velocity vector \mathbf{v} depends on the reference point on the moving object chosen to express its velocity, hence on the origin of $\{b\}$. Similarly, the moment vector \mathbf{m} depends on the reference point for expressing the moment acting on the object, hence on the origin of $\{b\}$. The dependency of \mathbf{v} and \mathbf{m} , and hence also of their invariant descriptors, on the origin of $\{b\}$ is a disadvantage compared to the screw invariants discussed in Section IV, which do not exhibit this dependency on the chosen reference point on the moving body.

Trajectory reconstruction: Reconstructing the trajectory $\mathbf{c}(\xi)$ from the invariant descriptor $(\tilde{\mathbf{c}}(\xi), \mathbf{i}(\xi))$ requires integrating the differential equation (20) from an initial value of the moving frame $\tilde{\mathbf{R}}$. In general, there exists no closed-form solution for the integration. Instead, the problem is discretized

so that $\tilde{\mathbf{R}}_{k+1}$ at sample $k+1$ is found from $\tilde{\mathbf{R}}_k$ at sample k as follows:

$$\tilde{\mathbf{R}}_{k+1} = \tilde{\mathbf{R}}_k \exp([\mathbf{i}_k]_\times \Delta\xi), \quad (21)$$

where we assume that \mathbf{i}_k remains constant over the integration step $\Delta\xi$. The matrix exponential operator $\exp(\cdot)$ maps the skew-symmetric matrix $[\mathbf{i}_k]_\times \Delta\xi$, which is part of the Lie algebra $so(3)$, into the corresponding change in orientation, which is part of the Lie group $SO(3)$. An explicit expression of the matrix exponential can be formulated using Rodrigues' rotation formula [36]. Given the initial value $\tilde{\mathbf{R}}_1$ at $k=1$, the moving frame can be reconstructed for all remaining samples $k=2\dots N$. From the moving frames $\tilde{\mathbf{R}}_k$, the corresponding vector trajectory \mathbf{c}_k at $k:1\dots N$ is found using (17):

$$\mathbf{c}_k = \tilde{\mathbf{R}}_k \tilde{\mathbf{c}}_k, \quad \forall k \in [1, N]. \quad (22)$$

Equations (21) and (22) are referred to as the *moving frame equation* and *object trajectory equation*, respectively. Table III on page 10 provides an overview of the vector invariants framework with application to translational velocity \mathbf{v} , rotational velocity $\boldsymbol{\omega}$, force \mathbf{f} , and moment \mathbf{m} trajectories.

B. Robust calculation of invariants using optimal control

Using the analytical formulas to calculate the vector invariants c and \mathbf{i} has three pitfalls: (1) near singularities, some of the invariants and axes of the moving frame $\tilde{\mathbf{R}}$ are ill-defined, hence the behavior of c and \mathbf{i} is not well-captured in such case; (2) the high-order derivatives cause sensitivity to noise, especially near singularities; and (3) the open-loop reconstructed trajectory using (21)-(22) is not guaranteed to match the original trajectory due to integration drift.

In addition, the invariants c and ω_κ are always positive by definition from (16)-(19) resulting in *frame flipping* issues: the FS frame must flip 180° when the direction of \mathbf{c} reverses or when the curve traced by the vector \mathbf{c} has an inflection point. Other definitions of moving frames exist to minimize the frame's rotation, such as Bishop frames [39], but these frames are not locally defined anymore.

To deal with these problems, we formulate an *Optimal Control Problem* (OCP) to calculate the vector invariants c and \mathbf{i} in a robust way. An OCP is a type of optimization problem that, for a given dynamical system, tries to find control inputs that minimize a specific cost function over a horizon. Here, the objective of the optimization problem is to find vector invariants that reconstruct the given measured vector trajectory. The latter can be quantified by introducing a trajectory reconstruction cost Δc_{MSE} containing the mean-squared error (MSE) between the reconstructed trajectory \mathbf{c}_k and measured trajectory $\mathbf{c}_k^{\text{meas}}$ for $k=1\dots N$:

$$\Delta c_{\text{MSE}}(\mathbf{c}, \mathbf{c}^{\text{meas}}) = \frac{1}{N} \sum_{k=1}^N \|\mathbf{c}_k - \mathbf{c}_k^{\text{meas}}\|^2. \quad (23)$$

To deal with measurement noise and the effects of singularities, a regularization cost is introduced. In general, the regularization cost can be any function of invariants, including derivatives of invariants. In [33], the terms in the regularization cost were chosen as the derivatives of all invariants and the

absolute values of the moving frame invariants. The trajectory reconstruction cost and regularization cost were added in the objective function of the OCP, and as a result they had to be weighted with appropriate weighting factors, requiring an extensive tuning process.

To reduce tuning efforts and improve interpretability of chosen parameters, this paper proposes a new formulation of the OCP. The trajectory reconstruction cost is moved to the constraints of the OCP and is bounded by a chosen tolerance value ϵ . For vector trajectories, this is worked out as follows:

$$\epsilon_c^2 \geq \Delta c_{\text{MSE}}(\mathbf{c}, \mathbf{c}^{\text{meas}}), \quad (24)$$

in which ϵ_c is interpreted as the desired tolerance on the trajectory reconstruction cost over the entire horizon. The subscript in ϵ_c refers to the type of trajectory that is considered. The value of ϵ can be set by the user based on the expected accuracy of the measurement system or the tolerable reconstruction error for the given application.

For the regularization term in the objective function, we take the squared norm of the moving frame invariants \mathbf{i} . Since \mathbf{i} signifies the rotational change of the moving frame, this will result in a more stable and smooth evolution of the moving frame $\tilde{\mathbf{R}}$. To avoid frame flipping issues, all invariants are allowed to become positive or negative, resulting in a smoother evolution of the FS frame.

The complete OCP to calculate the vector invariants $\tilde{\mathbf{c}}$ and \mathbf{i} from a given sequence of N measured vectors $\mathbf{c}_k^{\text{meas}}$ is:

$$\underset{\tilde{\mathbf{c}}_{[1]}, \tilde{\mathbf{i}}_{[1]}, \mathbf{c}_{[1]}, \tilde{\mathbf{R}}_{[1]}}{\text{minimize}} \quad \sum_{k=1}^{N-1} \|\tilde{\mathbf{i}}_k\|^2, \quad (25)$$

subject to:

$$\epsilon_c^2 \geq \Delta c_{\text{MSE}}(\mathbf{c}, \mathbf{c}^{\text{meas}}), \quad (\text{r.24})$$

$$\tilde{\mathbf{R}}_{k+1} = \tilde{\mathbf{R}}_k \exp([\tilde{\mathbf{i}}_k]_{\times} \Delta \xi), \quad \forall k \in [1, N] \quad (\text{r.21})$$

$$\mathbf{c}_k = \tilde{\mathbf{R}}_k \tilde{\mathbf{c}}_k, \quad \forall k \in [1, N] \quad (\text{r.22})$$

$$\tilde{\mathbf{R}}_1^T \tilde{\mathbf{R}}_1 = \mathbf{I}_{3 \times 3}. \quad (26)$$

Besides the trajectory error constraint (24), the constraints consist of the trajectory reconstruction equations (21)-(22), while constraint (26) imposes orthonormality conditions on the initial moving frame. The orthonormality of $\tilde{\mathbf{R}}$ is preserved throughout the entire horizon through the matrix exponential integrator in (21).

Initialization: The OCP (25)-(26) is a non-convex optimization problem and therefore multiple local minima may exist. The solution will converge towards one of the local minima depending on the initialization of the unknown variables. To obtain invariant solutions regardless of the reference frame in which the variables are represented, the variables must be initialized in an invariant way. One possibility is to calculate initial values for the invariants and moving frames using the analytical formulas (15)-(19), as was done in [33]. However, this initialization then suffers from the same three pitfalls mentioned at the beginning of this subsection.

We propose an alternative initialization approach here based on calculating an average moving frame, first introduced in [40] for screw trajectories, but adapted here to vector

trajectories in three steps. First, calculate the covariance matrix \mathbf{C}_c associated with the measured vector trajectory $\mathbf{c}_k^{\text{meas}}$:

$$\mathbf{C}_c = \frac{1}{N} \sum_{k=1}^N \mathbf{c}_k^{\text{meas}} (\mathbf{c}_k^{\text{meas}})^T. \quad (27)$$

Second, the orthogonal matrix of eigenvectors \mathbf{U}_c is calculated from \mathbf{C}_c . This matrix \mathbf{U}_c is invariant for changes in reference frame by definition and can be interpreted as a sort of average FS frame for the whole trajectory. The signs of the first two eigenvectors are chosen unambiguously by making the average value of the measurement vector $\mathbf{c}_k^{\text{meas}}$ positive along these directions. The third axis then follows from the cross product of the first two axes. All moving frames $\tilde{\mathbf{R}}_k$ are initialized with the average moving frame \mathbf{U}_c .

Third, to be consistent with the average moving frame, the moving frame invariants ω_κ and ω_τ are initialized with zero values. The object invariant c_k is initialized by projecting the vector trajectory $\mathbf{c}_k^{\text{meas}}$ onto the average moving frame and selecting the component along the first axis for all samples k .

Application to motion and force data: The calculation of vector invariants using OCP (25)-(26) can directly be applied to translational velocity trajectories $\mathbf{v}(\xi)$, rotational velocity trajectories $\boldsymbol{\omega}(\xi)$, force trajectories $\mathbf{f}(\xi)$, and moment trajectories $\mathbf{m}(\xi)$.

However, in many applications involving rigid-body motion, the measured data corresponds to position and orientation coordinates of the rigid body, instead of translational and rotational velocity vectors. In this case, one option is to use the following two-step approach: (1) estimate the rigid-body velocities from the measured position and orientation coordinates; (2) apply OCP (25)-(26) to the estimated velocity vectors. A downside of this first option is that integration of the optimized velocity vectors to reconstruct position and orientation coordinates will inevitably result in integration drift, such that there will be a deviation between the measured and reconstructed positions and orientations.

A second option is to adapt optimization problem (25)-(26) to include position or orientation measurements. Below, this is worked out assuming the measured positions are given by position vectors and the measured orientations by rotation matrices. The proposed approach can be adapted to deal with other orientation representations such as quaternions.

Including position measurements for case $\mathbf{c} = \mathbf{v}$: The reconstructed position vectors \mathbf{p}_k at each sample are introduced as additional variables in the OCP. The trajectory reconstruction cost $\Delta c_{\text{MSE}}(\mathbf{c}, \mathbf{c}^{\text{meas}})$ in (23) is now replaced by the MSE error between the reconstructed position \mathbf{p}_k and the measured position $\mathbf{p}_k^{\text{meas}}$ over $k = 1 \dots N$:

$$\Delta p_{\text{MSE}}(\mathbf{p}, \mathbf{p}^{\text{meas}}) = \frac{1}{N} \sum_{k=1}^N \|\mathbf{p}_k - \mathbf{p}_k^{\text{meas}}\|^2. \quad (28)$$

The reconstructed positions are defined by extending the integrator in (21) with a fourth row and column:

$$\begin{bmatrix} \tilde{\mathbf{R}}_{k+1} & \mathbf{p}_{k+1} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{R}}_k & \mathbf{p}_k \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \exp\left(\begin{bmatrix} [\tilde{\mathbf{i}}_k]_{\times} & \tilde{\mathbf{c}}_k \\ \mathbf{0}_{1 \times 3} & 0 \end{bmatrix} \Delta \xi\right). \quad (29)$$

The effect of these additions is that \tilde{c}_k , which in this case signifies the translational velocity in the moving frame, is used to update the position p_k .

Including orientation measurements for case $c = \omega$: The workflow is similar as for position. Variables for the reconstructed rotation matrices R_k at each sample are introduced. The trajectory reconstruction cost $\Delta c_{\text{MSE}}(c, c^{\text{meas}})$ in (23) is replaced by a first-order approximation of the mean-squared rotation angle [41] between the reconstructed rotation matrix R_k and the measured rotation matrix R_k^{meas} over $k = 1 \dots N$:

$$\Delta R_{\text{MSE}}(R, R^{\text{meas}}) = \frac{1}{N} \sum_{k=1}^N \left\| (R_k^{\text{meas}})^T R_k - I_{3 \times 3} \right\|_{F_u}^2, \quad (30)$$

where $\|\cdot\|_{F_u}$ corresponds to a Frobenius norm that is applied to the upper-triangular part of the matrix. The reconstructed rotation matrices R_k are defined by integrating c_k , which now signifies the rotational velocity, in an additional reconstruction constraint:

$$R_{k+1} = \exp([c_k]_{\times} \Delta \xi) R_k, \quad \forall k \in [1, N]. \quad (31)$$

Finally, adding the constraint $R_1^T R_1 = I_{3 \times 3}$ ensures that orthonormality is preserved throughout the trajectory.

IV. INVARIANT DESCRIPTOR FOR SCREW TRAJECTORIES

This section derives a local invariant descriptor for a screw trajectory $\mathcal{J}(\xi)$, parameterized with respect to a progress variable ξ :

$$\mathcal{J}(\xi) = \begin{pmatrix} \mathbf{a}(\xi) \\ \mathbf{b}(\xi) \end{pmatrix}, \quad \text{with } \mathbf{a}, \mathbf{b} \in \mathbb{R}^{3 \times 1}, \quad (32)$$

where \mathbf{a} is the direction part of the screw and \mathbf{b} the moment part. Again we omit the explicit dependency on ξ for conciseness while the derivative of \mathcal{J} with respect to the progress ξ is denoted as: $\mathcal{J}' = \frac{d\mathcal{J}}{d\xi}$.

A. Definition of invariants

The screw invariants are based on a generalization of *screw theory* for rigid-body kinematics and statics. In 1900, Ball [35] published a complete *theory of screws* based on the theorems of Chasles and Poinot. *Chasles' theorem* states that any rigid-body displacement can always be represented as a rotation and translation along an axis in space. *Poinot's theorem* is the dual to Chasles' theorem and it states that all forces and moments acting upon a rigid body can always be replaced by a single force and moment along an axis in space. This duality allows us to calculate the screw invariants of a wrench trajectory similar to how it was done for a twist trajectory in [25]. Hence, \mathcal{J} can represent either a twist \mathcal{t} or a wrench \mathcal{w} . In screw theory, the *screw axis* refers to the axis along which the twist or wrench can be expressed [42], but commonly it is also referred to as the *instantaneous screw axis* (ISA) to highlight its instantaneous nature.

The screw invariants are expressed in a local moving frame, referred to as the *Instantaneous Screw Axis frame* (ISA frame), which is constructed as follows. The orientation of the ISA frame is completely determined by \mathbf{a} , the direction part of the screw, i.e. ω for a twist and \mathbf{f} for a wrench. Hence, this

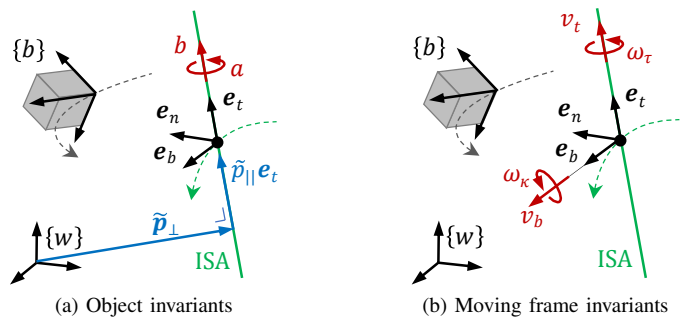


Fig. 3. Invariant descriptor for a screw trajectory: (a) object invariants a and b defined in the moving ISA frame given by the unit vectors e_t , e_n , e_b , and the origin $\tilde{p} = \tilde{p}_{\perp} + \tilde{p}_{\parallel} e_t$; (b) the moving frame invariants ω_{κ} , ω_{τ} , v_b , and v_t . The figure shows the special case where the screw \mathcal{J} corresponds to the screw twist \mathcal{t} , representing the motion of the rigid body $\{b\}$ w.r.t $\{w\}$.

orientation, represented by the rotation matrix \tilde{R} , is calculated in a similar way as the FS frame using (15) after replacing c by \mathbf{a} . Now, the tangent e_t is along the ISA, while the binormal e_b is in the direction in which the ISA is instantaneously rotating.

On the other hand, the origin of the ISA frame \tilde{p} is defined as the point on the ISA about which the ISA is instantaneously rotating. An analytic procedure to derive this origin was proposed in [25]. We summarize it here as a two-step procedure, illustrate it schematically in Fig. 3a, and refer to [25] for more detail. First, the point \tilde{p}_{\perp} on the ISA closest to the origin of the reference frame is determined:

$$\tilde{p}_{\perp} = \frac{\mathbf{a} \times \mathbf{b}}{\|\mathbf{a}\|^2}. \quad (33)$$

Second, the parallel displacement \tilde{p}_{\parallel} along the ISA (from \tilde{p}_{\perp} to \tilde{p}) is defined [25]:

$$\tilde{p}_{\parallel} = \frac{\|\mathbf{a}\|^2}{\|\mathbf{a} \times \mathbf{a}'\|} (\tilde{p}_{\perp}' \cdot e_n), \quad (34)$$

such that the *origin* of the ISA frame is found by their addition:

$$\tilde{p} = \tilde{p}_{\perp} + \tilde{p}_{\parallel} e_t. \quad (35)$$

As opposed to the FS frame, the ISA frame is a *complete* moving frame having both a position and orientation, which can be expressed using the pose matrix \tilde{T} :

$$\tilde{T} = \begin{bmatrix} \tilde{R} & \tilde{p} \\ \mathbf{0}_{3 \times 1} & 1 \end{bmatrix}. \quad (36)$$

With the ISA frame determined, the screw invariants can be defined. The first two invariants a and b are defined, respectively, as the components of \mathbf{a} and \mathbf{b} along the ISA:

$$a = \mathbf{a} \cdot e_t, \quad b = \mathbf{b} \cdot e_t. \quad (37)$$

These two invariants are referred to as *object invariants* since they relate to the object: for motion, they correspond to the rotation and translation of the object along the ISA, while for force, they correspond to the force and moment applied to the object along the ISA.

The screw trajectory is reconstructed from the object invariants using a screw transformation matrix $S(T)$ as in (7):

$$\mathcal{J} = S(\tilde{T}) \tilde{\mathcal{J}}, \quad \text{with } \tilde{\mathcal{J}} = (a \ 0 \ 0 \ b \ 0 \ 0)^T. \quad (38)$$

The first-order kinematics $\tilde{\mathbf{T}}'$ of the ISA frame $\tilde{\mathbf{T}}$ is completely determined by four invariants:

$$\tilde{\mathbf{T}}' = \tilde{\mathbf{T}} \begin{bmatrix} 0 & -\omega_\kappa & 0 & v_t \\ \omega_\kappa & 0 & -\omega_\tau & 0 \\ 0 & \omega_\tau & 0 & v_b \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (39)$$

Here, ω_κ and ω_τ correspond to the curvature rate and torsion rate of the vector \mathbf{a} , which is the direction part of the screw \mathfrak{J} . Hence, the invariants ω_κ and ω_τ are calculated using similar formulas as in (19), with \mathbf{c} replaced by \mathbf{a} . This is evident because, for all ξ , the orientation part $\tilde{\mathbf{R}}$ in (36) is calculated in exactly the same way as $\tilde{\mathbf{R}}$ in (15) for the FS frame. Hence, also its derivative is the same. The invariants v_b and v_t correspond to the translation of the origin of the ISA frame along \mathbf{e}_b and \mathbf{e}_t and are defined as [25]:

$$v_b = \tilde{\mathbf{p}}'_\perp \cdot \mathbf{e}_b, \text{ and } v_t = \tilde{\mathbf{p}}'_\perp \cdot \mathbf{e}_t + \tilde{\mathbf{p}}'_\parallel. \quad (40)$$

Equation (39) can be written compactly using (4):

$$\tilde{\mathbf{T}}' = \tilde{\mathbf{T}} [\dot{\mathbf{i}}]_\times, \quad \text{with } \dot{\mathbf{i}} = (\omega_\tau \ 0 \ \omega_\kappa \ v_t \ 0 \ v_b)^T, \quad (41)$$

where $\dot{\mathbf{i}}$ is referred to as the *moving frame invariant* since it models the first-order kinematics of the ISA frame. Fig. 3 visualizes the definition of the moving frame $\tilde{\mathbf{T}}$, the object invariants \mathbf{a} and \mathbf{b} , and the moving frame invariant $\dot{\mathbf{i}}$.

Singularities: The singularities of the screw invariants bear a strong relationship with the singularities of the vector invariants. The first type of singularity occurs when the direction vector is zero ($\mathbf{a} = \mathbf{0}$). The object invariant a is zero while the moving frame invariants ω_κ , ω_τ , v_t , and v_b are undefined. In other words, the complete moving frame is arbitrary. In this case, the moment vector \mathbf{b} , i.e., \mathbf{v} for twist and \mathbf{m} for wrench, can still be used to construct the orientation of the moving frame using the vector invariants defined in Section III. This singular case is also referred to as *infinite pitch* or *pure moment* for motion and force, respectively.

The second type of singularity occurs when the screw axis has an unchanging direction ($\mathbf{a} \times \mathbf{a}' = \mathbf{0}$) so that it remains parallel to itself. As a result, the invariant ω_κ is zero, while the other moving frame invariants ω_τ , v_t , and v_b are undefined. The perpendicular distance $\tilde{\mathbf{p}}_\perp$ is still defined. In other words, the direction of the ISA and perpendicular distance to the ISA are well-defined in this case, but the parallel distance $\tilde{\mathbf{p}}_\parallel$ and normal and binormal axes are arbitrary. In this case, the derivative of the perpendicular distance $\tilde{\mathbf{p}}'_\perp$ can be used instead of \mathbf{a} to construct the normal and binormal axes. Thus, it is possible to determine the complete orientation and perpendicular distance of the ISA frame, but not its position along the ISA since $\tilde{\mathbf{p}}_\parallel$ is undefined.

Invariant properties: All the invariant properties of the vector invariants in Section III also hold for the screw invariants. The key advantage of the screw invariants compared to vector invariants is their additional invariance to the choice of the location of the reference point on the object. In other words, screw invariants are invariant to the choices of *both* the world frame $\{w\}$ and body frame $\{b\}$ in which the screw coordinates \mathfrak{J} may be expressed. This property is also referred to as *bi-invariance* [36].

Trajectory reconstruction: Reconstructing the trajectory $\mathfrak{J}(\xi)$ from the invariant descriptor $(\tilde{\mathfrak{J}}(\xi), \dot{\mathbf{i}}(\xi))$ requires numerical integration of the differential equation (41). Discretizing the problem again, the moving frame $\tilde{\mathbf{T}}_{k+1}$ at sample $k+1$ can be found from $\tilde{\mathbf{T}}_k$ at sample k as follows:

$$\tilde{\mathbf{T}}_{k+1} = \tilde{\mathbf{T}}_k \exp([\dot{\mathbf{i}}_k]_\times \Delta\xi), \quad (42)$$

where we assume that $\dot{\mathbf{i}}_k$ remains constant over the integration step $\Delta\xi$. The matrix exponential operator $\exp(\cdot)$ maps the matrix $[\dot{\mathbf{i}}]_\times \Delta\xi$ which is part of the Lie algebra $se(3)$ into the corresponding change in pose which is part of the Lie group $SE(3)$, and has a closed-form expression under the given assumptions [36]. Given the initial value for $\tilde{\mathbf{T}}_1$ at $k=1$, the moving frame $\tilde{\mathbf{T}}_k$ can be reconstructed for all remaining samples $k=2\dots N$. From the moving frames, the corresponding screw trajectory \mathfrak{J}_k is retrieved using (38):

$$\mathfrak{J}_k = \mathbf{S}(\tilde{\mathbf{T}}_k) \tilde{\mathfrak{J}}_k, \quad \forall k \in [1, N]. \quad (43)$$

Table III (bottom half) provides an overview of the screw invariants framework with applications to rigid-body twist \mathfrak{t} and wrench \mathfrak{w} trajectories.

B. Robust calculation of invariants using optimal control

Calculating screw invariants $\tilde{\mathfrak{J}}$ and $\dot{\mathbf{i}}$ with the analytical formulas has similar problems related to singularities, sensitivity to noise, and integration drift during reconstruction, as mentioned in Section III-B for vector invariants.

To address these problems, we again formulate an OCP, of which the objective is to find the screw invariants that reconstruct the given measured screw trajectory in a robust way. This is reflected by introducing a cost $\Delta\mathfrak{J}_k$ that contains the difference between the reconstructed \mathfrak{J}_k and measured screw trajectory \mathfrak{J}_k^{meas} at sample k :

$$\Delta\mathfrak{J}_k = \mathfrak{J}_k - \mathfrak{J}_k^{meas} = \begin{pmatrix} \Delta\mathbf{a}_k \\ \Delta\mathbf{b}_k \end{pmatrix}. \quad (44)$$

Similarly as in Section III-B, a mean-squared trajectory reconstruction error is formulated, but the errors on the directional part of the screw $\Delta\mathbf{a}_{\text{MSE}}$ and the moment part of the screw $\Delta\mathbf{b}_{\text{MSE}}$ are considered separately:

$$\epsilon_a^2 \geq \Delta\mathbf{a}_{\text{MSE}}(\mathfrak{J}, \mathfrak{J}^{meas}) = \frac{1}{N} \sum_{k=1}^N \|\Delta\mathbf{a}_k\|^2, \quad (45)$$

$$\epsilon_b^2 \geq \Delta\mathbf{b}_{\text{MSE}}(\mathfrak{J}, \mathfrak{J}^{meas}) = \frac{1}{N} \sum_{k=1}^N \|\Delta\mathbf{b}_k\|^2, \quad (46)$$

bounded by tolerance ϵ_a for the direction part of the screw and tolerance ϵ_b for the moment part of the screw.

A regularization cost is introduced for the moving frame invariants to achieve a more stable and smooth evolution of the moving frame $\tilde{\mathbf{T}}$ in the presence of measurement noise and singularities:

$$\sum_{k=1}^{N-1} \|\dot{\mathbf{i}}_k\|_L^2 = \sum_{k=1}^{N-1} \omega_\kappa^2[k] + \omega_\tau^2[k] + \frac{1}{L^2} (v_n^2[k] + v_b^2[k]), \quad (47)$$

where L signifies a chosen length scale with units [m] to properly compare rotational and translational invariants.

TABLE III

OVERVIEW OF EQUATIONS FOR VECTOR INVARIANTS (TOP HALF) AND SCREW INVARIANTS (BOTTOM HALF). THE FIRST COLUMN SHOWS THE TYPE OF VECTOR AND SCREW TRAJECTORIES FROM WHICH INVARIANTS CAN BE CALCULATED. IN THE SECOND COLUMN, THE LOCAL MOVING FRAME IS DEFINED IN WHICH THE INVARIANTS ARE EXPRESSED. THE THIRD COLUMN DEFINES THE MOVING FRAME INVARIANT THAT CHARACTERIZES THE MOTION OF THE MOVING FRAME, AND THE OBJECT INVARIANT THAT CHARACTERIZES THE OBJECT'S TRAJECTORY IN THE MOVING FRAME. THE FOURTH COLUMN DETAILS HOW TO RECONSTRUCT THE TRAJECTORY FROM THE INVARIANT DESCRIPTOR, POSSIBLY INCLUDING THE OBJECT'S POSITION AND ROTATION.

Trajectory coordinates	Moving frame definition	Invariants definition	Reconstruction
Vectors c: $c = \omega$ for rot. velocity $c = v$ for transl. velocity $c = f$ for force $c = m$ for moment derivatives c' and c''	orientation frame $\tilde{R} = [e_t \ e_n \ e_b]$, $e_t = \frac{c}{\ c\ }$ $e_n = \frac{(c \times c') \times c}{\ (c \times c') \times c\ }$ $e_b = e_t \times e_n$	moving frame invariant $i = (\omega_\tau \ 0 \ \omega_\kappa)^T$: $\omega_\kappa = \frac{\ c \times c'\ }{\ c\ ^2}$ $\omega_\tau = \frac{((c \times c') \times (c \times c'')) \cdot c}{\ c \times c'\ ^2 \ c\ }$ object invariant $\tilde{c} = (c \ 0 \ 0)^T$ $c = c \cdot e_t$	moving frame equation $\tilde{R}' = \tilde{R} [i]_\times$ object trajectory equation $c = \tilde{R} \tilde{c}$ object orientation R equation: $R' = [c]_\times R$, when $c = \omega$ object position p equation: $p' = c$, when $c = v$
Screws $s = \begin{pmatrix} a \\ b \end{pmatrix}$: $s = \begin{pmatrix} \omega \\ v \end{pmatrix}$ for twist $s = \begin{pmatrix} f \\ m \end{pmatrix}$ for wrench derivatives s' and s''	pose frame $\tilde{T} = \begin{bmatrix} e_t & e_n & e_b & \tilde{p} \\ 0 & 0 & 0 & 1 \end{bmatrix}$: $e_t = \frac{a}{\ a\ }$ $e_n = \frac{(a \times a') \times a}{\ (a \times a') \times a\ }$ $e_b = e_t \times e_n$ $\tilde{p} = \tilde{p}_\perp + \tilde{p}_\parallel e_t$ $\tilde{p}_\perp = \frac{a \times b}{\ a\ ^2}$ $\tilde{p}_\parallel = \frac{\ a\ ^2}{\ a \times a'\ } (\tilde{p}' \cdot e_n)$	moving frame invariant $\tilde{i} = (\omega_\tau \ 0 \ \omega_\kappa \ v_t \ 0 \ v_b)^T$: $\omega_\kappa = \frac{\ a \times a'\ }{\ a\ ^2}$ $\omega_\tau = \frac{((a \times a') \times (a \times a'')) \cdot a}{\ a \times a'\ ^2 \ a\ }$ $v_b = \tilde{p}'_\perp \cdot e_b$ $v_t = \tilde{p}'_\perp \cdot e_t + \tilde{p}'_\parallel$ object invariant $\tilde{s} = (a \ 0 \ 0 \ b \ 0 \ 0)^T$: $a = a \cdot e_t$ $b = b \cdot e_t$	moving frame equation $\tilde{T}' = \tilde{T} [\tilde{i}]_\times$ object trajectory equation $s = S(\tilde{T}) \tilde{s}$ object pose T equation: $T' = [s]_\times T$, when $s = \begin{pmatrix} \omega \\ v \end{pmatrix}$

The complete OCP to calculate the screw invariants $\tilde{\mathcal{J}}$ and $\tilde{\mathcal{I}}$ from a given sequence of N measured screws \mathcal{J}_k^{meas} is:

$$\underset{\tilde{\mathcal{J}}_{[1]}, \tilde{\mathcal{I}}_{[1]}, \mathcal{J}_{[1]}, \tilde{T}_{[1]}}{\text{minimize}} \sum_{k=1}^{N-1} \|\tilde{\mathcal{I}}_k\|_L^2, \quad (48)$$

subject to:

$$\epsilon_a^2 \geq \Delta a_{\text{MSE}}(\mathcal{J}, \mathcal{J}^{meas}), \quad (r.45)$$

$$\epsilon_b^2 \geq \Delta b_{\text{MSE}}(\mathcal{J}, \mathcal{J}^{meas}), \quad (r.46)$$

$$\tilde{T}_{k+1} = \tilde{T}_k \exp([\tilde{\mathcal{I}}_k]_\times \Delta \xi), \quad \forall k \in [1, N] \quad (r.42)$$

$$\mathcal{J}_k = S(\tilde{T}_k) \tilde{\mathcal{J}}_k, \quad \forall k \in [1, N] \quad (r.43)$$

$$\tilde{T}_1^{-1} \tilde{T}_1 = I_{4 \times 4}. \quad (49)$$

Besides the trajectory error constraints (45)-(46), the constraints consist of the trajectory reconstruction equations (42)-(43), supplemented with an orthonormality constraint on the initial moving frame \tilde{T}_1 .

Initialization: Similarly as for vector invariants, an invariant initialization of the variables in the OCP can be done using either the analytical formulas or an approach based on an average moving frame [40]. The latter is summarized as follows. The orientation of the moving frames \tilde{T}_k is initialized using the same approach as for vector trajectories, based on the directional part of the measured screw trajectory \mathcal{J}_k^{meas} . The origin of the moving frames \tilde{T}_k is initialized using the average intersection point of all the instantaneous screw axes corresponding to the measured screw trajectory \mathcal{J}^{meas} [40].

The moving frame invariants ω_κ , ω_τ , v_b , and v_t are initialized with zero values. The object invariants a_k and b_k are initialized by transforming the screw trajectory \mathcal{J}_k^{meas} to the average moving frame and selecting the components along the first axis for each sample k .

Application to motion and force data: The calculation of screw invariants using OCP (48)-(49) can directly be applied to twist trajectories $\begin{pmatrix} \omega \\ v \end{pmatrix}(\xi)$ and wrench trajectories $\begin{pmatrix} f \\ m \end{pmatrix}(\xi)$.

Similarly as in Section III-B, if the measured motion data consists of positions and orientations of the rigid body, instead of twists, the optimization problem (48)-(49) can be adapted to include the object's pose. Below, this is worked out assuming the positions and orientations are given by homogeneous transformation matrices. The proposed approach can be adapted to other pose representations, such as dual quaternions.

Including pose measurements for case $s = \begin{pmatrix} \omega \\ v \end{pmatrix}$: Additional variables are introduced for the reconstructed pose matrices $T_k = \begin{bmatrix} R_k & p_k \\ 0_{1 \times 3} & 1 \end{bmatrix}$ at each sample k . The trajectory reconstruction costs Δa_{MSE} and Δb_{MSE} in (45)-(46) are replaced by the MSE errors between the reconstructed and the measured rotation matrices and positions:

$$\Delta R_{\text{MSE}}(R, R^{meas}) = \frac{1}{N} \sum_{k=1}^N \left\| (R_k^{meas})^T R_k - I_{3 \times 3} \right\|_{F_u}^2, \quad (50)$$

$$\Delta p_{\text{MSE}}(p, p^{meas}) = \frac{1}{N} \sum_{k=1}^N \|p_k - p_k^{meas}\|^2. \quad (51)$$

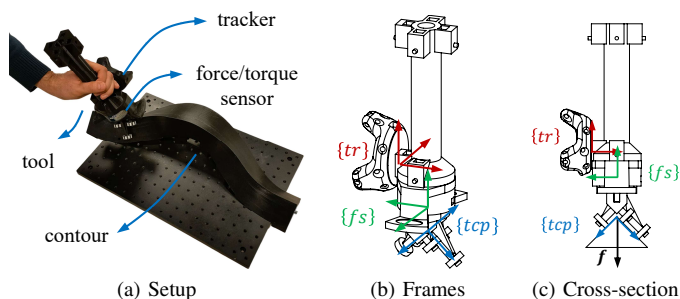


Fig. 4. The 3D contour following setup: (a) demonstration setup showing tool and contour, (b) 3D drawing showing the three assigned frames on the tool: tracker $\{tr\}$, force/torque sensor $\{fs\}$, and tool center point (TCP) $\{tcp\}$, (c) cross-section of the contour showing the reference contact force f .

The reconstructed pose matrices T_k are defined by integrating \mathcal{J}_k , in this case signifying the object's twist, in additional reconstruction constraints:

$$T_{k+1} = \exp([\mathcal{J}_k]_{\times} \Delta\xi) T_k, \quad \forall k \in [0, N]. \quad (52)$$

Finally, adding an orthonormality constraint on the initial pose T_1 , like in (49), ensures that orthonormality is preserved.

V. APPLICATION TO 3D CONTOUR FOLLOWING

This section demonstrates the use and benefits of screw and vector invariant descriptors for both motion and force trajectories by analyzing a human-demonstrated 3D contour following task, shown in Fig. 4. The human operator holds the tool while following the contour and pressing the tool's five contact wheels (diameter 22 mm) against the contour.

The contour consists of the edge between two contact surfaces that form an angle of 90° in each point along the contour (see Fig. 4c). Each of the five contact wheels takes away one degree of freedom for the motion of the tool relative to the contour, while creating one degree of freedom to apply a contact force. Hence, every point along the contour is characterized by a 1-dof vector space of possible twists and a 5-dof vector space of possible contact wrenches. This makes the application interesting, as we expect to obtain a very good repeatability of the invariant descriptors for motion among different demonstrations, while we expect to see the effect of human variation among the demonstrations in the invariant descriptors for force. Furthermore, the contour is designed to have a symmetric curvature profile and an antisymmetric torsion profile, both with respect to its midpoint. The contour starts and ends with a straight line segment (no curvature or torsion).

A. Experimental set-up

An HTC VIVE motion capture system, consisting of a tracker mounted on the tool and a camera (not shown in Fig. 4), records the pose trajectory of the tool with respect to a reference frame attached to the camera. This frame is chosen as the world frame $\{w\}$. The system records the pose trajectory with a frequency of 200 Hz and an accuracy in the order of a few millimeters and a few degrees. The contact

force/torque is measured with the same frequency using a 6-axis JR3 force/torque sensor. The nominal accuracy of this force/torque sensor is $\pm 1\%$ of its standard measurement range which is ± 800 N along the central axis (Z-axis) and ± 400 N in the horizontal plane (XY-plane) for the force, and ± 24 Nm in all directions for the moment.

Three frames are rigidly attached to the tool: the motion tracker frame $\{tr\}$, the force/torque sensor frame $\{fs\}$, and the tool-center-point (TCP) frame $\{tcp\}$. The $\{tcp\}$ frame is defined such that, when the tool is tracking a straight-line segment, the origin of $\{tcp\}$ corresponds to the intersection of the contour's edge and the symmetry axis of the cylindrical part of the tool (see Fig. 4c). One axis of the $\{tcp\}$ frame is along the edge and the remaining two axes are along the two surfaces. (For completeness: if the edge has curvature and/or torsion, the $\{tcp\}$ frame will deviate slightly from this contour-centered definition due to the complex contact geometry).

To introduce variation in the measurements, the tracker can be physically attached to the tool at different locations, further discussed in Section V-C. However, the location of the force sensor relative to the tool cannot be changed in the current set-up. Instead, and without loss of generality, we can artificially transform the physically measured wrenches to a different, virtual frame on the tool using a screw transformation matrix S as in (9). We can take this one step further and in a similar manner transform the physically measured wrenches to the world frame $\{w\}$. This way we can, again without loss of generality, simulate an alternative experimental set-up in which the force sensor is fixed to the world instead of the tool, i.e., representing a situation where the force sensor is placed underneath the support to which the contour is fixed.

B. Objectives

A first objective is to show that the invariant descriptors have a physical interpretation and to obtain evidence that they are calculated correctly using the OCPs. It is however very difficult to calculate the ground truth invariants given the complex contact geometry involving five contact wheels with finite radii and finite mutual distances. Instead, we calculated reference values to compare the invariants with, for the limit case of an infinitely small set of contact wheels and a contact wrench which corresponds to a pure force with constant magnitude $\|f\| = 25$ N, as shown in Figure 4c. In this limit case, the pose of $\{tcp\}$ is easily derived from the contour's CAD data. The wrench is then also uniquely defined and constant in any frame attached to the tool. From the reference pose and reference wrench we calculate the corresponding invariant descriptors using the same OCPs as for the human-demonstrated data.

A second objective is to confirm the invariant properties of the vector invariants and the screw invariants, as discussed in Sections III and IV, respectively. Furthermore, we want to point out the practical benefits of these properties in terms of relaxed calibration needs. We also want to show the difference between the invariant descriptors depending on the perspective of the input data: as the motion and wrench trajectories can be defined either with respect to the world or with respect to the

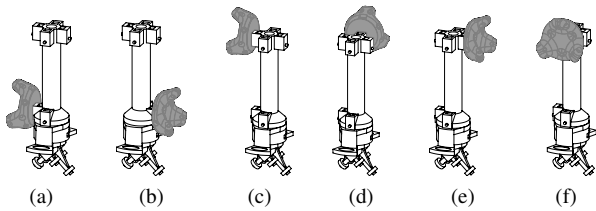


Fig. 5. The six different configurations of the motion tracker on the tool.

moving body, the corresponding moving frame invariants will also model the motion of the moving frame with respect to the world or the moving body, and hence they will be different.

A third objective is to confirm that the new formulation of the OCPs for calculating the invariants, as introduced in Sections III-B and IV-B, allows for an intuitive and effective parameter tuning.

A fourth and final objective is to confirm that the trajectories reconstructed from the calculated invariants are accurate and driftless, even when the trajectories are reconstructed in a different region in space.

C. Experiment design

Twelve demonstrations of the 3D contour following task were recorded. According to the experimental protocol, the operator was asked to repeat the task twelve times by starting from approximately the same location (at the end of the initial straight-line segment) and approximately ending at the same location (at the beginning of the final straight-line segment). The operator was also asked to try to maintain a constant pure contact force as shown in Fig. 4c, because this can be shown to be a very good strategy to ensure that the five wheels remain in contact with the contour. Both instructions contributed to limiting human variation in the task execution.

Each of the twelve demonstrations was recorded with the tracker physically attached to the tool in one of six different locations, shown in Fig. 5 (i.e. two demonstrations per tracker location), resulting in significantly different pose trajectories. Additionally, the physically measured contact wrenches were artificially transformed from the force sensor frame $\{fs\}$ to the tracker frame $\{tr\}$ used in the demonstration, simulating a change in location of the force sensor and resulting in significantly different wrench trajectories.

The screw invariants are expected to be invariant for the previous variations, but the vector invariants are not. To verify the repeatability of the vector invariants, it is necessary to pick a well-chosen reference point on the tool to express the position and moment trajectories, the same reference point for all demonstrations. The obvious choice was the origin of $\{tcp\}$, both for the position and moment trajectories. Hence, for this specific case, the measured pose and wrench trajectories of the twelve demonstrations were additionally transformed to the $\{tcp\}$ frame. Note that this requires calibration of the origins of $\{tr\}$ and $\{fs\}$ with respect to $\{tcp\}$, but this is required solely for the vector invariants, not the screw invariants.

The twelve resulting pose trajectories were further modified by changing the pose of the world reference frame relative to

TABLE IV

EACH OF THE TWELVE DEMONSTRATION TRIALS IS LISTED WITH ONE OF THE SIX CONFIGURATIONS OF THE MOTION TRACKER (SEE FIG. 5) AND WITH THE ARTIFICIAL TRANSFORMATION THAT WAS APPLIED TO THE MEASURED POSES TO SIMULATE A LARGE VARIATION IN THE WORLD (I.E. CAMERA) FRAME. THESE TRANSFORMATIONS ARE SPECIFIED USING XYZ EULER ANGLES AND A TRANSLATION VECTOR.

Trial # +config.	Artificial transformation		Trial #	Artificial transformation	
	angles [°]	transl. [m]	+config.	angles [°]	transl. [m]
1 (a)	[0, -60, 0]	[-3, 2, -1]	7 (d)	[-90, 0, 45]	[1, 0.5, -0.5]
2 (a)	[30, 0, -180]	[-1, 2.5, -1]	8 (d)	[0, 90, -135]	[-1, 0, -0.5]
3 (b)	[0, -60, -180]	[-1, 2, -2]	9 (e)	[60, 0, -150]	[1, -2, 2]
4 (b)	[45, 0, 0]	[-3, 2.5, -2]	10 (e)	[0, 45, -60]	[3, -2, 2]
5 (c)	[-90, 45, 90]	[-1, 0.5, 0.5]	11 (f)	[0, -45, -180]	[3, -1.5, 1]
6 (c)	[0, 90, 45]	[1, 0, 0.5]	12 (f)	[-60, 0, -45]	[1, -1.5, 1]

the contour, simulating a change in camera viewpoint. Table IV lists the different artificial transformations that were applied to each of the twelve trials together with their corresponding tracker configuration. Changing the world reference frame is not expected to affect the screw or vector invariants for motion. Obviously, such change also does not affect the wrench trajectories and the resulting wrench invariants since the force sensor is attached to the tool, not to the world.

To simulate the alternative experimental set-up in which the force sensor is fixed to the world, the twelve measured wrench trajectories were artificially transformed to a single frame, fixed to the world underneath the contour, about halfway the contour. The exact location of this frame relative to the contour is not known or calibrated, but this does not affect the screw invariants.

Finally, to show that the trajectories reconstructed from the invariants are accurate and driftless, the motion and force trajectories were reconstructed at a new location and were compared with the measured trajectories, after they were globally transformed to the same location.

D. Data processing

This section explains how the screw and vector invariants are calculated from the measured pose and wrench trajectories and the reference pose and wrench trajectories.

Preprocessing: There are two preprocessing steps: segmentation and re-parameterization.

The purpose of segmentation is to extract the motion-in-contact part from a complete demonstration, which also includes an approach-to-contact motion, a retract motion, and two stationary parts while in contact, at the start and the end of the actual contour tracking motion. Segmentation is achieved by using a threshold for the magnitude of the force (5 N) to detect contact, and a threshold for the magnitude of the translational velocity of the TCP (0.05 m/s) to remove the stationary parts of the demonstration.

Re-parameterization involves changing the time-based input trajectories to trajectories that are function of an alternative progress variable. As discussed in Section II-C, for tasks with dominant translation like contour following, the integral of the translational velocity is a natural choice. Accordingly, we chose the progress variable ξ as the integral of the TCP's velocity, because it best reflects the arc length along the contour.

Finally, we rescaled the progress variable to a dimensionless variable ranging from 0 to 1, making it independent of the scale of the contour.

Calculation of the invariants: The force invariants were calculated from the wrench measurements using the original OCPs: (25)-(26) for vector invariants and (48)-(49) for screw invariants. The motion invariants were calculated from the pose measurements using the adapted versions of the OCPs using (28)-(31) for the vector invariants and (50)-(52) for the screw invariants. All OCPs were specified using the nonlinear optimization framework CasADi [43] and solved using a primal-dual Newton method with IPOPT [44].

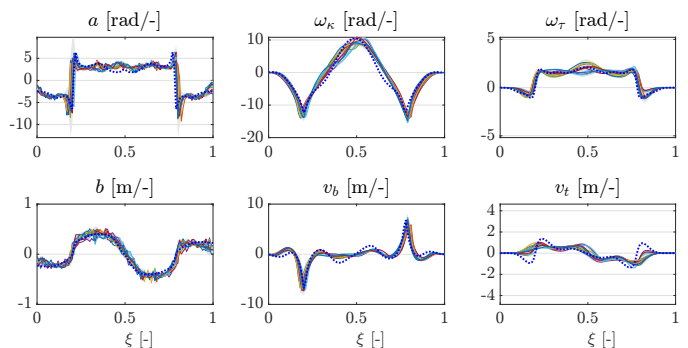
Initialization of the OCP: For initializing the moving frames, we followed the initialization approaches based on an average moving frame as explained in Sections III-B and IV-B with a minor variation. Due to the symmetry and antisymmetry in the contour, averages calculated over the trajectory might become approximately zero. To avoid confusion in determining the signs, we calculated the average over the last two-thirds of the trajectory. The motion invariants are calculated using pose measurements as input. However, the initialization routine uses velocity and twist data as input. Hence, for motion, numerical differentiation is performed to estimate the velocities and twists as inputs for the initialization routine.

Tuning of the OCPs: Vector and screw invariants for motion and force were calculated using measured position, orientation, force, and moment trajectories as inputs. For every type of trajectory, a tolerance ϵ related to the desired accuracy of the reconstructed trajectory had to be chosen, such as for example in (24) for the vector trajectories. To maintain an intuitive relation between the values for these tolerances and the accuracy of the measurement systems, we expressed these tolerances in the respective sensor frames. All reported results were obtained using the following intuitive values: $\epsilon_p = 2$ mm for position and $\epsilon_R = 2^\circ$ for orientation, which corresponded to a rough estimation of the accuracy of the motion capture system, and $\epsilon_f = 0.8$ N for force and $\epsilon_m = 0.16$ Nm for moment, which corresponded to the estimated noise levels of the force/torque measurements. To calculate screw invariants, a length scale L must be chosen to properly compare rotational and translational invariants. For the application we chose $L = 0.5$ m, i.e. the length of the contour's edge.

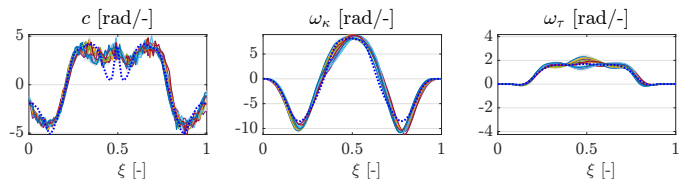
Reconstructing trajectories from invariants: The pose trajectory $\mathbf{T}(\xi)$ of the tool and the contact wrench trajectory $\mathbf{w}(\xi)$ were reconstructed from the screw invariant descriptors at a new location in space. This was done by supplying new initial moving frames $\mathbf{T}(0)$ for both motion and wrench, and a new initial pose of the tool $\mathbf{T}(0)$ in the trajectory reconstruction equations (42), (43) and (52). The frames were specified by globally transforming the original calculated initial frames to the new location. Time-based trajectories $\mathbf{T}(t)$ and $\mathbf{w}(t)$ were obtained by re-applying the time profile $\xi(t)$ that was extracted from the demonstration using the inverse of the reparameterization in (11).

E. Results and discussion

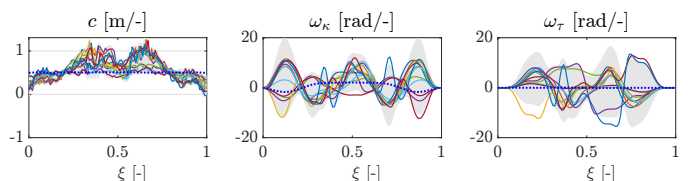
Figures 6-11 summarize the experimental results. Figures 6-8 contain screw and vector invariants: motion invariants with



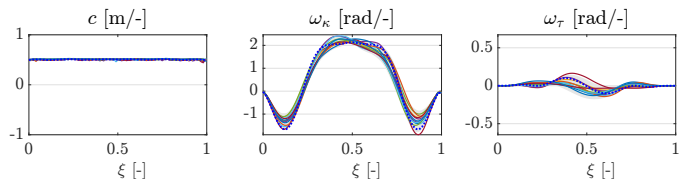
(a) Screw invariants calculated from measured pose of tracker



(b) Vector invariants calculated from measured orientation of tracker



(c) Vector invariants calculated from measured position of tracker



(d) Vector invariants calculated from estimated position of tool point

Fig. 6. Screw and vector invariants calculated from position and orientation of tracker or tool point on manipulated tool, measured in the world frame.

respect to the world (Fig. 6), force invariants with respect to the tool (Fig. 7), and force invariants with respect to the world (Fig. 8). Additional results for other cases can be generated in the provided software [34]. Each plot contains twelve thin lines corresponding to each of the twelve demonstrations. The thick dotted blue line represents the reference values for the limit case. The variation among the demonstrations is given by the two-sigma band depicted in gray. For all invariants, the dimensionless arc length ξ defined in Section V-D is chosen as the progress variable. Figure 10 visualizes the corresponding evolution of the different moving frames for one demonstration. Note that in the subfigures of Fig. 10, the contour is only positioned approximately³ because its true pose was not recorded. Figure 11 visualizes motion and force trajectories reconstructed from the invariants at a new location. Below, the results are discussed in detail.

Motion invariants with respect to world: Fig. 6a shows the screw invariants for motion. The object invariants a and

³This approximation is based on the spatial alignment of the recorded trajectory of $\{tcp\}$ with the edge of the contour [45].

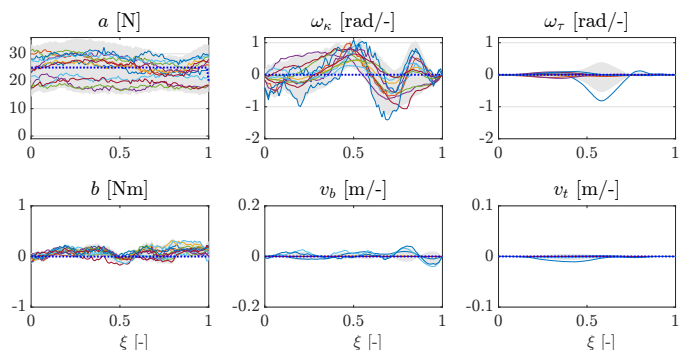
b describe the rotation and translation of the object along the screw axis, while the moving frame invariants ω_κ , ω_τ , v_b , and v_t describe the rotation and translation of the moving frame attached to the screw axis. Recalling that each demonstration has different tracker and world frames, the relatively small variations between demonstrations confirm the invariance properties of the screw invariants. Even in the presence of human variations, relatively small variations between the demonstrations and good correspondence with the reference values from the limit case are obtained. This is explained by the constrained motion, allowing variations in only one dof. Due to the special design of the contour, clear symmetric or antisymmetric profiles with respect to the midpoint were expected and were obtained. The motion of the moving frames for one trial is visualized in Figure 10a.

Fig. 6b shows the vector invariants for orientation. According to the analytical formulas, they should correspond to the invariants for the directional component of the screw for motion (top row in Fig. 6a). The reason why they are not exactly the same is explained by the OCP-based calculation approach. In this approach, the calculation for orientation and translation are coupled into a single OCP for the screw invariants and hence they are subject to a joint regularization. Nevertheless, the overall evolution of the vector invariants for orientation and the invariants for the directional component of the screw are still similar.

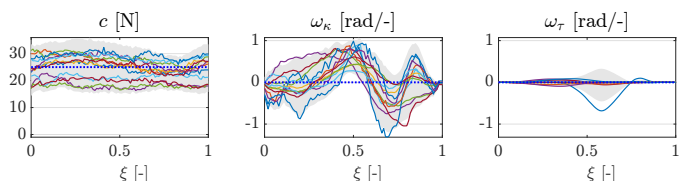
Turning to the vector invariants for translation, Fig. 6c confirms that they are not invariant for changes of the tracker location, serving as the reference point to define the translation of the tool. This non-invariance can be resolved by picking the same reference point for all demonstrations, in this case the origin of $\{tcp\}$, as shown in Fig. 6d. In this figure, c represents the magnitude of the translational velocity vector, while ω_κ and ω_τ represent the dimensionless curvature and torsion of the point curve tracked by the origin of $\{tcp\}$. From all the figures, Fig. 6d exhibits the best correspondence between the reference limit case and the demonstrations. This is to be expected since the limit case models the translation of the $\{tcp\}$ -frame along the edge. For this limit case, c is approximately constant (about 0.5 m/-). Its integral (also close to 0.5 m) represents the arc length of the point curve tracked by the origin of $\{tcp\}$. The small values for the torsion rate indicate that this point curve lies, at least locally, approximately in a plane. The small differences between the limit case and the demonstrations are due to the finite dimensions of the tracking tool and due to measurement errors. The moving FS frames corresponding to these vector invariants for one demonstration are visualized in Fig. 10d.

Given the geometry-based definition of the invariants, they can be used to segment the trajectory into meaningful parts. For example, in the screw invariants in Fig. 6a, we notice two inflection points at $\xi \approx 0.2$ and at $\xi \approx 0.8$: while the rotational velocity a changes sign at these progress values, the peaks in v_b show that the position of the ISA is rapidly shifting from a center of curvature on one side of the contour to the other side of the contour. This is confirmed by the visualization of the corresponding ISAs in Fig. 10a.

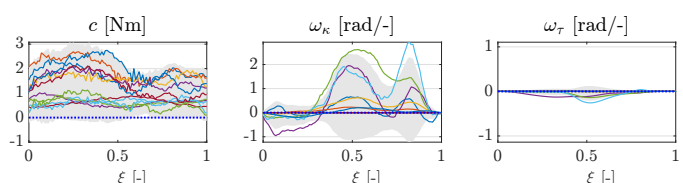
Force invariants with respect to tool: More variation



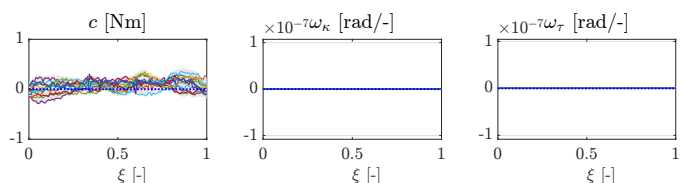
(a) Screw invariants calculated from wrench transformed to tracker frame



(b) Vector invariants calculated from force transformed to tracker frame



(c) Vector invariants calculated from moment transformed to tracker frame



(d) Vector invariants calculated from moment transformed to tool center frame

Fig. 7. Screw and vector invariants calculated from measured force and moment trajectories, transformed to either the tracker or the tool center frame.

among demonstrations was expected in the wrench invariants because, although the operator was instructed to apply a pure force in a specified direction and through a specific point of the tool, the operator could at any time apply a wrench in a local 5-dof space. The higher noise levels associated with force measurements may cause additional variation compared to the motion invariants. Note that a pure force that remains parallel to itself corresponds to the second type of singularity, mentioned in Section IV. This singularity applies to both vector and screw invariants.

Fig. 7a shows the screw invariants. Recalling that each demonstration has a different transformed force sensor frame, these plots confirm that the screw invariants are invariant for such transformation. The plot of invariant a shows that the operator was able to maintain relatively constant magnitudes of the force along the contour, although the magnitudes varied between demonstrations. The plots of ω_κ and ω_τ reveal a limited change of direction of the force (hence of the ISA), almost in a single plane (ω_τ is very small). This plane is orthogonal to the edge of the contour. This can be found

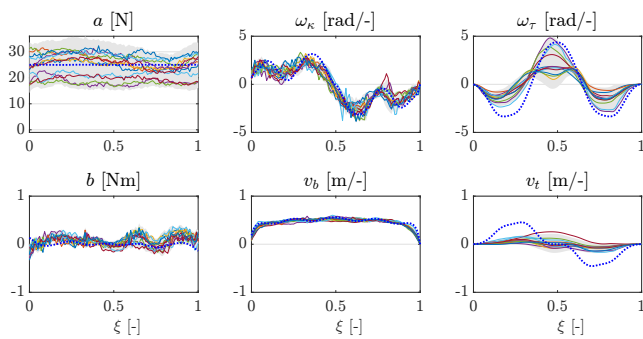


Fig. 8. Screw invariants calculated from measured wrench transformed to the world frame.

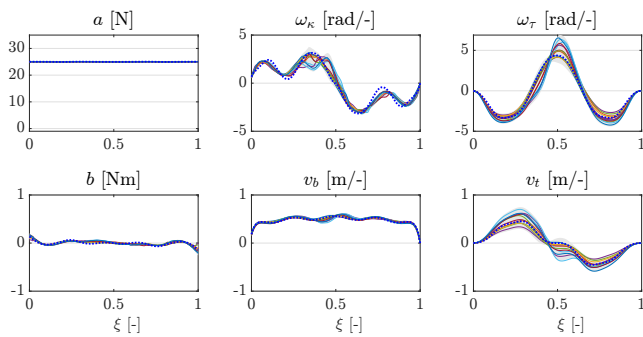


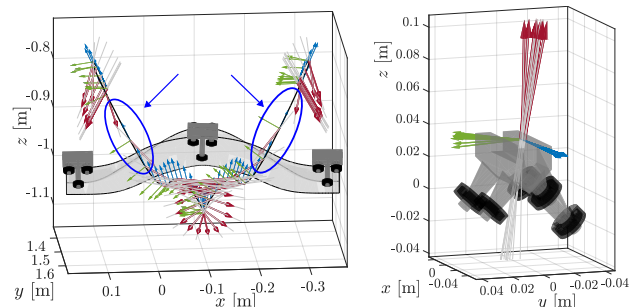
Fig. 9. Screw invariants calculated from a *simulated wrench* in the world frame to study the reduction in variation with respect to Figure 8.

by analyzing the direction of the ISAs and is confirmed in Fig. 10b, where the first (red) and second axes (green) of the successive moving frames lie approximately in a plane that is orthogonal to the tangent of the contour.

The second row of Fig. 7a shows that there is no significant moment about the ISA (b is very small) and that the origin of the moving frame does not translate much, both parallel to and along the ISA (both v_b and v_t remain very small). The latter can also be seen in Fig. 10b.

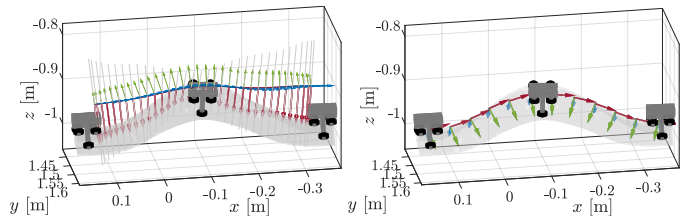
Fig. 7b shows the vector invariants for force, which are nearly identical to the directional component of the screw invariants (top row in Fig. 7a). As for the vector invariants for moment, Fig. 7c confirms that they are not invariant for a change in position of the force sensor frame, resulting in moment magnitudes up to 2.5 Nm. Similarly as for translational velocity, this can be resolved by picking the TCP as the common reference point for expressing the moments in all demonstrations, as shown in Fig. 7d. Since these moment values are relatively close to zero w.r.t. the specified tolerance $\epsilon_m = 0.16$ Nm, they can be perfectly described by a stationary Frenet-Serret frame. Therefore, the solutions of ω_κ and ω_τ are below 10^{-8} , the stopping criterion in the OCP solver.

Force invariants with respect to world: Figure 8 shows the screw invariants for the simulated, alternative experimental set-up in which the force sensor is assumed fixed to the world. This simulation, in particular the transformation of the physical force measurements to the world, is subject to two additional errors: the calibration between the force sensor and the motion tracker, and the camera measurements. Again, the invariants are shown to be insensitive to this transformation.



(a) Motion ISA frames in world

(b) Force ISA frames in tool



(c) Force ISA frames in world

(d) Transl. velocity FS frames in world

Fig. 10. Moving frames calculated from data of Trial 5: (a) ISA frames calculated from the tool's motion in the world. The red arrows correspond to the ISA itself, which is the first axis of the moving frame. The green and blue arrows correspond to the second and third axes, respectively. The two blue circles indicate the translation of the ISA frames towards the other side of the contour at the inflection points; (b) ISA frames calculated from interaction wrench, relative to the tool; (c) ISA frames calculated from wrench w.r.t the world; (d) Frenet-Serret (FS) frames calculated from the position of the tool center point (TCP). The red arrows correspond to the tangent along the contour, which is the first axis of the moving frame.

Fig. 10c shows the evolution of the corresponding moving frame. During execution, the force ISA remains approximately perpendicular to the contour edge while it approximately intersects the contour edge. To do this for the entire trajectory, the force axis has to translate along the contour with a velocity v_b , which is approximately equal to the translation of the TCP per dimensionless arc length (about 0.5 m/-). Integrating this value over the total horizon again results in the length of the entire contour (~ 0.5 m).

To further study the variation in the force invariants with respect to the world, we replaced the real wrench measurements in each demonstration with a constant simulated wrench $\mathbf{w} = 25 [0 \cos(\frac{\pi}{4}) \sin(\frac{\pi}{4}) 0 0 0]^T$ expressed in the TCP frame. By using this simulated wrench, the effects of human variations and wrench measurement noise were eliminated. Consequently, the corresponding force invariants with respect to the world are only influenced by errors in the calibration between the TCP and the motion tracker, and noise in the motion measurements. Invariant descriptors of this case are shown in Fig. 9. We observe greater repeatability in force invariants compared to Fig. 8 as a result of eliminating the human variation and wrench measurement noise. This analysis confirms that the larger variation in the force invariants compared to motion invariants is mainly due to human variation in a 5-DOF space, not due to errors in the computation.

Reconstructed motion and force trajectories: Figure 11 visualizes a reconstruction of the motion and force trajectory at a new location starting from the invariants of one of the

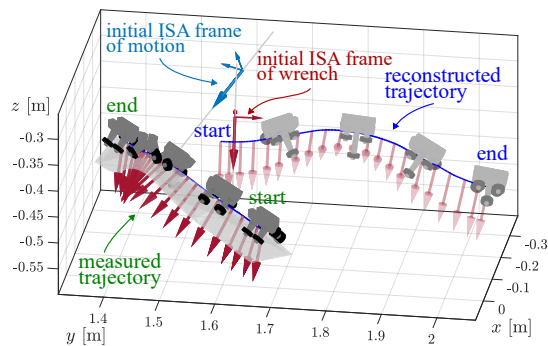


Fig. 11. Reconstruction of the motion and wrench trajectory of Trial 7 at a new location given the screw invariants of the measured trajectory seen from the world. The motion of the follower is shown using the position of the TCP and the orientation of the follower. The force on the follower is depicted by the red arrows at the corresponding position of the TCP. The initial ISA frames for motion and wrench, necessary for reconstruction, are also depicted.

TABLE V

RECONSTRUCTION ERROR FOR THE MOTION AND FORCE TRAJECTORY OF FIGURE 11. THE CHOSEN TOLERANCES ARE GIVEN AS A REFERENCE.

Trajectory	RMS reconstruction error	Tolerance in OCP
position	$2.00 + 7 \times 10^{-9}$ mm	$\epsilon_p = 2.00$ mm
orientation	$2.00 + 9 \times 10^{-9}$ °	$\epsilon_R = 2.00^\circ$
force	$0.80 + 4 \times 10^{-9}$ N	$\epsilon_f = 0.80$ N
moment	$0.16 + 8 \times 10^{-9}$ Nm	$\epsilon_m = 0.16$ Nm

trials. Table V compares these reconstructed trajectories with the measured trajectories on which a global transformation has been applied so that they are in the same location. The RMS-differences between the trajectories were found to be in agreement with the chosen tolerances for the trajectory accuracy constraint in the OCPs. The small difference is due to the convergence tolerance of the numerical solver of the OCPs, which was set to 10^{-8} . These results confirm that the trajectories reconstructed from the calculated invariants have the accuracy specified in the OCPs and are also driftless. Evidently, these results also hold for the special case of a reconstruction at the same location as the measurements.

VI. APPLICATION TO PEG-ON-HOLE ALIGNMENT

This section examines motion and force trajectories resulting from a human-demonstrated peg-on-hole task in which the operator holds a tool to which the peg with diameter 50 mm is attached, as shown in Fig. 12. The task aims to ensure that peg and hole are aligned before peg insertion starts, even if the hole is inaccurately located. To accomplish this, the operator first establishes a stable three-point contact between the hole and a purposefully misaligned peg, as shown in Fig. 12c. This contact involves two rim-rim contact points and one surface-rim contact point. We recorded 12 demonstrations of the alignment, each starting from such initial three-point contact and finishing at full alignment. The operator was instructed to perform alignment while maintaining the three-point contact. All demonstrations were performed by the same operator, but in two batches, respectively of 7 and of 5 trials, at different points in time.

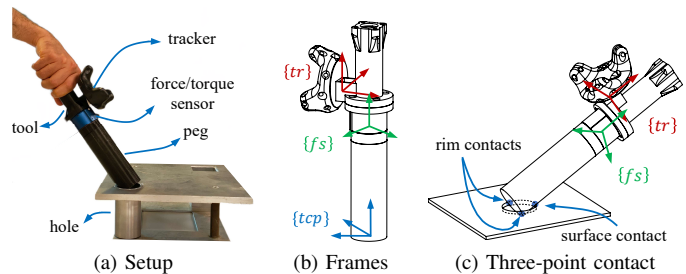


Fig. 12. Peg-on-hole setup with defined frames and illustration of the contact.

Although this task appears to involve a pure rotation, previous theoretical studies have shown that, apart from a rotation, the motion involves a small, but non-zero translation, see e.g. [46]. It is evident that, due to the three-point contact and neglecting friction, this task involves two instantaneous 3-dof vector spaces: one for the wrench and one for the motion. So, the operator can use different motion and wrench strategies during the demonstration.

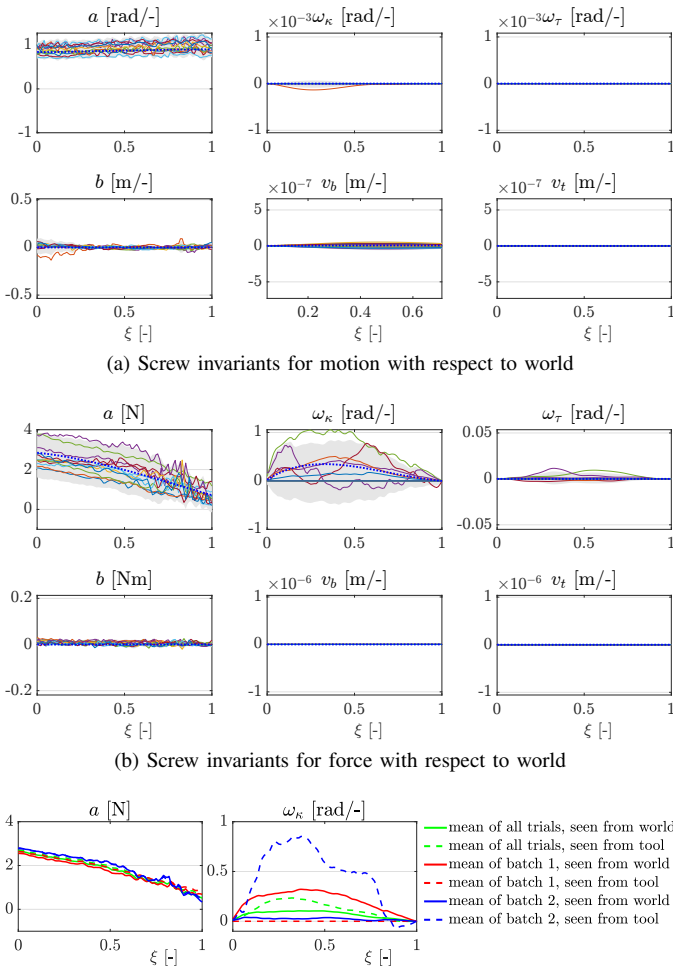
A. Experimental set-up and data processing

The motion capture system is the same as for the contour following task of Section V. The force/torque sensor is again a 6-axis JR3 sensor with a nominal accuracy of $\pm 1\%$, but its measurement ranges are smaller: ± 200 N along Z and ± 100 N along X and Y for force, and ± 5 Nm in all directions for moment. The same three frames as on the contour following tool were defined on the demonstration tool here (see Fig. 12b), with the TCP frame located at the tip of the peg.

Unlike in the contour following application, we did not use different tracker positions on the tool and did not artificially transform the pose measurements by changing the world frame, because invariance with respect to such variations was sufficiently proven in the contour following experiment. We also did not transform the wrench measurements to different locations on the tool, but we did transform them to a frame fixed to the world to investigate the wrench invariants with respect to the world.

Data was processed in roughly the same way as in the contour following application, with the following changes or additions. The segmentation thresholds were set to 1 N and 0.35 rad/s for force and angular velocity, respectively. Since the task is predominantly rotational, the integral of the rotational velocity was chosen as the progress variable to reparameterize the time-based input trajectories, and it was also rescaled to a dimensionless variable ranging from 0 to 1. The initialization of the OCPs based on average moving frames (Section III-B and IV-B) was done on the complete measured trajectories. Furthermore, since the force/torque sensor was more accurate than the one used for contour following (same relative accuracy for a smaller range), we lowered tolerances ϵ_f and ϵ_m in the OCPs to 0.3 N and 0.1 Nm, respectively.

Finally, to investigate if the operator used different wrench strategies in the two batches, we calculated the invariants corresponding to the average of the wrench trajectories for each of the two batches and compared them to the invariants corresponding to the average of all demonstrations. This was



(c) First two screw invariants for the averaged wrench trajectories of measurement batch 1 and 2, with respect to either world or tool.

Fig. 13. Screw invariants for motion and force for the peg-on-hole task.

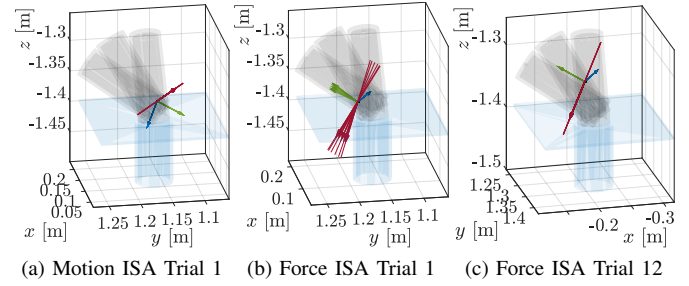
done both with respect to the tool and with respect to the world. Three remarks have to be made here: 1) we took the average of the wrench trajectories rather than the average of the invariants, because averaging is a linear operation, but the subsequent calculation of the invariants is not; 2) we can just average the measured wrench trajectories, because all wrench trajectories were recorded at the same physical sensor location on the tool and were additionally transformed to the same virtual location in the world⁴; 3) because averaged wrench trajectories are less noisy, we could further lower ϵ_f to 0.17 N and ϵ_m to 0.05 Nm.

B. Results and discussion

Figure 13 and 14 summarize the experimental results, focusing on screw invariants for motion and force. Additional results for the vector invariants can be generated in the provided software [34].

Motion invariants with respect to world: Fig. 13a shows the screw invariants of motion. The object invariant a corresponds to the relative rotation of the peg with respect to

⁴If the wrench trajectories are recorded at different locations, we can always calculate the invariants of each wrench trajectory and then reconstruct all wrench trajectories at a common new location, as explained in Section V-D.



(a) Motion ISA Trial 1 (b) Force ISA Trial 1 (c) Force ISA Trial 12

Fig. 14. Moving frames for trials in the peg-on-hole alignment task, all seen from the world frame. The ISAs are depicted using red arrows: (a) Moving frames calculated from the tool's motion for Trial 1 (batch 1), showing a stationary ISA; (b) Moving frames calculated from the wrench for Trial 1 (batch 1), showing a moving ISA; (c) Moving frames of the wrench for Trial 12 (batch 2), showing a stationary ISA.

the hole around the screw axis. Its value is almost constant because the re-parameterization is based on the rotation and it lies around 1 rad/-. This makes sense since the integral of a corresponds to the initial misalignment angle, which is about 1 rad. Object invariant b is small, meaning there is almost no translation along the screw axis. The moving frame invariants ω_κ , ω_τ , v_b , and v_t are zero or near zero, indicating that the moving frame is stationary. Hence, with the chosen values of the tolerances in the OCP, the relative motion between peg and hole can be explained by a pure rotation about a fixed axis. As a result, the small theoretical translation mentioned earlier vanishes due to the chosen tolerances. This conclusion holds for all twelve trials, indicating that the operator used the same motion strategy in all trials: a pure alignment of peg and hole achieved by a planar motion, without exciting the two other relative degrees of freedom, being relative rotations about peg or hole axes. This is confirmed by Fig. 14a where all the instantaneous screw axes and frames coincide.

Force invariants with respect to world: Fig. 13b shows the screw invariants for the individual demonstration trials. Object invariant a , which corresponds to the magnitude of the force vector, is remarkably small, going from less than 4N to zero during the alignment. As for the moving frame invariants, only ω_κ is significantly different from zero, at least for six of the twelve trials. This indicates that, for those six trials the instantaneous screw axis (i.e. the force vector) is rotating with respect to the world in a plane about a fixed point; for the other six trials the instantaneous screw axis remains fixed to the world, i.e. the force vector does not rotate. This in turn indicates that the operator used different wrench strategies in the trials.

Force invariants with respect to world and tool for averaged wrench trajectories: Based on the previous result, we hypothesized that the operator had used different wrench strategies in the two batches of demonstration trials, because they were executed at different points in time. Figure 13c therefore shows the screw invariants for the averaged wrench trajectories of batch 1 (red), batch 2 (blue), and the whole set of trials (green). We show both the invariants with respect to the world (solid lines) and to the tool (dashed lines). Only the first two rotational invariants are shown, as the other invariants

are zero (v_b, v_t) or almost zero (ω_τ, b), similarly as in Fig. 13b. Visually, there is an evident distinction between the trials of batch 1 and batch 2: while for batch 1 the average of the trials shows a force vector that rotates with respect to the world and remains fixed with respect to the peg, the opposite is true for the average of the trials of batch 2. To illustrate this further, we plot the resulting moving frames for one trial from batch 1 and one trial from batch 2 in Figures 14b and 14c. These figures clearly show that the ISA is rotating for the trial from batch 1 while the ISA is stationary for the trial from batch 2. A more in-depth study and statistical analysis of these wrench strategies go beyond the scope of this paper, but we conclude that different strategies can lead to successful alignments. This is a valuable result for designing robot force control strategies. It is also interesting to see that it could be obtained in an experiment with only small contact forces.

VII. DISCUSSION AND CONCLUSION

The aim of this work was to introduce local invariant descriptors for force and moment trajectories. Inspired by the invariant descriptors for motion, we exploited the duality between motion and force. We extended existing concepts for motion trajectories to general vector and screw trajectories, and then applied them to force and moment trajectories.

Two types of descriptors were introduced: vector and screw invariants. Vector invariants describe 3D vector trajectories (translational velocities, rotational velocities, forces, and moments) while screw invariants describe 6D screw trajectories (twists and wrenches). Both types of descriptors are invariant for the choice of world reference frame, but the screw invariants are also independent of the reference point on the moving body that is chosen to represent the translational velocity or the moment. Hence, for obtaining the screw invariants there is no need to choose a reference point on the moving body. This is particularly useful in cases where there is no natural choice for such reference point, the reference point can change between executions, or the calibration of the reference point is time consuming. The newly proposed descriptors can be applied to both motion and force trajectories, as opposed to existing invariant descriptors, which were mainly intended for point curves and/or motion trajectories [16], [19]–[27].

Invariant properties allow to recognize and model tasks that were demonstrated in different environments, without the need for accurate calibration. For trajectory reconstruction and generation however, calibration efforts cannot be avoided since for these applications we need initial values as pointed out in the related paragraphs in Sections III and IV. More in detail, we need initial values for the orientation $\tilde{\mathbf{R}}_1$ or location $\tilde{\mathbf{T}}_1$ of the moving frame, and, if we reconstruct the object trajectory up to position and orientation, we also need respective initial values \mathbf{p}_1 and \mathbf{R}_1 . In practical applications, these initial values are to be derived from environment models or from sensor measurements, either directly or using model-based state estimators.

Following a similar line of reasoning, the invariants themselves cannot be used to describe or check relations *between* motion and force, such as reciprocity between a screw twist

and a screw wrench, unless the transformation between the respective moving frames in which the motion and force invariants are represented is known. In general, this requires calibration. For example, to check the *reciprocity condition*, i.e., the condition that no power is dissipated in the contact: $\mathbf{f} \cdot \mathbf{v} + \mathbf{m} \cdot \boldsymbol{\omega} = 0$, it is required that both twist and wrench, $\boldsymbol{\mathcal{T}}$ and $\boldsymbol{\mathcal{W}}$, are expressed in the same reference frame and have the same reference point. Therefore, we cannot just plug in the \mathbf{a} and \mathbf{b} object invariants of the respective screw invariants, because they are defined relative to their respective moving frames, and, in turn, to the frames in which the twist and wrench trajectories were measured. Hence, to study relations between motion and force, we need a calibration between the respective measurement frames.

The vector and screw invariants can be made independent of time, and hence independent of the applied time-based motion profile, by expressing them as a function of a suitable geometric progress variable. Again, this additional invariance is useful for the recognition of a task demonstrated in different situations and for the generalization of task models to new situations, as shown in [15] for the case of motion in free space. However, the way we chose the progress variable in the 3D contour following application introduced an important limitation. We did not follow the approach to choose an invariant progress variable that is independent of the reference point for translation, as suggested in [25] and briefly discussed in Section II-C. Instead, we chose the progress variable as the integral of the translational velocity of the origin of $\{tcp\}$. So, based on our understanding of the contact geometry, we handpicked this point as a reference point to define arc length as a representation of task progress, and used it to represent all invariants. The benefit of this approach is that we obtained vector and screw invariants for both motion and force that are interpretable and can be compared with each other and with the reference values of the limit case (Figures 6-8). Definitely, this progress value is a very good choice and probably even the best one for this application, but it would be better to derive a suitable progress value from the measured trajectories in an invariant data-driven way. This is subject of our future work.

Besides analytical procedures to derive the invariant descriptors, an approach was presented based on optimal control, where the main idea is to find the invariant descriptor that reconstructs the given window of measurements. This provides a way to deal with singularities and measurement noise and hence enables a much more robust calculation of the descriptors. Compared to our previous work on the calculation of motion descriptors [33], we improved the stability and repeatability of the calculation by reformulating the OCP. In the reformulated problem, trajectory reconstruction errors became inequality constraints that were bounded by a desired tolerance value. This resulted in more intuitive tuning, as we showed that these tolerance values could be directly related to the accuracy or noise characteristics of the used motion and force sensor. The reformulated OCP also resulted in a more stable behavior of the moving frame. For example, when the values of the noisy moments are below the specified tolerance, they can be described by a stationary moving frame (see Fig. 7d and Fig. 10b). Finally, the reformulation resulted in

a smaller number of tuning parameters, further reducing the tuning effort. Overall, we obtained a good repeatability of the invariants, as can be seen in Figures 6-8, even for noisy data.

Continuity of the moving frames is ensured by minimizing the values of the moving frame invariants in the OCPs. This also provides some degree of continuity for the reconstructed trajectories. To enforce a more specific continuity on the reconstructed trajectory, it is advised to include the time profile as variables in the OCPs in order to enforce acceleration and/or jerk limit constraints. These constraints can be either on the spatial trajectory itself or on the robot joint trajectories in case a kinematic model of the robot is available, similar to [15].

The paper contains a practical experiment involving a human-demonstrated 3D contour following task which showcases the use, verifies the invariant properties, and validates the robust calculation of screw and vector invariant descriptors for both motion and force. As expected, this experiment yielded highly repeatable motion descriptors due to the highly constrained motion (in a 1-dof instantaneous twist space), and less repeatable force descriptors due to human variation (in a 5-dof instantaneous wrench space). Evidently, apart from human variation, other disturbances were present in this experiment such as sensor inaccuracy and measurement noise. A second experiment, involving human demonstration of a peg-on-hole alignment task, showed how motion and force invariants can be used to detect and characterize different human strategies in contact tasks.

The concepts and algorithms presented in this paper are intended as an initial toolbox [34] to support future work on invariant representations of motion and force trajectories and their use in physics-informed machine learning of force-controlled manipulation skills.

APPENDIX

RELATION BETWEEN THE KINEMATICS OF THE FS FRAME AND THE FRENET-SERRET EQUATION

In differential geometry, the Frenet-Serret equation is usually written as follows:

$$\begin{bmatrix} \mathbf{T}' \\ \mathbf{N}' \\ \mathbf{B}' \end{bmatrix} = \begin{bmatrix} 0 & \omega_\kappa & 0 \\ -\omega_\kappa & 0 & \omega_\tau \\ 0 & -\omega_\tau & 0 \end{bmatrix} \begin{bmatrix} \mathbf{T} \\ \mathbf{N} \\ \mathbf{B} \end{bmatrix}, \quad (53)$$

with ω_κ the curvature, ω_τ the torsion, and \mathbf{T} , \mathbf{N} , \mathbf{B} the unit vectors of the FS frame. This equation can be shown to be equal to the kinematics of the FS frame as defined in (18):

$$\tilde{\mathbf{R}}' = \tilde{\mathbf{R}} \begin{bmatrix} 0 & -\omega_\kappa & 0 \\ \omega_\kappa & 0 & -\omega_\tau \\ 0 & \omega_\tau & 0 \end{bmatrix}. \quad (r.18)$$

In the previous expression, first expand $\tilde{\mathbf{R}}$ into its individual columns using the unit vectors defined in (15):

$$[\mathbf{e}'_t \quad \mathbf{e}'_n \quad \mathbf{e}'_b] = [\mathbf{e}_t \quad \mathbf{e}_n \quad \mathbf{e}_b] \begin{bmatrix} 0 & -\omega_\kappa & 0 \\ \omega_\kappa & 0 & -\omega_\tau \\ 0 & \omega_\tau & 0 \end{bmatrix}. \quad (54)$$

Applying the transpose to both sides of (54) results in:

$$\begin{bmatrix} \mathbf{e}'_t{}^T \\ \mathbf{e}'_n{}^T \\ \mathbf{e}'_b{}^T \end{bmatrix} = \begin{bmatrix} 0 & -\omega_\kappa & 0 \\ \omega_\kappa & 0 & -\omega_\tau \\ 0 & \omega_\tau & 0 \end{bmatrix}^T \begin{bmatrix} \mathbf{e}_t{}^T \\ \mathbf{e}_n{}^T \\ \mathbf{e}_b{}^T \end{bmatrix}. \quad (55)$$

Since the transpose of a skew-symmetric matrix is equal to the negative of the matrix, (55) corresponds to the Frenet-Serret equation in (53).

REFERENCES

- [1] A. G. Billard, S. Calinon, and R. Dillmann, *Springer Handbook of Robotics*. Cham, Switzerland: Springer, 2016, ch. Learning from humans, pp. 1995–2014.
- [2] M. Werling, L. Gröll, and G. Bretthauer, “Invariant trajectory tracking with a full-size autonomous road vehicle,” *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 758–765, 2010.
- [3] J. Duffy, “The fallacy of modern hybrid control theory that is based on orthogonal complements of twist and wrench spaces,” *Journal of Robotic Systems*, vol. 7, no. 2, pp. 139–144, 1990.
- [4] H. Lipkin, “Geometry and mappings of screws with applications to the hybrid control of robotic manipulators,” Ph.D. dissertation, University of Florida, 1985.
- [5] H. Lipkin and J. Duffy, “Hybrid twist and wrench control for a robotic manipulator,” *Journal of Mechanisms, Transmissions, and Automation in Design*, vol. 110, no. 2, pp. 138–144, 06 1988.
- [6] H. Bruyninckx and O. Khatib, “Gauss’ principle and the dynamics of redundant and constrained manipulators,” in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 3, 2000, pp. 2563–2568 vol.3.
- [7] J. De Schutter, D. Torfs, H. Bruyninckx, and S. Dutr e, “Invariant hybrid force/position control of a velocity controlled robot with compliant end effector using modal decoupling,” *The International Journal of Robotics Research*, vol. 16, no. 3, pp. 340–356, 1997.
- [8] A. Barrau and S. Bonnabel, “The invariant extended kalman filter as a stable observer,” *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1797–1812, 2016.
- [9] S. Bonnabel, “Left-invariant extended kalman filter and attitude estimation,” in *2007 46th IEEE Conference on Decision and Control*, 2007, pp. 1027–1032.
- [10] D. Bennequin, R. Fuchs, A. Berthoz, and T. Flash, “Movement timing and invariance arise from several geometries,” *PLOS Computational Biology*, vol. 5, no. 7, pp. 1–27, 07 2009.
- [11] Y. Meirovitch, D. Bennequin, and T. Flash, “Geometrical invariance and smoothness maximization for task-space movement generation,” *IEEE Transactions on Robotics*, vol. 32, no. 4, pp. 837–853, 2016.
- [12] Q. C. Pham and Y. Nakamura, “A new trajectory deformation algorithm based on affine transformations,” *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 1054–1063, Aug 2015.
- [13] C. Belta and V. Kumar, “Euclidean metrics for motion generation on $se(3)$,” *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 216, no. 1, pp. 47–60, 2002.
- [14] R. Laha, R. Sun, W. Wu, D. Mahalingam, N. Chakraborty, L. F. Figueredo, and S. Haddadin, “Coordinate invariant user-guided constrained path planning with reactive rapidly expanding plane-oriented escaping trees,” in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 977–984.
- [15] M. Vochten, T. De Laet, and J. De Schutter, “Generalizing demonstrated motion trajectories using coordinate-free shape descriptors,” *Robotics and Autonomous Systems*, vol. 122, p. 103291, 2019.
- [16] E. Calabi, P. J. Olver, C. Shakiban, A. Tannenbaum, and S. Haker, “Differential and numerically invariant signature curves applied to object recognition,” *International Journal of Computer Vision*, vol. 26, no. 2, pp. 107–135, Feb 1998.
- [17] S. Wu and Y. F. Li, “On signature invariants for effective motion trajectory recognition,” *The International Journal of Robotics Research*, vol. 27, no. 8, pp. 895–917, 2008.
- [18] I. Weiss, “Geometric invariants and object recognition,” *International Journal of Computer Vision*, vol. 10, no. 3, pp. 207–231, 1993.
- [19] Y. Piao, K. Hayakawa, and J. Sato, “Space-time invariants for recognizing 3D motions from arbitrary viewpoints under perspective projection,” *Transactions of the Institute of Electronics, Information and Communication Engineers*, vol. E89-D, no. 7, pp. 2268–2274, 2006.
- [20] S. Wu and Y. F. Li, “Flexible signature descriptions for adaptive motion trajectory representation, perception and recognition,” *Pattern Recognition*, vol. 42, no. 1, pp. 194–214, 2009.
- [21] Z. Shao and Y. Li, “Integral invariants for space motion trajectory matching and recognition,” *Pattern Recognition*, vol. 48, no. 8, pp. 2418–2432, 2015.

- [22] C. Rao, A. Yilmaz, and M. Shah, “View-invariant representation and recognition of actions,” *International Journal of Computer Vision*, vol. 50, no. 2, pp. 203–226, 2002.
- [23] Y. Guo, Y. Li, and Z. Shao, “Dsrfr: A flexible trajectory descriptor for articulated human action recognition,” *Pattern Recognition*, vol. 76, pp. 137–148, 2018.
- [24] —, “Rrv: A spatiotemporal descriptor for rigid body motion recognition,” *IEEE transactions on cybernetics*, vol. 48, no. 5, pp. 1513–1525, 2017.
- [25] J. De Schutter, “Invariant description of rigid body motion trajectories,” *Journal of Mechanisms and Robotics*, vol. 2, no. 1, pp. 011004/1–9, 2010.
- [26] D. Lee, R. Soloperto, and M. Saveriano, “Bidirectional invariant representation of rigid body motions and its application to gesture recognition and reproduction,” *Autonomous Robots*, vol. 42, no. 1, pp. 125–145, Jan. 2018.
- [27] M. Vochten, T. De Laet, and J. De Schutter, “Comparison of rigid body motion trajectory descriptors for motion representation and recognition,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 3010–3017.
- [28] H. Guggenheimer, *Differential Geometry*, ser. McGraw-Hill series in higher mathematics. Dover Publications, 1977.
- [29] T. Moons, E. J. Pauwels, L. Van Gool, and A. Oosterlinck, “Foundations of semi-differential invariants,” *International Journal of Computer Vision*, vol. 14, no. 1, pp. 25–47, 1995.
- [30] S. Wu and Y. Li, “Motion trajectory reproduction from generalized signature description,” *Pattern Recognition*, vol. 43, no. 1, pp. 204–221, 2010.
- [31] R. T. Arn, P. Narayana, T. Emerson, B. A. Draper, M. Kirby, and C. Peterson, “Motion segmentation via generalized curvatures,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 12, pp. 2919–2932, 2018.
- [32] G. Perantoni and D. J. Limebeer, “Optimal control of a formula one car on a three-dimensional track—part 1: Track modeling and identification,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 5, p. 051018, 2015.
- [33] M. Vochten, T. De Laet, and J. De Schutter, “Robust optimization-based calculation of invariant trajectory representations for point and rigid-body motion,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 5598–5605.
- [34] “Invariant trajectory descriptors for motion and force,” 2023. [Online]. Available: https://gitlab.kuleuven.be/robotgenskill/public_code/invariants_mat
- [35] R. S. Ball, *A Treatise on the Theory of Screws*. Cambridge university press, 1900 (republished 1998).
- [36] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 1994.
- [37] A. Gray, E. Abbena, S. Salamon *et al.*, *Modern differential geometry of curves and surfaces with Mathematica*, 3rd ed. CRC press, 2006.
- [38] T. De Laet, S. Bellens, R. Smits, E. Aertbelien, H. Bruyninckx, and J. De Schutter, “Geometric relations between rigid bodies (part 1): Semantics for standardization,” *IEEE Robotics Automation Magazine*, vol. 20, no. 1, pp. 84–93, March 2013.
- [39] R. L. Bishop, “There is more than one way to frame a curve,” *American Mathematical Monthly*, pp. 246–251, 1975.
- [40] A. Ancillao, M. Vochten, A. Verduyn, J. De Schutter, and E. Aertbelien, “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*, vol. 17, no. 10, pp. 1–22, 10 2022.
- [41] D. Huynh, “Metrics for 3d rotations: Comparison and analysis,” *Journal of Mathematical Imaging and Vision*, vol. 35, pp. 155–164, 10 2009.
- [42] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2014.
- [43] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “CasADI – A software framework for nonlinear optimization and optimal control,” *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.
- [44] A. Wächter and L. T. Biegler, “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.
- [45] O. Sorkine-Hornung and M. Rabinovich, “Least-squares rigid motion using svd,” *Computing*, vol. 1, no. 1, pp. 1–5, 2017.
- [46] H. Bruyninckx, S. Dutre, and J. De Schutter, “Peg-on-hole: a model based solution to peg and hole alignment,” in *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, vol. 2. IEEE, 1995, pp. 1919–1924.



collaboration.



physical human-robot interaction.



the capturing and modeling of craftsmanship skills. Additionally, he has also a research activity in quantitative analysis of spasticity and related phenomena.



programming by human demonstration; and generalizing human-demonstrated robot skills to new situations. In 2018 he received an ERC Advanced Grant (ROBOTGENSKILL) on this last topic.

Maxim Vochten is a Postdoctoral Researcher within the Robotics Research Group at KU Leuven, Belgium. He earned his MSc and PhD degrees in Mechanical Engineering from KU Leuven, in 2013 and 2018, respectively. His doctoral work focused on data-efficient trajectory learning with application to motion recognition and generalization, using concepts from differential geometry, screw theory, and optimal control. His current research interests include human intent estimation, robot trajectory generation, and reactive control for human-robot

Ali Mousavi Mohammadi earned his B.Sc. and M.Sc. degrees in Mechanical Engineering from Ferdowsi University of Mashhad, Iran, in 2013 and 2016, respectively. In the two years following his graduation, he worked as a part-time researcher at FUM Robotics Lab. In 2019, he joined the Robotics Research Group at KU Leuven’s Mechanical Engineering Department. Currently, he is a PhD candidate carrying out his research on force-based manipulation tasks learning. His research interests include robot programming by demonstration and

Arno Verduyn received a masters degree in Mechanical Engineering at KU Leuven in 2020. He continued his research as a PhD candidate as part of the Robotics Research Group. He is currently working on invariant descriptors of rigid-body trajectories and force trajectories. His research is supervised by Professor Joris De Schutter.

Tinne De Laet is associate professor at the Faculty of Engineering Science, KU Leuven. She obtained a doctoral degree in Mechanical Engineering in 2010 at KU Leuven, Belgium supported by a scholarship of the Research Foundation - Flanders (FWO). She was a post-doctoral researcher of the Research Foundation - Flanders (FWO) at KU Leuven from 2010-2013. Since 2013 she is the Head of the Tutorial Services of Engineering Science where her research focuses on using learning analytics, conceptual learning and metacognition in mechanics.

Erwin Aertbelien was born in Antwerp, Belgium, in 1972. He earned his M.Sc. and Ph.D. degrees in mechanical engineering from the University of Leuven in 1995 and 2008, respectively. He specializes in robot task specification and control. He is the developer and originator of the constraint-based task specification formalism known as eTaSL. His research interests encompass all forms of human-robot collaboration, including lower-limb and upper-limb exoskeletons, co-manipulation with robot manipulators, teleoperation using brain interfaces, and

Joris De Schutter is full professor at KU Leuven. He received the MSc degree in mechanical engineering from KU Leuven, Belgium (1980), the MSc degree from MIT (1981), and the PhD degree in mechanical engineering, also from KU Leuven (1986). He pioneered in robot force control and sensor-based control using optimization, contributing to the iTaSC and eTaSL software frameworks. His current research interests include: robot control and programming based on models, sensors and constrained optimization; human-robot interaction; programming by human demonstration; and generalizing human-demonstrated robot skills to new situations. In 2018 he received an ERC Advanced Grant (ROBOTGENSKILL) on this last topic.