IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Analytical parameter conversion of CPC and POE models

Oleg Sumenkov[1]

Abstract—**This paper presents an analytical framework for parameter conversion between Complete and Parametrically Continuous (CPC) and Product of Exponentials (POE) kinematic models of serial-chain mechanisms. The approach, grounded in Lie group and algebra theory, formulates and proves three key lemmas to enable exact POE-to-CPC parameter conversion. Building upon established POE-DH and CPC-DH transitions, the proposed framework facilitates flexible model selection based on application-specific needs, independent of the initial parameterization. Primarily designed for robot calibration, this method also serves as a unifying tool for comparing and analyzing calibration results across different kinematic conventions. The framework's effectiveness is demonstrated through numerical validation on the PUMA 560 robot, confirming its accuracy and practical applicability.**

Index Terms—**Kinematics, Formal Methods in Robotics and Automation, Calibration and Identification.**

I. INTRODUCTION

CALIBRATION, trajectory planning and motion control problems of serial-link mechanisms depend, among other things, on the choice of a specific kinematic model. Among the various kinematic models developed by now, one of the earliest ones is the Denavit-Hartenberg (DH) convention [1]. The DH convention remains a cornerstone in the field of robotics [2]–[5], serving as a universally accepted standard providing a minimum parameter representation (presumably, with the fastest on-board calculations) and straight-forward frames assignment procedure.

The DH convention has been widely applied in calibrating industrial manipulators for years [6]–[8], starting from the moment of its formulation [9]. Then, in [10] it was noticed that small changes in parallel consecutive joint geometry cause discontinuity in DH parameters (the so-called *parameter singularity*), significantly increasing the difficulty of their identification.

After that, two main characteristics were introduced in [11] to determine a "good" kinematic model: *completeness* and *parametric continuity* (originally called *proportionality*). The *completeness* means that "*... has the capability of relating the joint displacements to the tool pose for any manipulator while allowing for the arbitrary placement of the reference frame and arbitrary assignment of the zero position.*" [12]. The second property is *proportionality* — small changes in the robot geometry cause small variations in the model parameters.

It turns out, that the commonly used DH convention is neither complete nor parametrically continuous. This fact motivated researchers to develop a convention free from of these issues: Hayati model [13], where additional y-axis rotation for nominally parallel consecutive revolute joints was introduced; S-model by [14], where two extra parameters were added to DH that ensure model *completness* and flexible assignment rules; Wu's model [15]; "Zero-reference" model [16] based on Rodrigues' formula, ensuring *continuity*, and many others. However, $\{Base\}$ and $\{Tool\}$ frames cannot be chosen arbitrarily in most model due to the specific assignment rules. In that case, the *completeness* property [11] does not fulfil[1].

Therefore, a convention similar to the DH model was required, ensuring both *parametric continuity* and *completeness* properties. This need was addressed by the Complete and Parametrically Continuous (CPC) model, introduced in [17], [18]. The convention is based on the theoretically minimal four-parameters-based line representation by Roberts [19] that ensures *parametric continuity*. To ensure the *completeness* property, two additional parameters have been introduced for the $\{Base\}$ and $\{Tool\}$ frames.

An alternative approach, called the *Product of Exponentials* (POE), based on Lie group theory, was proposed by R. Brockett [20]. In the POE framework, only the $\{Tool\}$ frame needs to be defined freely to derive the forward kinematics. The forward kinematics in POE describes the displacement and rotation of the $Tool$ frame in the zero-configuration (zero joint angles) as a twist around the specified joint axis, expressed in the $Base$ frame — similar to the "Zero-reference" model [16]. In contrast, the forward kinematics in DH and CPC is represented by the relative movements of adjacent link-fixed frames.

Due to the smoothness of exponential mapping and screw-motion basics, POE possesses both *parametric continuity*

Manuscript received August, 4, 2025; Revised October, 22, 2025; Accepted November, 23, 2025.

This paper was recommended for publication by Editor Della Santina C. upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by a grant of the Russian Science Foundation project No. 23-79-10213 (https://www.rscf.ru/project/23-79-10213/).

[1] The author is with Mathematical Robotics Science Division, Sirius University of Science and Technology, Sirius Federal Territory, Russian Federation `sumenkov.oy@talantiuspeh.ru`.

Digital Object Identifier (DOI): see top of this page.

[1] For "Zero-reference" both properties are fulfilled, but the error model derivation is based on strong assumptions [17]

and *completeness* properties. However, the *minimality* and identifiability of POE parameters remained uncovered before [21], which had raised doubts regarding the robustness of the identification algorithms dealing with POE.

Initially, it was believed that the POE is much more complicated in terms of model description and purely computational terms compared to the DH. Nevertheless, in [22] it was clearly shown that the POE calculation is less computationally expensive. Later, in [23], [24] it was illustrated in numerous examples that POE model design is natural and intuitive, with no need to keep in mind assignment rules compared to other conventions.

Currently, while POE and DH are the most widely adopted kinematic models in robotics [24], [25], the CPC model proposed by [17] is frequently referenced [26]–[29] in discussions on singularity-free kinematic representations, particularly in robot calibration and parameter identification area.

In this scenario, the selection of the appropriate convention is contingent upon a thorough analysis of an application's specifics and a problem context. Thus, parameter conversion algorithms, which enable the representation of one kinematic model in terms of another analytically, hold significant interest. Most of these conversions have already been performed in the context of the models discussed as shown in Fig. 1.

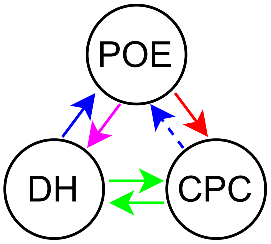

Fig. 1. Conversions schemes between DH, CPC, and POE models

The **green** arrows denote transitions proven in [17], [18], where the general scheme to extract CPC parameters from *any* homogeneous transformations in close form is given. The simplicity of the proposed transitions is one of the strong arguments that supports the CPC model. However, it is unclear how to perform **POE–CPC** in this framework. The **blue** arrow represents the conversion that was initially suggested for **DH–POE** transition in [23], [24], [30], which can be easily applied to **CPC–POE** as well (**dashed blue** arrow). Lastly, the **magenta** arrow highlights an exceptional result on the analytical **POE–DH** conversion, presented recently in [25], [31], which served as a *motivation* for the paper.

The paper extends existing conversion schemes by proposing an analytical framework for the **POE–CPC** transition (marked with **red** arrow in Fig. 1). This framework is based on the methodology proposed in [20] and further developed in [23]–[25].

The motivation for the proposed solution can be briefly outlined as follows: 1) complementing the existing DH, CPC, and POE conversion scheme in Fig. 1 made by [17], [23]–[25], [30], [31]; 2) provides comparability of existing methods and techniques, mostly in the calibration and identification fields, performed in different conventions, for instance, [17], [21], [30], [32]–[34]; 3) allows to choose more *convenient convention* for a particular problem and then go back to the *required model regardless of the specified model*, using the proposed framework.

The structure of this article is outlined as follows: Section II introduces the foundational concepts related to CPC and POE models, setting the groundwork for the discussion. Section III demonstrates the main contribution, detailing the proposed parameter conversion framework. Section IV presents the simulation results of the parameter conversion framework of the PUMA 560 model.

A. Notation

In the paper, matrices are in bold uppercase (e.g. $\mathbf{T} \in \mathbb{SE}(3)$), vectors in bold italic lowercase, i.e., $\boldsymbol{b} \in \mathbb{R}^3$, the scalars are in italic lowercase ($q \in \mathbb{R}$). Here and below, $\mathbf{Rot}_{\boldsymbol{k}}(\gamma) \in \mathbb{SE}(3)$ represents rotation about vector $\boldsymbol{k}$ by angle γ, $\mathbf{Tran}_{\boldsymbol{l}}(v) \in \mathbb{SE}(3)$ signifies translation along vector $\boldsymbol{l}$ by distance v, $\mathbf{Tran}_{[\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}]}(\boldsymbol{l})$ understood as $\mathbf{Tran}_{\boldsymbol{x}}(l_x)\,\mathbf{Tran}_{\boldsymbol{y}}(l_y)\,\mathbf{Tran}_{\boldsymbol{z}}(l_z)$ with $\boldsymbol{l} = \begin{bmatrix} l_x & l_y & l_z \end{bmatrix}^\top$. The $\widetilde{\mathbf{R}}$ denotes $\mathbb{SO}(3)$ part of corresponding $\mathbf{R} \in \mathbb{SE}(3)$, $\{\cdot\}$ stands for the frame notation (for example, $\{Base\}$ frame). Operator $\hat{\cdot}$ denotes mapping from vector space into corresponding Lie algebra structure: $\mathbb{R}^3 \mapsto so(3)$ or $\mathbb{R}^6 \mapsto se(3)$, depending on the context. Cross product of vectors $\boldsymbol{a}$, $\boldsymbol{b} \in \mathbb{R}^3$ is denoted as $\boldsymbol{a} \times \boldsymbol{b}$.

II. PRELIMINARIES

The section provides brief information on the CPC and POE model on chain links frame assignment and representation.

A. Complete and Parametrically Continuous model

The complete and parametrically continuous (CPC) model proposed in [17], [18] ensures both *parametric continuity* and *completeness* simultaneously while being as formalised as the DH convention. In this framework, the homogeneous transformation matrix $\mathbf{T}$ from $\{i\}$ to $\{i-1\}$ frame is decoupled following the idea of [35] (indices are omitted)

$$\mathbf{T} = \mathbf{Q}\,\mathbf{V}, \tag{1}$$

where $\mathbf{Q}$ is the *motion matrix* that describes the joint movement, i.e., $\mathbf{Q} = \mathbf{Rot}_{\boldsymbol{z}}(q)$ for revolute joint and $\mathbf{Tran}_{\boldsymbol{z}}(q)$ for prismatic joint. In turn, the *shape matrix* $\mathbf{V}$ defines link geometry and is mostly based on theoretically minimal line representation, shown in [19]. The line is defined by a unit vector $\boldsymbol{b}$ ($||\boldsymbol{b}|| = 1$), that is restricted by upper half space of the origin frame, and the projection (decomposed into l_x and l_y) of the line point on a plane passing through the origin with normal $\boldsymbol{b}$ (see [17], Sec. II). In this case [19], the *shape matrix* takes the form

$$\mathbf{V} = \mathbf{Rot}_{\boldsymbol{k}}(\alpha)\,\mathbf{Tran}_{\boldsymbol{x}}(l_x)\,\mathbf{Tran}_{\boldsymbol{y}}(l_y), \tag{2}$$

where $\boldsymbol{k} = (\boldsymbol{z} \times \boldsymbol{b})\,/\,||\boldsymbol{z} \times \boldsymbol{b}|| = \begin{bmatrix} -b_y/\sqrt{b_y^2+b_x^2} & -b_x/\sqrt{b_y^2+b_x^2} & 0 \end{bmatrix}^\top$, $\alpha = \arccos\big(\boldsymbol{z}^\top \boldsymbol{b}\big) = \arccos\big(b_z\big)$, $\boldsymbol{b} = \begin{bmatrix} b_x & b_y & \sqrt{1-b_x^2-b_y^2} \end{bmatrix}^\top$. It was suggested [17] to supplement (2) with two extra parameters: β, that depends on the choice of the $\boldsymbol{x}$-axis direction, and l_z to enable arbitrary frame assignment, in other words, *completeness* property

$$\mathbf{V} = \mathbf{R}(\boldsymbol{b})\,\mathbf{Rot}_{\boldsymbol{z}}(\beta)\,\mathbf{Tran}_{[\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}]}(\boldsymbol{l}), \tag{3}$$

where $\boldsymbol{l} = \begin{bmatrix} l_x & l_y & l_z \end{bmatrix}^\top$; $\mathbf{Tran}_{[\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}]}(\boldsymbol{l})$ understood as sequential movements on l_x along $\boldsymbol{x}$, on l_y along $\boldsymbol{y}$, on l_z along $\boldsymbol{z}$; $\mathbf{R}(\boldsymbol{b})$ is $\mathbf{Rot}_{\boldsymbol{k}}(\alpha)$ that is explicitly rewritten in terms of $\boldsymbol{b}$ based on CPC model assumptions

$$\mathbf{R}(\boldsymbol{b}) = \begin{bmatrix} 1-\frac{b_x^2}{1+b_z} & \frac{-b_x b_y}{1+b_z} & b_x & 0 \\ \frac{-b_x b_y}{1+b_z} & 1-\frac{b_y^2}{1+b_z} & b_y & 0 \\ -b_x & -b_y & b_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4}$$

Visualized transformations of (3) are shown below in Fig. 2.

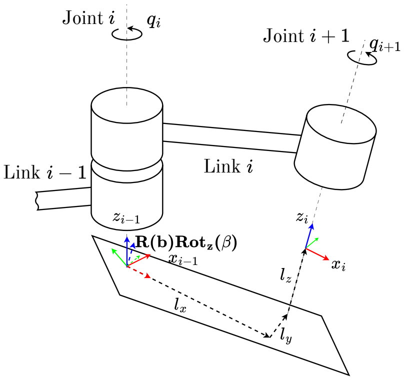

Fig. 2. CPC modelling convention for a revolute joint

It should be emphasized that the seven-parameter model might be reduced by introducing additional assignment rules that are quite convenient. Specifically, it was demonstrated in [17] that l_z and β parameters of (3) are redundant ($l_z = 0$ and $\beta = 0$) for all frames, except $\{Base\}$ and $\{Tool\}$.

B. Product of Exponential model

In turn, it was proposed in [20] to model open-loop kinematics chains using the POE formulas. These studies were first implemented in kinematic calibration problems in [30], highlighting the following advantages: 1) no need to perform frames assignment to each link — only $\{Tool\}$ frame in manipulator zero configuration ($\boldsymbol{q} = 0$) required, similar to [16] model; 2) providing closed-form solution of linearized forward kinematics equations, based on screw theory; 3) *parametric continuity* property naturally holds due to the smoothness of exponential mapping; 4) offering a generalized kinematic description that might be useful in various applications, not only in kinematic calibration.

Thus, we have to define $\mathbf{T}_{Tool} \in \mathbb{SE}(3)$ what is $\{Tool\}$ pose in $\{Base\}$ frame at zero configuration ($\boldsymbol{q} = \mathbf{0}$). Then, let each joint correspond to a *twist* $\boldsymbol{\xi} \in \mathbb{R}^6$ in terms of screw theory [20], [24]. Namely, it takes the following form in the case of *revolute* and *helical* (with helix pitch h) joints

$$\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{p} \times \boldsymbol{\omega} \end{bmatrix}, \tag{5}$$

where $\boldsymbol{\omega}$ is a unit vector of the joint axis ($||\boldsymbol{\omega}|| = 1$), $\boldsymbol{p}$ is a point, lying the joint axis, with $\boldsymbol{\omega}^\top \boldsymbol{v} = 0$ and $\boldsymbol{\omega}^\top \boldsymbol{v} = h$ for *revolute* and *helical* joints, respectively (as shown in Fig. 3).

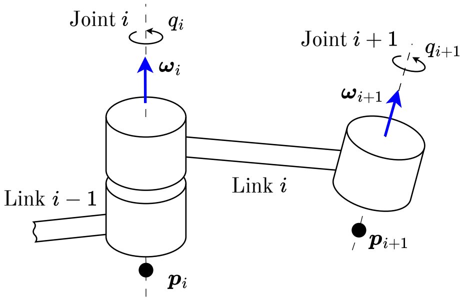

Fig. 3. POE modelling convention for a revolute joint

A *prismatic* joint $\boldsymbol{\xi}$ is defined as

$$\boldsymbol{\xi} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{v} \end{bmatrix}, \tag{6}$$

where $\boldsymbol{v}$ is a unit vector of the joint axis ($||\boldsymbol{v}|| = 1$).

In addition, a twist vector $\boldsymbol{\xi}$ can be rewritten in matrix form with Lie algebra $se(3)$ structure using operator $\hat{\ }$ that denotes mapping into the corresponding Lie algebra structure, i.e., $\hat{\ }: \mathbb{R}^3 \mapsto so(3)$ or $\mathbb{R}^6 \mapsto se(3)$, depending on the context:

$$\hat{\boldsymbol{\xi}} = \begin{bmatrix} \hat{\boldsymbol{\omega}} & \boldsymbol{v} \\ \mathbf{0}^\top & 0 \end{bmatrix}, \quad \hat{\boldsymbol{\omega}} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \tag{7}$$

Then, to obtain forward kinematics one has to remind [20] that the Lie algebra element $\boldsymbol{\xi}$ (*joint twist*) and joint variable $q \in \mathbb{S}$ form the corresponding one-parameter Lie subgroup $\mathbf{T} \in \mathbb{SE}(3)$ via *exponential mapping*

$$\mathbf{T} = \exp(q\hat{\boldsymbol{\xi}}). \tag{8}$$

For example, consider a joint rotating about the $\mathbf{z}$-axis by an angle q, with the transformation $\mathbf{T} = \mathbf{Rot}_{\boldsymbol{z}}(q) = \exp(q\hat{\boldsymbol{\xi}}_{rev})$, where the twist $\boldsymbol{\xi}$ takes the form

$$\boldsymbol{\xi}_{rev} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^\top. \tag{9}$$

Similarly, for a *prismatic* joint with the transformation $\mathbf{T} = \mathbf{Tran}_{\boldsymbol{z}}(q) = \exp(q\hat{\boldsymbol{\xi}}_{psm})$, where

$$\boldsymbol{\xi}_{psm} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^\top, \tag{10}$$

reflecting pure translation along the $\mathbf{z}$-axis. Finally, a *helical* joint, combining rotation and translation, has the transformation $\mathbf{T} = \mathbf{Rot}_{\boldsymbol{z}}(q), \mathbf{Tran}_{\boldsymbol{z}}(qh) = \exp(q\hat{\boldsymbol{\xi}}_{hel})$, where

$$\boldsymbol{\xi}_{\text{hel}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & h \end{bmatrix}^\top, \tag{11}$$

with a helix pitch h. Therefore, choosing appropriate joint twists (5),(6) and $\mathbf{T}_{Tool}$ as $\{Tool\}$ frame pose at $\boldsymbol{q} = \mathbf{0}$ (zero configuration), where q_i are joint variables $i = \overline{1, n}$, forward kinematics for serial n-linked mechanism can be expressed using (8) as

$$\mathbf{T}_0^n = \exp(q_1\hat{\boldsymbol{\xi}}_1)\,\exp(q_2\hat{\boldsymbol{\xi}}_2)\cdot\ldots\cdot\exp(q_n\boldsymbol{\xi}_n)\,\mathbf{T}_{Tool}. \tag{12}$$

Another important property from Lie group theory [36] is that for any given twist vectors $\hat{\boldsymbol{\xi}}_1$, $\hat{\boldsymbol{\xi}}_2 \in se(3)$, the adjoint representation relates the twists via

$$\hat{\boldsymbol{\xi}}_1 = \mathbf{T}\,\hat{\boldsymbol{\xi}}_2\,\mathbf{T}^{-1}, \tag{13}$$

where $\mathbf{T} \in \mathbb{SE}(3)$. Then, (13) can be rewritten [25] using *adjoint mapping* $Ad(\mathbf{T}) : se(3) \mapsto se(3)$, in linear form

$$\boldsymbol{\xi}_1 = Ad(\mathbf{T})\,\boldsymbol{\xi}_2, \tag{14}$$

where adjoint transformation has the form

$$Ad(\mathbf{T}) = \begin{bmatrix} \widetilde{\mathbf{T}} & \mathbf{0}_{3\times3} \\ \hat{\boldsymbol{t}}\,\widetilde{\mathbf{T}} & \widetilde{\mathbf{T}} \end{bmatrix}, \tag{15}$$

where $\widetilde{\mathbf{T}}$ and $\boldsymbol{t}$ are $\mathbb{SO}(3)$ and $\mathbb{R}^3$ parts of $\mathbf{T}$, respectively.

III. MAIN RESULTS

As was stated, the paper proposes an analytical **POE–CPC** parameter conversion framework using the methodology consistently developed in [20], [23]–[25]. It should be emphasised again that it is **unclear** how to extract CPC parameters from POE, following the general formula in [17] (section V). That is the aim for the study.

A. POE–CPC conversion

The **POE–CPC** parameter conversion framework provided in **Theorem** 1 is based on the **Lemmas 1-3** and **Statement 1**, which are formulated and proven below.

Lemma 1. *CPC parameters of revolute joint transformation* $\mathbf{T} = \exp(q\hat{\boldsymbol{\xi}})$ *in POE form can be obtained using decomposition* $\mathbf{T} = \mathbf{V}\,\mathbf{Rot}_{\boldsymbol{z}}(q)\,\mathbf{V}^{-1}$ *with* $\mathbf{V} = \mathbf{R}(\boldsymbol{b})\,\mathbf{Rot}_{\boldsymbol{z}}(\beta)\,\mathbf{Tran}_{[x,y,z]}(\boldsymbol{l})$.

Proof. For an arbitrary *revolute* joint twist vector $\boldsymbol{\xi}$ there exist $\mathbf{V} \in \mathbb{SE}(3)$ so that

$$\exp(q\hat{\boldsymbol{\xi}}) = \mathbf{V}\,\exp(q\hat{\boldsymbol{\xi}}_{rev})\,\mathbf{V}^{-1} = \exp(q\mathbf{V}\,\hat{\boldsymbol{\xi}}_{rev}\,\mathbf{V}^{-1}), \tag{16}$$

where $\mathbf{V} = \mathbf{R}(\boldsymbol{b})\,\mathbf{Rot}_z(\beta)\,\mathbf{Tran}_{[x,y,z]}(\boldsymbol{l})$ is defined by (3), $\boldsymbol{\xi}_{rev} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^\top$ (see(9)). Equation (16) is equivalent to $\hat{\boldsymbol{\xi}} = \mathbf{V}\,\hat{\boldsymbol{\xi}}_{rev}\,\mathbf{V}^{-1}$ and, consequently, can be expressed in *adjoint form*, using (14) (15)

$$\boldsymbol{\xi} = Ad(\mathbf{V})\boldsymbol{\xi}_{rev} = \begin{bmatrix} \widetilde{\mathbf{V}} & \mathbf{0}_{3\times3} \\ \hat{\boldsymbol{t}}\,\widetilde{\mathbf{V}} & \widetilde{\mathbf{V}} \end{bmatrix}\boldsymbol{\xi}_{rev}, \tag{17}$$

where $\widetilde{\mathbf{V}} \in \mathbb{SO}(3)$ and $\boldsymbol{t} \in \mathbb{R}^3$ are parts of $\mathbf{V}$. One can obtain **POE-CPC** parameter relations substituting $\mathbf{V}$ in close form (3) (according to **CPC** convention) to (17), denoting $\cos(\beta)$, $\sin(\beta)$ as c_β, s_β, respectively

$$\begin{aligned} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} &= \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}, \\ \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} &= \begin{bmatrix} k_1 s_\beta + k_2 c_\beta & -k_2 s_\beta + k_1 c_\beta \\ -k_2 s_\beta - k_3 c_\beta & k_3 s_\beta - k_2 c_\beta \\ -k_4 s_\beta + k_5 c_\beta & -k_5 s_\beta - k_4 c_\beta \end{bmatrix} \begin{bmatrix} l_x \\ l_y \end{bmatrix} \end{aligned} \tag{18}$$

where $k_1 = \frac{b_y^2+b_z^2+b_z}{b_z+1}$, $k_2 = \frac{b_x b_y}{b_z+1}$, $k_3 = \frac{b_x^2+b_z^2+b_z}{b_z+1}$, $k_4 = b_x$, $k_5 = b_y$.

Surprisingly, the l_z component of $\boldsymbol{l}$ is a redundant parameter via *straightforward calculations*, regardless of CPC model assumptions [17].

Moreover, (18) can be simplified, taking into account parameter β redundancy ($\beta = 0$ for all frames, except $\{Base\}$ and $\{Tool\}$ as mentioned earlier),

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}, \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{b_x b_y}{b_z+1} & \frac{b_y^2+b_z^2+b_z}{b_z+1} \\ -\frac{b_x^2+b_z^2+b_z}{b_z+1} & -\frac{b_x b_y}{b_z+1} \\ b_y & -b_x \end{bmatrix}}_{=\mathbf{D}} \underbrace{\begin{bmatrix} l_x \\ l_y \end{bmatrix}}_{=\boldsymbol{l}}. \tag{19}$$

Let us denote rows of $\mathbf{D}$ as $\boldsymbol{d}_i$ with $i = \overline{1, 3}$. We assume first that $|\omega_3| = max\big(|\omega_1|, |\omega_2|, |\omega_3|\big)$ without loss of generality, that implies $\omega_3 \neq 0$ since $||\boldsymbol{\omega}|| = 1$. Then, form matrix $\mathbf{K} \in \mathbb{R}^{2\times2}$ with rows $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$. It can be easily checked that $\mathbf{K}$ is non-singular since $\det(\mathbf{K}) = b_z$ applying the assumption above and basic CPC model properties: $||\boldsymbol{b}|| = 1$ and $b_z \neq -1$ (see above the (3)). Therefore, l_x and l_y parameters can uniquely be found as $\boldsymbol{l} = \mathbf{K}^{-1}\begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$. It remains to demonstrate that $v_3 = \boldsymbol{d}_3^\top\boldsymbol{l}$ also satisfies (19) that can be done straight-forward using the POE property of revolute joints $\boldsymbol{\omega}^T\boldsymbol{v} = 0$ (see below(5)), that is $v_3 = -(\omega_1 v_1 + \omega_2 v_2)/\omega_3$.

The cases when $|\omega_1|$ or $|\omega_2|$ are $max\big(|\omega_1|, |\omega_2|, |\omega_3|\big)$ are proven in the same technique, forming non-singular $\mathbf{K}$ with $\boldsymbol{d}_2$, $\boldsymbol{d}_3$ and $\boldsymbol{d}_1$, $\boldsymbol{d}_3$, respectively, that is always possible since $\det(\mathbf{K})$ equals to b_x and $-b_y$ as well. Similarly, it is shown using $\boldsymbol{\omega}^T\boldsymbol{v} = 0$ that $v_1 = \boldsymbol{d}_1^\top\boldsymbol{l}$ and $v_2 = \boldsymbol{d}_2^\top\boldsymbol{l}$ is also a solution of (19).

Thus, the solution of (19) does exist, and conversion from POE to CPC parameters for *revolute* joint can be done. □

Lemma 2. *CPC parameters of prismatic joint transformation* $\mathbf{T} = \exp(q\hat{\boldsymbol{\xi}})$ *in POE form can be obtained using decomposition* $\mathbf{T} = \mathbf{V}\,\mathbf{Tran}_z(q)\,\mathbf{V}^{-1}$ *with* $\mathbf{V} = \mathbf{R}(\boldsymbol{b})\,\mathbf{Rot}_z(\beta)\,\mathbf{Tran}_{[x,y,z]}(\boldsymbol{l})$.

Proof. For a *prismatic* joint twist vector $\boldsymbol{\xi}$ there exist $\mathbf{V} \in \mathbb{SE}(3)$ such that

$$\exp(q\hat{\boldsymbol{\xi}}) = \mathbf{V}\,\exp(q\hat{\boldsymbol{\xi}}_{psm})\,\mathbf{V}^{-1} = \exp(q\mathbf{V}\,\hat{\boldsymbol{\xi}}_{psm}\,\mathbf{V}^{-1}), \tag{20}$$

where $\mathbf{V} = \mathbf{R}(\boldsymbol{b})\,\mathbf{Rot}_z(\beta)\,\mathbf{Tran}_{[x,y,z]}(\boldsymbol{l})$ (3) and $\boldsymbol{\xi}_{psm} = \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 1\end{bmatrix}^\top$ (10). Similar to Lemma 1 proof, (20) simplified to $\hat{\boldsymbol{\xi}} = \mathbf{V}\,\hat{\boldsymbol{\xi}}_{psm}\,\mathbf{V}^{-1}$ can be rewritten in adjoint form (15)

$$\boldsymbol{\xi} = Ad(\mathbf{V})\,\boldsymbol{\xi}_{psm}. \tag{21}$$

Then, explicit substitution of $\mathbf{V}$ as (3) to (21) yields

$$\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{bmatrix} = \begin{bmatrix}0\\ 0\\ 0\end{bmatrix}, \quad \begin{bmatrix}v_1\\ v_2\\ v_3\end{bmatrix} = \begin{bmatrix}b_x\\ b_y\\ b_z\end{bmatrix}. \tag{22}$$

Since $\boldsymbol{\omega} = \mathbf{0}$ and $||\boldsymbol{v}|| = 1$ hold for POE *prismatic* joint, $\boldsymbol{b}$ can be found uniquely and other parameters can be thought of as redundant: $\beta = 0$ and $\boldsymbol{l} = \mathbf{0}$. ☐

Lemma 3. *CPC parameters of helical joint transformation* $\mathbf{T} = \exp(q\hat{\boldsymbol{\xi}})$ *in POE form can be obtained using decomposition* $\mathbf{T} = \mathbf{V}\,\mathbf{Rot}_z(q)\,\mathbf{Tran}_z(hq)\,\mathbf{V}^{-1}$ *with* $\mathbf{V} = \mathbf{R}(\boldsymbol{b})\,\mathbf{Rot}_z(\beta)\,\mathbf{Tran}_{[x,y,z]}(\boldsymbol{l})$.

Proof. For arbitrary *helical* twist vector $\boldsymbol{\xi}$ there exist unique $\mathbf{V}$ so that:

$$\begin{aligned}\exp(q\hat{\boldsymbol{\xi}}) = \mathbf{V}\,\exp(q\hat{\boldsymbol{\xi}}_{hel})\,\mathbf{V}^{-1} &= \\ = \mathbf{V}\,\mathbf{Rot}_{\boldsymbol{z}}(q)\,\mathbf{Tran}_{\boldsymbol{z}}(hq)\,\mathbf{V}^{-1}&,\end{aligned} \tag{23}$$

where $\mathbf{V} = \mathbf{R}(\boldsymbol{b})\,\mathbf{Rot}_z(\beta)\,\mathbf{Tran}_{[x,y,z]}(\boldsymbol{l})$ from (3), $\boldsymbol{\xi}_{hel} = \begin{bmatrix}0 & 0 & 1 & 0 & 0 & h\end{bmatrix}^\top$ (11) and $h = \boldsymbol{\omega}^\top\boldsymbol{v}$. Transformation (23) can also be rewritten in linear form (14) using adjoint mapping (15) as before. Then, substituting $\mathbf{V}$ in close form (3) with redundant $\beta = 0$, as for (19), one can obtain

$$\begin{gathered}\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{bmatrix} = \begin{bmatrix}b_x\\ b_y\\ b_z\end{bmatrix}, \\ \begin{bmatrix}v_1\\ v_2\\ v_3\end{bmatrix} = \begin{bmatrix}\frac{b_x b_y}{b_z+1} & \frac{b_y^2+b_z^2+b_z}{b_z+1} & b_x\\ -\frac{b_x^2+b_z^2+b_z}{b_z+1} & -\frac{b_x b_y}{b_z+1} & b_y\\ b_y & -b_x & b_z\end{bmatrix}\begin{bmatrix}l_x\\ l_y\\ h\end{bmatrix}.\end{gathered} \tag{24}$$

It can be seen easily verified that due to $\boldsymbol{b} = \boldsymbol{\omega}$ coordinate change $\boldsymbol{u} = \boldsymbol{v} - h\boldsymbol{\omega}$ transforms (24) as

$$\boldsymbol{\omega} = \boldsymbol{b}, \boldsymbol{u} = \begin{bmatrix}\frac{b_x b_y}{b_z+1} & \frac{b_y^2+b_z^2+b_z}{b_z+1}\\ -\frac{b_x^2+b_z^2+b_z}{b_z+1} & -\frac{b_x b_y}{b_z+1}\\ b_y & -b_x\end{bmatrix}\begin{bmatrix}l_x\\ l_y\end{bmatrix}, \tag{25}$$

with $\boldsymbol{\omega}^T\boldsymbol{u} = 0$. Applying the methodology presented in Lemma 1 (see (19)) to the transformed system (25), the existence of a solution to (24) is established. Then, the conversion from POE to CPC parameters for *helical* joint can be done. ☐

The generalized version of the **Statement 1** was originally formulated and proven in [17] (Sec. V)[2], and is placed below to maintain the algorithm integrity.

Statement 1. *CPC parameters of arbitrary constant transformation* $\mathbf{T} = \exp(\hat{\boldsymbol{\xi}})$ *in POE form can be uniquely obtained from* $\mathbf{V} = \mathbf{R}(\boldsymbol{b})\,\mathbf{Rot}_{\boldsymbol{z}}(\beta)\,\mathbf{Tran}_{[x,y,z]}(\boldsymbol{l})$.

Proof. Let us first consider $\widetilde{\mathbf{R}}, \widetilde{\mathbf{T}} \in \mathbb{SO}(3)$ parts of $\mathbf{R}(\boldsymbol{b})$ (4) and given arbitrary twist $\mathbf{T} = \exp(\hat{\boldsymbol{\xi}}) = \begin{bmatrix}\widetilde{\mathbf{T}} & \boldsymbol{t}\\ \mathbf{0} & 1\end{bmatrix}$, respectively. Following the CPC model (3), one can write down an expression for rotational components

$$\widetilde{\mathbf{T}} = \widetilde{\mathbf{R}}(\boldsymbol{b})\,\widetilde{\mathbf{Rot}}_z(\beta), \tag{26}$$

where $\widetilde{\mathbf{Rot}}_z(\beta) \in \mathbb{SO}(3)$. Since the next transformation to $\mathbf{R}(\boldsymbol{b})$ is $\widetilde{\mathbf{Rot}}_z(\beta)$, $\boldsymbol{b}$ is uniquely defined as the third column of $\widetilde{\mathbf{T}}$: $\boldsymbol{b} = \begin{bmatrix}\tilde{t}_{31} & \tilde{t}_{32} & \tilde{t}_{33}\end{bmatrix}^\top$, based on CPC model properties (see [17] section V). Consequently, β is found from

$$\widetilde{\mathbf{Rot}}_z(\beta) = \widetilde{\mathbf{R}}^\top(\boldsymbol{b})\,\widetilde{\mathbf{T}}. \tag{27}$$

Finally, $\boldsymbol{l} = \begin{bmatrix}l_x & l_y & l_z\end{bmatrix}^\top$ can be calculated as follows

$$\boldsymbol{l} = \left(\widetilde{\mathbf{R}}(\boldsymbol{b})\,\widetilde{\mathbf{Rot}}_z(\beta)\right)^\top \boldsymbol{t}, \tag{28}$$

where $\boldsymbol{t}$ is displacement vector $\mathbb{R}^3$ of $\mathbf{T}$. ☐

Thus, we are ready to formulate the algorithm and corresponding theorem for **POE–CPC** conversion, that similar in structure to **POE–DH** conversion case [25].

Theorem 1. *Algorithm 1 enables POE-CPC conversion given any set of admissible POE parameters.*

Proof. Based on (12) manipulator forward kinematics in POE-form can be formulated as

$$\mathbf{T}_0^{Tool} = \exp(q_1\hat{\boldsymbol{\xi}}_1)\cdot\ldots\cdot\exp(q_n\hat{\boldsymbol{\xi}}_n)\,\exp(\hat{\boldsymbol{\xi}}_{Tool}), \tag{29}$$

Then, depending on joint type, we apply **Lemmas 1–3** to $\exp(q_1\hat{\boldsymbol{\xi}}_1)$

$$\begin{aligned}\mathbf{T}_0^{Tool} = \mathbf{V}_0\exp(q_1\hat{\boldsymbol{\xi}}_1')\,\mathbf{V}_0^{-1}\,\exp(q_2\hat{\boldsymbol{\xi}}_2)\cdot\ldots&\cdot\\ \cdot\exp(q_n\hat{\boldsymbol{\xi}}_n)\,\exp(\hat{\boldsymbol{\xi}}_{Tool})&,\end{aligned} \tag{30}$$

where $\mathbf{V}_0 = \mathbf{R}(\boldsymbol{b}_0)\,\mathbf{Rot}_z(\beta_0)\,\mathbf{Tran}_{[x,y,z]}(\boldsymbol{l}_0)$ (with $\beta_0 = 0$) is transformation defined by converted CPC parameters, $\boldsymbol{\xi}_1'$ is unit *revolute* or *prismatic*, or *helical* twist vector — (9), (10) or (11), respectively. Rearranging (30) using well-known

---

[2] The algorithm cannot be applied when the matrix exponential depends on a joint variable, as in $T = \exp(q\hat{\xi})$. However, if the transformation is constant, i.e., $T = \exp(\hat{\xi})$, the algorithm [17] becomes applicable.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Algorithm 1 POE-CPC parameter conversion algorithm

```
Require: POE parameters $\boldsymbol{\xi}_i$, $\boldsymbol{\xi}_{Tool}$ with $i = \overline{1,n}$ that define
    forward kinematics as
        $\mathbf{T}_0^{Tool} = \exp(q_1\hat{\boldsymbol{\xi}}_1) \cdot \ldots \cdot \exp(q_n\hat{\boldsymbol{\xi}}_n)\,\exp(\hat{\boldsymbol{\xi}}_{Tool})$.
for $i = \overline{1,n}$ with $\mathbf{V}_{-1} = \mathbb{I}$ do
    Rearrange $\big(\mathbf{V}_{-1} \cdot \ldots \cdot \mathbf{V}_{i-2}\big)^{-1}\exp(q_i\hat{\boldsymbol{\xi}}_i)$ as
        $\exp\Big(q_i\big(\mathbf{V}_{-1} \cdot \ldots \cdot \mathbf{V}_{i-2}\big)^{-1}\hat{\boldsymbol{\xi}}_i\,\big(\mathbf{V}_{-1} \cdot \ldots \cdot \mathbf{V}_{i-2}\big)\Big)\cdot\big(\mathbf{V}_{-1} \cdot \ldots \cdot \mathbf{V}_{i-2}\big)^{-1}$;
    Denote $\boldsymbol{\xi}'_i = Ad\Big(\big(\mathbf{V}_{-1} \cdot \ldots \cdot \mathbf{V}_{i-2}\big)^{-1}\Big)\boldsymbol{\xi}_i$ using (13), (14) where $\boldsymbol{\xi}'_i = [\boldsymbol{\omega}'^{\top}_i\ \boldsymbol{v}'^{\top}_i]^{\top}$;
    if $\boldsymbol{\xi}'_i$ is revolute: $||\boldsymbol{\omega}'_i|| = 1$ and $\boldsymbol{\omega}'^{\top}_i\boldsymbol{v}'_i = 0$ then
        Apply Lemma 1 to $\exp(q_i\hat{\boldsymbol{\xi}}'_i)$
            obtain $\mathbf{V}_{i-1}$ and $\mathbf{Q}_i = \exp(q_i\hat{\boldsymbol{\xi}}_{rev})$;
            extract $\boldsymbol{b}_{i-1}$, $\boldsymbol{l}_{i-1}$, $\beta_{i-1} = 0$ from $\mathbf{V}_{i-1}$;
    else if $\boldsymbol{\xi}'_i$ is prismatic: $||\boldsymbol{\omega}'_i|| = 0$ then
        Apply Lemma 2 to
            obtain $\mathbf{V}_{i-1}$ and $\mathbf{Q}_i = \exp(q_i\hat{\boldsymbol{\xi}}_{psm})$;
            extract $\boldsymbol{b}_{i-1}$, $\boldsymbol{l}_{i-1} = \mathbf{0}$, $\beta_{i-1} = 0$ from $\mathbf{V}_{i-1}$;
    else if $\boldsymbol{\xi}'_i$ joint is helical: $||\boldsymbol{\omega}'_i|| = 1$ and $\boldsymbol{\omega}'^{\top}_i\boldsymbol{v}'_i = h$ then
        Apply Lemma 3 to
            obtain $\mathbf{V}_{i-1}$ and $\mathbf{Q}_i = \exp(q_i\hat{\boldsymbol{\xi}}_{hel})$;
            extract $\boldsymbol{b}_{i-1}$, $\boldsymbol{l}_{i-1}$, $\beta_{i-1} = 0$ from $\mathbf{V}_{i-1}$;
    end if
end for
Compute $\mathbf{V}_n$ and CPC parameters $\boldsymbol{b}_{i-1}$, $\boldsymbol{l}_{i-1}$, $\beta_{i-1} = 0$
    applying Statement 1 to $\big(\mathbf{V}_{-1}\cdot\ldots\cdot\mathbf{V}_{n-1}\big)^{-1}\exp(\hat{\boldsymbol{\xi}}_{Tool})$;
return CPC parameters $\boldsymbol{b}_i$, $\boldsymbol{l}_i$, $\beta_i$ for $i = \overline{0,n}$.
```

property $\mathbf{N}\exp(\mathbf{M})\mathbf{N}^{-1} = \exp(\mathbf{N}\mathbf{M}\mathbf{N}^{-1})$ and denoting $\mathbf{Q}_i := \exp(q_i\hat{\boldsymbol{\xi}}'_i)$ for $i = \overline{1,n}$, we obtain

$$\mathbf{T}_0^{Tool} = \mathbf{V}_0\,\mathbf{Q}_1\,\exp(q_2\mathbf{V}_0^{-1}\hat{\boldsymbol{\xi}}_2\mathbf{V}_0)\,\mathbf{V}_0^{-1}\cdot\ldots\cdot \\ \cdot\exp(q_n\hat{\boldsymbol{\xi}}_n)\,\exp(\hat{\boldsymbol{\xi}}_{Tool}). \quad (31)$$

Simplifying $\mathbf{V}_0^{-1}\hat{\boldsymbol{\xi}}_2\mathbf{V}_0$ with adjoint operator (14), (13) define $\boldsymbol{\xi}'_2 = Ad(\mathbf{V}_0^{-1})\boldsymbol{\xi}_2$, then, (31) is transformed to

$$\mathbf{T}_0^{Tool} = \mathbf{V}_0\,\mathbf{Q}_1\,\exp(q_2\hat{\boldsymbol{\xi}}'_2)\,\mathbf{V}_0^{-1}\cdot\ldots\cdot \\ \cdot\exp(q_n\hat{\boldsymbol{\xi}}_n)\,\exp(\hat{\boldsymbol{\xi}}_{Tool}). \quad (32)$$

Steps (30)-(32) are applied recursively for $i = \overline{2,n}$ with $\boldsymbol{\xi}'_i = Ad\Big(\big(\mathbf{V}_0\cdot\ldots\cdot\mathbf{V}_{i-2}\big)^{-1}\Big)\boldsymbol{\xi}_i$, yielding the final transformation

$$\mathbf{T}_0^{Tool} = \mathbf{V}_0\,\mathbf{Q}_1\,\mathbf{V}_1\mathbf{Q}_2\cdot\ldots\cdot \\ \cdot\mathbf{V}_{n-1}\,\mathbf{Q}_n\,\big(\mathbf{V}_0\cdot\ldots\cdot\mathbf{V}_{n-1}\big)^{-1}\exp(\hat{\boldsymbol{\xi}}_{Tool}). \quad (33)$$

Applying Statement 1 to the arbitrary twist $\mathbf{V}_{n-1}^{-1}\exp(\hat{\boldsymbol{\xi}}_{\text{Tool}})$ from (33) (which is independent of q_i), we derive $\mathbf{V}_n$ and its CPC parameters $\boldsymbol{b}_n$, $\boldsymbol{l}_n$, β_n. This yields the forward kinematics in CPC form:

$$\mathbf{T}_0^{\text{Tool}} = \mathbf{V}_0\mathbf{Q}_1\,\mathbf{V}_1\mathbf{Q}_2\cdot\ldots\cdot\mathbf{Q}_{n-1}\mathbf{V}_{n-1}\,\mathbf{Q}_n\mathbf{V}_n, \quad (34)$$

where each $\mathbf{T}_i = \mathbf{Q}_i\mathbf{V}_i$ is the homogeneous transformation (1) for frame i, composed of the motion matrix $\mathbf{Q}_i$ and shape matrix $\mathbf{V}_i$ (3). ☐

B. CPC–POE conversion

The reverse transition from the CPC to the POE formulation can be achieved with minor modifications to Theorem 1 and the approach presented in [23], [24], [30].

Starting from the forward kinematics in CPC form (1) given by

$$\mathbf{T}_0^{\text{Tool}} = \mathbf{V}_0\mathbf{Q}_1\,\mathbf{V}_1\mathbf{Q}_2\cdot\ldots\cdot\mathbf{Q}_{n-1}\mathbf{V}_{n-1}\,\mathbf{Q}_n\mathbf{V}_n, \quad (35)$$

where $\mathbf{Q}_i$, defined in (38), depends on the joint type (revolute, prismatic or helical) for $i = \overline{1,n}$. Each $\mathbf{Q}_i$ can be expressed as $\exp(q_i\hat{\boldsymbol{\xi}}_i)$, with $\boldsymbol{\xi}_i$ defined by (9), (10) or (11). Substituting this into (35) yields

$$\mathbf{T}_0^{\text{Tool}} = \mathbf{V}_0\exp(q_1\hat{\boldsymbol{\xi}}_1)\,\mathbf{V}_1\exp(q_2\hat{\boldsymbol{\xi}}_2)\cdot\ldots\cdot \\ \cdot\exp(q_{n-1}\hat{\boldsymbol{\xi}}_{n-1})\mathbf{V}_{n-1}\,\exp(q_n\hat{\boldsymbol{\xi}}_n)\mathbf{V}_n. \quad (36)$$

By introducing the transformed twist $\hat{\boldsymbol{\xi}}'_i = \big(\mathbf{V}_0\cdot\ldots\cdot\mathbf{V}_{i-1}\big)\,\hat{\boldsymbol{\xi}}_i\,\big(\mathbf{V}_0\cdot\ldots\cdot\mathbf{V}_{i-1}\big)^{-1}$, or equivalently in adjoint form $\boldsymbol{\xi}'_i = Ad(\mathbf{V}_0\cdot\ldots\cdot\mathbf{V}_{i-1})\boldsymbol{\xi}_i$, and utilizing the identity $\mathbf{N}\exp(\mathbf{M})\mathbf{N}^{-1} = \exp(\mathbf{N}\mathbf{M}\mathbf{N}^{-1})$ holding for any $\mathbf{N}$, $\mathbf{M} \in \mathbb{SE}(3)$, we can rewrite (36) as:

$$\mathbf{T}_0^{\text{Tool}} = \exp(q_1\hat{\boldsymbol{\xi}}'_1)\exp(q_2\hat{\boldsymbol{\xi}}'_2)\cdot\ldots\cdot \\ \cdot\exp(q_{n-1}\hat{\boldsymbol{\xi}}'_{n-1})\,\exp(q_n\hat{\boldsymbol{\xi}}'_n)\exp(\hat{\boldsymbol{\xi}}'_{Tool}), \quad (37)$$

where $\exp(\hat{\boldsymbol{\xi}}'_{Tool}) = \mathbf{V}_0\cdot\ldots\cdot\mathbf{V}_n$.

IV. PARAMETER CONVERSION EXAMPLE

We validate the proposed algorithm through numerical experiments using the PUMA 560 manipulator model as a benchmark test case. This choice ensures the comparability across the studies since the nominal and actual PUMA 560 parameters in POE-form, provided by [30], have been utilized in numerous papers, for instance, [26], [31], [32]. Actual POE parameters $\boldsymbol{\xi}_i$ and $\{Tool\}$ frame parameters $\boldsymbol{\xi}_{tool}$ of the PUMA 560 with helix pitches $h_i = \boldsymbol{\omega}_i^{\top}\boldsymbol{v}_i$ for $i = \overline{1,6}$ are shown below in Table I.

Among the six components of $\boldsymbol{h}$ analyzed, four exhibit a screw/helical structure. It should be emphasised that CPC model, as well as DH, was initially designed for prismatic and revolute joints only, allowing to decompose the $\mathbf{T}$ (1) into the joint movement $\mathbf{Q}$ and the link geometry $\mathbf{V}$ transformations $\mathbf{T} = \mathbf{Q}\,\mathbf{H}$ [17]. Then, one can modify the *motion matrix* as $\mathbf{Q} = \mathbf{Rot}_{\boldsymbol{z}}(q)\,\mathbf{Tran}_{\boldsymbol{z}}(q\,h)$ using helix pitch

TABLE I
ACTUAL POE PARAMETERS OF PUMA 560 MANIPULATOR [30]

	$\boldsymbol{\omega}$			$\boldsymbol{v}$			h
$\boldsymbol{\xi}_1$	0.04	-0.02	0.999	0.02	0.04	0	0
$\boldsymbol{\xi}_2$	0	-1.000	0	-0.02	0	0.05	0
$\boldsymbol{\xi}_3$	0.178	-0.984	-0.001	-0.07	0.009	-101	0.080
$\boldsymbol{\xi}_4$	0.062	0.013	-0.998	-51	249	0.075	-4.96e-5
$\boldsymbol{\xi}_5$	0.001	-1.000	0	-20.6	-0.021	-249	8.24e-7
$\boldsymbol{\xi}_6$	0.095	0.031	-0.995	-51	249	0	-2.874
$\boldsymbol{\xi}_{tl}$	0.02	-0.01	0.01	249	51	-20.6	—

h. Consequently, the methodology proposed in [31] for DH convention

$$\mathbf{Q} = \mathbf{Rot}_{\boldsymbol{z}}(j\,q)\,\mathbf{Tran}_{\boldsymbol{z}}(q\,h), \tag{38}$$

$j = 1$ and $h = 0$ for revolute, $j = 0$ and $h = 1$ for prismatic, and $j = 1$ and $h \neq 0$ for helical joints can be applied to CPC model as well. Thus, during the validation phase, we will incorporate the analysis of helix pitches by applying (38).

Then, we apply the **Theorem 1** result to obtain CPC parameters of Puma 560. While converting parameters from POE (30)-(32), it is necessary to normalize a twist vector $\boldsymbol{\xi}$, as was stated in [31]: $\boldsymbol{\xi} = \lambda\bar{\boldsymbol{\xi}}$, where $\boldsymbol{\xi} = \begin{bmatrix}\boldsymbol{\omega}^\top & \boldsymbol{v}^\top\end{bmatrix}^\top$, so that $||\boldsymbol{\omega}|| = \lambda$ for *revolute* and *helical* twists, $||\boldsymbol{v}|| = \lambda$ for *prismatic* twist, and $\bar{\boldsymbol{\xi}}$ is normalized twist vector.

The assessment of parameter conversion algorithm accuracy was conducted by comparing the $\{Tool\}$ frame position and orientation in each configuration for POE and CPC. Specifically, 1000 test points were randomly generated in $\mathcal{Q}$-space by sampling each $\boldsymbol{q}_i \in \mathbb{R}^6$ from a uniform distribution: $\boldsymbol{q}_i \sim \mathcal{U}_{[-\pi,\pi]^6} \in \mathbb{R}^6$ with $i = \overline{1,1000}$. Then, denote the forward kinematics for converted CPC and actual POE parameters (Table I) as $\mathbf{T}^i_{conv}$ and $\mathbf{T}^i_{act}$. The position and orientation errors, e^i_{pos} and e^i_{or}, are computed as follows

$$\begin{aligned} e^i_{pos} &= \left|\left|\boldsymbol{t}^i_{act} - \boldsymbol{t}^i_{conv}\right|\right|, \\ e^i_{or} &= \left|\left|\mathrm{EUL}\left(\widetilde{\mathbf{T}}^{i\top}_{act}\,\widetilde{\mathbf{T}}^i_{conv}\right)\right|\right|, \end{aligned} \tag{39}$$

where $\boldsymbol{t}_{act}$, $\boldsymbol{t}_{conv} \in \mathbb{R}^3$ and $\widetilde{\mathbf{T}}_{act}$, $\widetilde{\mathbf{T}}_{conv} \in \mathbb{SO}(3)$ are parts of $\mathbf{T}_{act}$ and $\mathbf{T}_{conv}$, respectively, $\mathrm{EUL}(\cdot) : \mathbb{SO}(3) \mapsto \mathbb{R}^3$ are the Euler angles (**ZYX**–configuration) extracted from rotation matrix. The corresponding conversion errors are shown in Table II.

TABLE II
COMPARISON OF **POE–DH-CPC** AND **POE–CPC** CONVERSION FRAMEWORKS: POSITION AND ORIENTATION ERRORS FOR THE PUMA 560 ROBOT ACROSS 1000 CONFIGURATIONS

		POE-DH-CPC[a]	**POE-CPC**
Mean	pos., 10^{-16} m	4.3985	2.2022
	or., 10^{-15} rad	2.3828	2.4062
Max	pos., 10^{-16} m	10.5545	7.3824
	or., 10^{-15} rad	4.4111	3.9381

[a] The POE-DH conversion was done with the framework presented in [31], then the DH-CPC conversion was done implementing results in [17].

Here, the alternative approach for converting **POE** to **CPC** was evaluated on the same configuration set using a sequential transformation pipeline: **POE–DH** [31] (implemented via the open-source software https://github.com/drliaowu/GeneralPOE2DH) followed by **DH–CPC** [17], in accordance with the conversion scheme shown in Fig. 1.

It can be observed that the conversion errors obtained using the proposed direct **POE–CPC** method are consistently smaller than those produced by the sequential **POE–DH–CPC** approach, both remaining on the same numerical order as the default *MATLAB* round-off error (approximately 1×10^{-16}). Nevertheless, the principal advantage of the *proposed* method lies beyond mere numerical precision. First, the resulting **CPC** parameters are strictly non-redundant ($l_z = \beta = 0$ for all frames except $\{Tool\}$), which is crucial for calibration and parameter identifiability. Second, the **CPC** model obtained through the **POE–DH–CPC** transformation **inevitably inherits** the well-known ill-conditioning of the **DH** parameters — one of the key motivations for the development of alternative kinematic representations [12], [13]. The affected parameters are highlighted in red in Table IV.

TABLE III
CPC PARAMETERS OF THE PUMA 560 MANIPULATOR CONVERTED USING THE PROPOSED **POE–CPC** ALGORITHM

$\boldsymbol{b}$			$\boldsymbol{l}$			β
0.040	-0.020	0.999	-0.040	0.020	0	0
-0.000	-1.000	0.020	-0.009	-0.022	0	0
0.178	0.006	0.984	100.972	3.946	0	0
0.103	-0.995	-0.001	153.092	9.939	0	0
-0.277	0.961	-0.013	6.593	-29.523	0	0
-0.123	0.992	-0.031	-0.725	-0.175	0	0
-0.098	-0.063	0.993	0.668	1.575	1.653	0.132

TABLE IV
CPC PARAMETERS OF THE PUMA 560 MANIPULATOR CONVERTED BY SEQUENTIAL APPLICATION OF THE **POE–DH** [31] AND **DH–CPC** [17] ALGORITHMS

$\boldsymbol{b}$			$\boldsymbol{l}$			β
0.040	-0.020	0.999	0.000	0.045	0.999	1.107
-0.894	-0.447	0.020	0.009	1.021	-0.020	2.034
-0.178	-0.006	0.984	0.511	-101.047	-558.589	-1.536
-0.998	-0.069	-0.001	153.304	552.752	-0.419	-1.502
-0.177	0.984	-0.013	1.837	30.190	-0.391	-2.964
-0.033	0.999	-0.031	3.176	-0.792	0.025	-3.108
0.085	-0.011	-0.996	-3.245	0.338	1.488	2.897

A. *Expected Application of the Proposed Scheme*

In a practical calibration workflow, the proposed conversion framework can be applied as follows. First, the nominal manipulator model is described using any standard convention (such as **DH**, **CPC**, or **POE**). This model can then be transformed into other representations through the conversion pathways illustrated in Fig. 1 — for example, **DH–POE** using [23]–[25]; **DH–CPC** using [17], [18]; **POE–DH** using [25], [31]; **POE–CPC** using the *proposed* framework; **CPC–DH** using [17], [18]; and **CPC–POE** based on the results of [23]–[25] modified in a *proposed* framework. After defining the model, a set of calibration configurations is collected, and parameter identification is performed independently for each representation (**DH**, **CPC**, and **POE**). The identified models are then validated on a test dataset to determine the most accurate one. Finally, the best-performing calibrated model is converted back into the convention required for the target application, again following the pathways in Fig. 1.

V. CONCLUSION

The paper has presented an analytical framework for parameter conversion between CPC and POE kinematic models of serial-chain mechanisms. Using Lie group theory, we established three lemmas that enable exact and consistent transformations between these representations. The proposed approach complements existing conversion schemes (e.g., **DH–CPC**, **DH–POE**), completing the comprehensive interoperability algorithm of the kinematic model.

The solution's significance is threefold: it bridges the remaining gap in kinematic model conversions, enabling full interoperability between DH, CPC, and POE conventions; the framework facilitates direct comparison of calibration and identification results across different conventions; users can now select the most convenient convention for a specific problem and seamlessly convert results to any required model.

Validated numerically on the PUMA 560 robot, this framework (MATLAB code provided https://github.com/oys5/POE_CPC_conversion with additional 4 DoF RRRP SCARA and 7 DoF Kuka LWR4+ robots conversion examples) not only preserves accuracy but also ensures backward compatibility with DH- and POE-based algorithms, avoids parametric discontinuities, and simplifies CPC parameter retrieval for any serial-link mechanism.

REFERENCES

[1] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *Journal of Applied Mechanics*, vol. 22, no. 2, pp. 215–221, 1955.

[2] A. Joubair and I. Bonev, "Kinematic calibration of a six-axis serial robot using distance and sphere constraints," *The International Journal of Advanced Manufacturing Technology*, vol. 77, 03 2014.

[3] K. Kamali, A. Joubair, I. Bonev, and P. Bigras, "Non-kinematic calibration of an industrial robot under multidirectional external loads," in *IEEE International Conference on Robotics and Automation*, 2016.

[4] S. Kolyubin, A. Shiriaev, and A. Jubien, "Consistent kinematics and dynamics calibration of lightweight redundant industrial manipulators," *The International Journal of Advanced Manufacturing Technology*, vol. 101, p. 243–259, 2019.

[5] L. Žlajpah and T. Petrič, "Kinematic calibration for collaborative robots on a mobile platform using motion capture system," *Robotics and Computer-Integrated Manufacturing*, vol. 79, p. 102446, 2023.

[6] H. Zhen, "Error analysis of robot manipulators and error transmission functions," *Proc. of the 15th ISIR, Tokyo*, pp. 873–878, 1985.

[7] D. Payannet, M. Aldon, and A. Liegeois, "Identification and compensation of mechanical errors for industrial robots," in *Proc. 15th Int. Symp. on Industrial Robots*, 1985, pp. 857–864.

[8] R. Ibarra and N. D. Perreira, "Determination of linkage parameter and pair variable errors in open chain kinematic linkages using a minimal set of pose measurement data," *Journal of Mechanisms, Transmissions, and Automation in Design*, vol. 108, no. 2, pp. 159–166, 06 1986.

[9] C.-H. Wu, "The kinematic error model for the design of robot manipulator," *American Control Conference*, pp. 497–502, 1983.

[10] B. Mooring, "The effect of joint axis misalignment on robot positioning accuracy," in *Proceedings of 1983 ASME Computers in Engineering Conference*, August 1983, p. 151–155.

[11] L. Everett, M. Driels, and B. Mooring, "Kinematic modelling for robot calibration," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 4, 1987, pp. 183–189.

[12] B. Mooring, M. Driels, and Z. Roth, *Fundamentals of Manipulator Calibration*. USA: John Wiley & Sons, Inc., 1991.

[13] S. A. Hayati, "Robot arm geometric link parameter estimation," in *The 22nd IEEE Conference on Decision and Control*, 1983, pp. 1477–1483.

[14] H. Stone, A. Sanderson, and C. Neuman, "Arm signature identification," in *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, vol. 3, 1986, pp. 41–48.

[15] W. Veitschegger and C.-H. Wu, "Robot accuracy analysis based on kinematics," *IEEE Journal on Robotics and Automation*, vol. 2, no. 3, pp. 171–179, 1986.

[16] B. Mooring and G. Tang, "An improved method for identifying the kinematic parameters in a six axis robot," in *International Computers in Engineering Conference and Exhibit*, 1984, p. 79–84.

[17] H. Zhuang, Z. Roth, and F. Hamano, "A complete and parametrically continuous kinematic model for robot manipulators," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1, 1990, pp. 92–97.

[18] H. Zhuang, "Kinematic modeling, identification and compensation of robot manipulators," Ph.D. dissertation, USA, 1989.

[19] K. Roberts, "A new representation for a line," in *Proceedings CVPR '88: The Computer Society Conference on Computer Vision and Pattern Recognition*, 1988, pp. 635–640.

[20] R. Brockett, "Robotic manipulators and the product of exponentials formula," in *Mathematical Theory of Networks and Systems*, P. Fuhrmann, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1984, pp. 120–129.

[21] X. Yang, L. Wu, J. Li, and K. Chen, "A minimal kinematic model for serial robot calibration using poe formula," *Robotics and Computer-Integrated Manufacturing*, vol. 30, no. 3, pp. 326–334, 2014.

[22] F. Park, "Computational aspects of the product-of-exponentials formula for robot kinematics," *IEEE Transactions on Automatic Control*, vol. 39, no. 3, pp. 643–647, 1994.

[23] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*, 1st ed. USA: CRC Press, Inc., 1994.

[24] K. M. Lynch and F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*, 1st ed. USA: Cambridge University Press, 2017.

[25] L. Wu, R. Crawford, and J. Roberts, "An analytic approach to converting poe parameters into d–h parameters for serial-link robots," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2174–2179, 2017.

[26] R. He, Y. Zhao, S. Yang, and S. Yang, "Kinematic-parameter identification for serial-robot calibration based on poe formula," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 411–423, 2010.

[27] T. Sun, B. Lian, S. Yang, and Y. Song, "Kinematic calibration of serial and parallel robots based on finite and instantaneous screw theory," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 816–834, 2020.

[28] J. Jiang, X. Luo, Q. Luo, L. Qiao, and M. Li, "An overview of hand-eye calibration," *The International Journal of Advanced Manufacturing Technology*, vol. 119, no. 1, pp. 77–97, 2022.

[29] W. Wang, Q. Guo, Z. Yang, Y. Jiang, and J. Xu, "A state-of-the-art review on robotic milling of complex parts with high efficiency and precision," *Robotics and Computer-Integrated Manufacturing*, vol. 79, p. 102436, 2023.

[30] K. Okamura and F. C. Park, "Kinematic calibration using the product of exponentials formula," *Robotica*, vol. 14, pp. 415 – 421, 1996.

[31] L. Wu, R. Crawford, and J. Roberts, "Geometric interpretation of the general poe model for a serial-link robot via conversion into d-h parameterization," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 7360–7366.

[32] N. V. Toan and P. B. Khoi, "A svd-least-square algorithm for manipulator kinematic calibration based on the product of exponentials formula," *Journal of Mechanical Science and Technology*, vol. 32, pp. 5401 – 5409, 2018.

[33] P. Yang, K. Luo, and R. Wu, "A calibration method for spatial error of industrial robot under variable load conditions based on double-ball rotary structure and three contact displacement sensors," *Precision Engineering*, vol. 85, pp. 291–303, 2024.

[34] O. Y. Sumenkov, D. D. Kulminskii, and S. V. Gusev, "Kinematic calibration of an industrial manipulator without external measurement devices," *Russian Journal of Nonlinear Dynamics*, vol. 20, no. 5, pp. 979–1001, 2024.

[35] P. L. Broderick and R. J. Cipra, "A method for determining and correcting robot position and orientation errors due to manufacturing," *Journal of Mechanisms, Transmissions, and Automation in Design*, vol. 110, no. 1, pp. 3–10, 03 1988.

[36] F. Warner, *Foundations of Differentiable Manifolds and Lie Groups*, ser. Graduate Texts in Mathematics. Springer, 1983.