Representing Uncertain Spatial Transformations in Robotic Applications in a Structured Framework Leveraging Lie Algebra

Marco Sewtz *, Lukas Burkhard, Xiaozhou Luo, Leon Dorscht, and Rudolph Triebel

Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany
Email: Marco.Sewtz@dlr.de (M.S.); Lukas.Burkhard@dlr.de (L.B.); Xiaozhou.Luo@dlr.de (X.L.); Leon.Dorscht@dlr.de (L.D.); Rudolph.Triebel@dlr.de (R.T.)
*Corresponding author

***Abstract*—Accurately representing spatial transformations in robotics is crucial for reliable system performance. Traditional methods often fail to account for internal inaccuracies and environmental factors, leading to significant errors. This work introduces a framework that incorporates uncertainty into transformation trees using Lie Algebra, offering a consistent and realistic computation of spatial transformations. Our approach models inaccuracies from sensor decalibration, joint position errors, mechanical stress, and gravitational influences, as well as environmental uncertainties from perception limitations. By integrating probabilistic models into transformation calculations, we provide a robust and adaptable solution for various robotic applications. The framework is implemented using a C++ library with a Python wrapper, leveraging hierarchical transformation trees to simplify kinematic chains and apply uncertainty propagation. Real-world examples demonstrate the framework's effectiveness: compensating for gravitational bending in a robotic arm and handling uncertainties in a mapping task with an uncertain kinematic. These applications highlight the framework's ability to enhance the accuracy and reliability of tasks such as manipulation, navigation, and interaction with environments. This contribution aims to advance robotic systems' performance by providing a comprehensive method for managing spatial transformation uncertainties.**

***Keywords*—robotics, transformation tree, uncertainty modeling, Lie Algebra**

I. Introduction

In the dynamic landscape of robotics, accurately representing spatial transformations is pivotal for reliable system performance. Conventional methods, which treat provided transformations as precise and deterministic, face difficulties with inherent inaccuracies within the system and environmental complexities. This work highlights the need for spatial representations in robotics that account for inaccuracies, often referred to as scene graphs. These representations allow modeling not only the spatial relationships in a robot-environment system but also the gaps in our knowledge about it.

An illustrative example is the distinction between a robotic arm's repetition accuracy, which signifies its capability to consistently reach the same point in a workspace, and its absolute accuracy. For conventional robotic systems, the first can be assumed to be "exact". However, the error of the latter can be higher by several orders of magnitude which motivates the modeling of this error. Position measurements, constrained by both physical limitations and environmental influences, frequently fall short of the requisite precision. This constraint becomes especially critical in applications requiring high accuracy, such as surgical robotics [1].

An additional example is the process of registering a robot with respect to its environment, a task achieved through either an inaugural calibration procedure [2] or by means of the navigation implemented in mobile robotic systems [3].

Interestingly, various scholarly works [4, 5] have considered robot uncertainty within specific domains, such as the kinematic structure or autonomous navigation components. However, there is limited progress in combining these several domains into one single representation like a scene graph to achieve a unified consideration of inaccuracy-aware spatial relations. Conventional approaches that disregard uncertainty in scene graphs fall short in capturing the intricacies of real-world scenarios.

This paper advocates for a paradigm shift by introducing a framework that incorporates uncertainty into scene graphs, offering a more realistic and robust representation of transformations. By addressing challenges posed by both robot internal inaccuracies and the uncertainty of the robot's interaction with the environment, our approach aims to enhance the reliability and performance of robotic systems in practical applications.

We use the following terminology in this paper: Robotic systems can be subject to errors that cause *inaccurate* pose

Manuscript received March 22, 2024; revised May 28, 2024; accepted July 30, 2024; published January 9, 2025.

doi: 10.18178/ijmerr.14.1.1-9

calculations, either within the system or with respect to its environment. A common simplification is to model such inaccuracies in a probabilistic way, thus subjecting nominal relative poses to an additional *uncertainty*. For a multitude of robotic applications, such uncertainty is modeled as a *zero-mean normal distribution*, thus an uncertain pose consists of a nominal pose and a covariance matrix. Generally, this simplification trades the exact representation of robotic errors for the availability of powerful mathematical tools and is thus well established in the robotic community. We adopt this error modeling as well, which allows us to immediately integrate the probabilistic pose information from other software components into our scene graph.

II. Related Work

Accurately describing the spatial relationships of a robot and its environment is a key aspect of robotics specifically and mechanical mechanisms generally. Accurately describing the spatial relationships between a robot and its environment is crucial in both robotics and mechanical engineering. This involves not only understanding the robot's position and orientation within its workspace but also how it interacts with various objects and obstacles around it. The ability to model and predict these interactions is crucial for tasks such as navigation, manipulation, and automated decision-making. Furthermore, a precise understanding of spatial relationships enhances the robot's efficiency, safety, and adaptability in complex and dynamic environments. Consequently, advancements in this area have significant implications for the development of more sophisticated and capable robotic systems.

Commencing with the early explorations in formulating a framework for kinematics in mechanical structures [6, 7], the field witnessed significant strides with one of the pivotal works by Denavit and Hartenberg [8]. In this groundbreaking contribution, the authors devised a structured yet elegant methodology to comprehensively describe the chain of transformations associated with robotic arms. Subsequent endeavors augmented the toolbox of robot kinematics representation, for example by considering the underlying Lie-Algebra of spatial transformations [9]. Advancements in the use of conformal geometric algebra have provided a unified approach to geometric reasoning, simplifying the computation of kinematics and dynamics of serial manipulators [10]. Moreover, neural network-based approaches and deep reinforcement learning have enhanced the precision and efficiency of solving inverse kinematics problems for high degrees of freedom manipulators [11, 12]. Our recent work [13] provides a kinematic robot description that allows considering inaccuracies from joint position measurements, mechanical stress-induced deformations, and gravitational influences in a probabilistic manner.

In the field of robotic navigation, numerous approaches account for the uncertainty of relative transformations, particularly in the domain of Simultaneous Localization and Mapping (SLAM). For instance, methods such as those proposed by Kaess *et al.* in iSAM2 [14] and Kümmerle *et al.* in g2o [15] utilize the covariance or information matrix to appropriately weigh different spatial transformations within a graph optimization framework. Recent advancements include the development of distributed pose graph optimization, which enhances collaborative SLAM by efficiently managing local and global uncertainties [16], and the integration of multi-level graph partitioning to improve scalability and accuracy [17]. These techniques enhance the accuracy and reliability of mapping and localization by effectively managing the inherent uncertainties in sensor measurements and environmental interactions.

The interaction of a robot with objects in its environment, specifically the uncertainties inherent in the workspace, has been investigated in Ref. [18]. Additionally, significant progress has been made in modeling the uncertainty in the perception process itself, including both classical [4] and deep-learning-based methods [5]. Recent research efforts have focused on sparse iterative approaches [19] to further enhance robustness in uncertain environments.

Finally, the hand-eye calibration of a robot is nothing else but an additional transformation between the real and the nominal robot geometry and can thus also be subject to inaccuracies, as discussed by Nguyen *et al.* [2]. Recent studies have further explored these uncertainties, proposing methods to enhance the accuracy and robustness of hand-eye calibration [20, 21]. These advancements highlight the ongoing need to address and mitigate calibration inaccuracies in vision-guided robotic systems.

In the end, all these sub-fields of robotics provide a multitude of different types of spatial transformations, where potentially all of them are subjected to errors which are being modeled as uncertainties.

Systematic approaches to order a multitude of interconnected transformations, particularly within the area of Virtual Reality (VR) [22, 23], and robotic simulators [24, 25] considered the utilization of a scene graph to represent relative spatial relationships. This scene graph, akin to a tree structure, comprises multiple nodes arranged in a parent-child manner. This innovative approach enhanced the representation and simulation capabilities in both virtual reality and robotic simulation domains. The current state of the art is *tf* [26], the scene graph framework of Robot Operating System (ROS).

Interestingly, very little work has been published that considers the uncertainty of spatial information by interconnecting the different realms of robotics. Initial efforts have been directed towards acknowledging uncertainty within the scene graph, for example [27]. However, these early attempts typically fall short in correctly modeling the error propagation using Lie Algebra. Alternatively, some implementations resort to sampling-based approaches to represent the overall uncertainty within the system, such as Ref. [28], which however comes with computational costs.

The Lie-Algebra allows to acknowledge the manifold character of spatial relationships and is a powerful tool to compute and propagate uncertainty along chains of spatial transformations. An introduction to it together with the application to robotic navigation is provided by Barfoot *et al.* [29]. Similarly, Lie-Algebra-based concepts are provided for the error propagation within robotic manipulators, either for single errors [30] or as our comprehensive kinematic model [13].

Despite the widespread use of Lie Algebra in uncertainty estimation, to the best of our knowledge, no existing approach formulating a scene graph for robotics has integrated Lie Algebra-based uncertainty propagation. In our ongoing work, we aim to address this gap and demonstrate the efficacy of incorporating Lie Algebra into a scene graph framework for a more nuanced and accurate representation of uncertainty in kinematic systems.

III. Robotic and Environmental Configuration State

Accurate assessment of the current configuration state in robotic systems holds significant importance across various applications. This is particularly pronounced in scenarios involving non-static components equipped with perception sensors, where precise positional data is crucial for effective operation. Registering cameras affixed to robotic manipulators to the robot's origin is imperative for seamlessly integrating spatial information within the correct coordinate framework.

Knowledge of the system's distance to the environment is indispensable for collision avoidance, especially when navigating confined spaces. To achieve this, it is crucial to carefully observe and organize the positions of joints into a transformation tree. This tree not only helps illustrate how the coordinate framework depends on a specified starting point known as the root frame, but also aids in obtaining an accurate estimate of the robot's spatial volume and movement range.

However, overlooking the inherent uncertainty in these measurements and the subtle non-static characteristics of certain links—attributable to mechanical stress and gravitational forces—can lead to erroneous state estimations. These factors can significantly impact the reliability of the robot's operation, particularly in dynamic or unpredictable environments.

In the ensuing discussion, we elaborate on Representing the Robotic and Environmental State (RCES) as a transformation tree. We discuss the methodology for constructing this tree, highlighting the importance of each node and its relationship to the overall framework. Subsequently, we introduce Lie Algebra as a robust solution for modeling uncertainty in this process. Lie Algebra provides a mathematical structure that allows for the representation and manipulation of spatial transformations, which is essential for accurately modeling the uncertainties and variances in the robot's configuration.

Finally, we detail our implementation of a managed and centralized approach for addressing the RCES problem within an Inter-Process Communication (IPC) framework. This approach not only centralizes the data processing but also ensures that all components of the robotic system are synchronized and updated in real-time, enhancing the overall accuracy and efficiency of the system.

Throughout this work, we intend to conceptualize the inaccuracies within the system as a form of uncertainty. This approach is motivated by the computational convenience afforded through the utilization of a probabilistic model, as opposed to employing distinct models tailored to individual system errors. By treating all potential errors as probabilistic uncertainties, we can simplify the computational processes and improve the robustness of the system's state estimation.

We believe that this comprehensive approach to modeling and managing uncertainties will significantly enhance the performance and reliability of robotic systems, particularly in complex and dynamic operational environments.

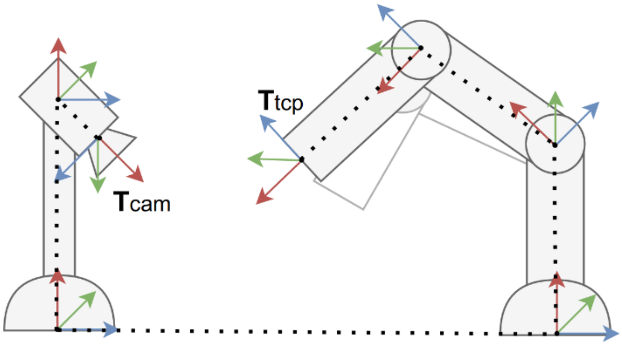

Fig. 1. An illustrated example of a robotic manipulator and an external camera.

A. *Transformation Tree*

Deriving the transformation between two coordinate frames is a pivotal task in robotics. A widely employed approach involves modeling the system as a hierarchical tree of frame transformations illustrates a typical example involving a robotic manipulator and an external camera, as depicted in Fig. 1. To get the transformation between the coordinate frames $\boldsymbol{T}_{cam}$ and $\boldsymbol{T}_{TCP}$, the entire path involving multiple individual transformations must be calculated. In this example, the impact of uncaptured deviations in kinematics from the real world can be observed. The manipulator bends due to gravitational forces, causing the actual position of $\boldsymbol{T}_{TCP}$ to differ from the expected position derived from a naive approach based on exact measurements. This discrepancy highlights the importance of accounting for real-world factors such as mechanical flexibilities and external forces in kinematic modeling to ensure accurate predictions and reliable performance in practical applications.

A key optimization involves consolidating static displacements into a singular transformation, effectively pruning the tree for computational efficiency. This means that static transformations, which do not change over time, are combined into a single transformation matrix. Movable connections are represented as rotations or translations centered around joints, contributing to a chain of static links and dynamic joints. This approach not only streamlines computational complexity but also provides a comprehensive understanding of a robotic system's kinematic properties, enhancing both efficiency and reliability.

One significant advantage of using a hierarchical tree structure is that it can be directly derived from a Computer Aided Design (CAD) model, which inherently uses the same representation. CAD models are typically organized into a hierarchy of parts and subassemblies, mirroring the structure of the transformation tree. This direct correlation allows for seamless integration and accurate transfer of geometric data from design to implementation.

Following the comprehensive description of robot kinematics within the previously mentioned tree structure, the process of retrieving the direct transformation between any two arbitrary frames unfolds by traversing the path articulated within this structured tree. This systematic approach ensures a clear and methodical procedure for obtaining the specific transformation information required for precise spatial relationships between frames within the robotic system.

By organizing transformations into a hierarchical tree structure, we can simplify complex kinematic chains into more manageable sub-problems. This not only reduces the computational burden but also makes the system more scalable and adaptable to changes. Furthermore, the hierarchical model aids in debugging and enhances the modularity of the kinematic analysis, facilitating easier updates and maintenance. An illustration of this is provided in Fig. 2.

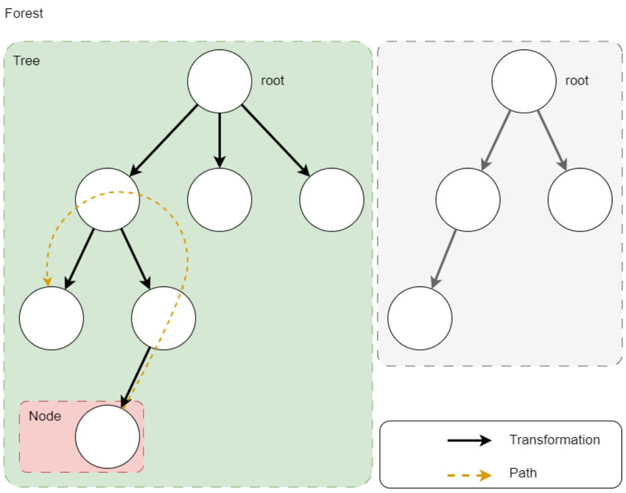

Fig. 2. A schematic overview of the forest and tree structure holding all transformation information.

B. Transformations and Unvertainty

Our treatment of uncertainties follows our previous work on probabilistic robot kinematics [13], which in turn builds upon the mathematical foundations provided by Refs. [29, 30]. We briefly introduce the applied methods here, but refer the interested reader to the related works for more thorough insights. For a general introduction to Lie Algebra in the scope of robotics, we recommend the excellent [31, 32], whose notation we mostly follow.

Lie Algebra provides a mathematical framework for describing the properties and behaviors of Lie groups, which are groups that also have the structure of a smooth manifold. This framework is particularly useful in robotics for representing rotations and rigid body transformations, as these operations form the basis of many kinematic and dynamic calculations.

A pose $\boldsymbol{T}_{AB} \in \mathrm{SE}(3)$ describes the position and orientation of an object B with respect to a reference frame A. The Special Euclidean group SE(3) includes both rotations and translations in three-dimensional space. While a pose quantity is generally an element of the manifold SE(3), it can be described *locally* by its linear tangent space representation $\boldsymbol{\xi} = [\boldsymbol{\rho}\boldsymbol{\theta}]^T \in \mathbb{R}^6$ related by the exponential map:

$$\boldsymbol{T} = \mathrm{Exp}(\boldsymbol{\xi}). \tag{1}$$

Here, $\boldsymbol{\rho}$ denotes the translational component and $\boldsymbol{\theta}$ the rotational component of the tangent space element. The exponential map allows for the conversion between the tangent space (Lie Algebra) and the manifold (Lie group).

In Lie Algebra, the tangent space at the identity element of a Lie group forms a vector space called the Lie Algebra of the group. For SE(3), this tangent space can be represented as a six-dimensional vector comprising three translational and three rotational components. The adjoint representation provides a way to map local tangent space quantities between different coordinate frames.

Local tangent space quantities can be mapped between two different local spaces using the *adjoint matrix* **Ad** as:

$$\boldsymbol{\xi}^A = \mathbf{Ad}(\boldsymbol{T}_{AB})\boldsymbol{\xi}^B, \tag{2}$$

with

$$\mathbf{Ad} = \begin{bmatrix} \boldsymbol{R} & [\boldsymbol{t}]_\times \boldsymbol{R} \\ \mathbf{0} & \boldsymbol{R} \end{bmatrix} \in \mathbb{R}^{6\times 6}, \tag{3}$$

where $\boldsymbol{R}$ is the rotation matrix of $\boldsymbol{T}$ and $[\boldsymbol{t}]_\times$ is the skew-symmetric matrix formed by the translation vector. The term $[\boldsymbol{t}]_\times \boldsymbol{R}$ illustrates how local rotation errors can create translation errors further down a chain of transformations, with the magnitude depending on the distance from the original error's location.

To understand this, consider that any rotation in three-dimensional space can be represented as an element of the SO(3) group, the special orthogonal group, which deals with rotation matrices. Similarly, SE(3) extends this concept to include translations. The Lie algebra of SO(3) consists of skew-symmetric matrices that represent infinitesimal rotations, while the Lie algebra of SE(3) includes both infinitesimal rotations and translations.

We describe the error of a pose as a local deviation $\boldsymbol{\xi}_{\mathrm{B,err}}$ of a nominal pose$\boldsymbol{T}_{AB}$, i.e., in the tangent space of the pose's reference frame B. The corresponding covariance matrix $\boldsymbol{\Sigma}_{AB} = \mathbb{E}\left[\boldsymbol{\xi}_{\mathrm{B,err}}\boldsymbol{\xi}_{\mathrm{B,err}}^T\right] \in \mathbb{R}^{6\times 6}$ is therefore a locally defined tangent space quantity. This covariance matrix encapsulates the uncertainty in both the translational and rotational components of the pose.

The two essential mathematical operations on poses needed for the scene graph are concatenation and inversion. The *concatenation* operation combines two transformations, such as $\boldsymbol{T}_{AB}$ and $\boldsymbol{T}_{BC}$, to yield the transformation from A to C:

$$\boldsymbol{T}_{AC} = \boldsymbol{T}_{AB} \times \boldsymbol{T}_{BC}, \tag{4}$$

$$\boldsymbol{\Sigma}_{AC} = \mathbf{Ad}_{\boldsymbol{T}_{BC}^{-1}} \boldsymbol{\Sigma}_{AB} \mathbf{Ad}_{\boldsymbol{T}_{BC}^{-1}}^{T} + \boldsymbol{\Sigma}_{BC}. \tag{5}$$

Here, the two covariance matrices are transformed into the common reference frame \$C\$ using the adjoint matrix, where they can be added due to the linearity of the tangent space. The covariance composition in Equation 5 is a first-order approximation (referred to as second order in some publications) and is discussed in detail in [29].

Analogously, the *inverse* operation calculates the transformation from B to A given the transformation from A to B:

$$\boldsymbol{T}_{AB} = \boldsymbol{T}_{AB}^{-1}, \tag{6}$$

$$\boldsymbol{\Sigma}_{BA} = \mathbf{Ad}_{\boldsymbol{T}_{AB}} \boldsymbol{\Sigma}_{AB} \mathbf{Ad}_{\boldsymbol{T}_{AB}}^{T}. \tag{7}$$

This shifts the uncertainty from the tangent space of B to the tangent space of A. This representation can implicitly consider *exact* transformations, as zero-covariances simply vanish in Eqs. (5) and (7).

For a more detailed introduction to Lie Algebra and its application in robotics, readers may refer to [32] and other comprehensive resources like Refs. [33, 34].

C. Implementation

The presented methodology has been implemented within a C++ library, and the corresponding source code is accessible online https://github.com/DLR-RM/tf-dude. Additionally, a wrapper for the scripting language Python is provided, facilitating ease of use and integration into various applications. Each coordinate frame is characterized by a node element. A frame is precisely defined by its pose matrix $\boldsymbol{T}$ and an accompanying covariance matrix $\boldsymbol{\Sigma}$, which may be set to zero for precisely known transformations. Distinctive identification of each frame is facilitated through the application of a unique character string.

Furthermore, the mathematical operations of *concatenation* and *inverse* for each frame are executed leveraging the computational capabilities provided by the *manif* library [32], which is augmented by the uncertainty propagation framework. This ensures that transformations account for any uncertainties in the positional data, thereby enhancing the robustness of the system.

The hierarchical structure is implemented using the Boost.Graph data structure [35]. Each vertex encapsulates a frame as its payload, and the edges define the direction of transformations. To determine a path between two nodes within the tree, a Breadth-First Search (BFS) routing algorithm is employed. The cumulative transformation along the identified path is computed based on the direction specified by the graph's edges, facilitating a comprehensive understanding of the transformations between the starting and ending points of the path.

The system allows for the addition of multiple root nodes, thereby declaring new trees that remain disconnected from preceding ones. It is imperative to underscore that the establishment of a path between nodes situated on distinct trees within the forest is not feasible. Each root node initiates an independent tree structure, and inter-tree connectivity is explicitly precluded within the system's framework.

The communication backend is implemented in an IPC-agnostic way, meaning that it can support various implementations of IPC such as ROS [36], ROS2 [37], native DDS [38], links and nodes [39] or other systems. This flexibility is achieved through the use of generic adapters that must be overloaded by the implementation using a plugin functionality. These adapters abstract the communication details, allowing the core library to remain independent of the specific IPC mechanism employed. This design ensures that the system can be easily integrated into different robotic frameworks without requiring significant modifications to the underlying codebase.

The default operational paradigm involves centralized control over all trees, nodes, and computations via a central server. A connected client has the capability to perform various operations such as creating, retrieving, updating, or deleting (CRUD) nodes. Additionally, the client can request the cumulative transformation of a specific path. Other clients can also access this information, but their requests must be routed through the server. This centralized architecture ensures efficient management and coordination of resources.

This implementation offers significant advantages in terms of flexibility and scalability. By leveraging well-established libraries and algorithms, the system ensures high performance and reliability. Furthermore, the clear separation of responsibilities between the server and clients facilitates efficient resource management and provides a robust framework for complex robotic applications. The IPC-agnostic design further enhances the system's adaptability, making it suitable for a wide range of robotic platforms and use cases.

This architecture is illustrated in Fig. 3.

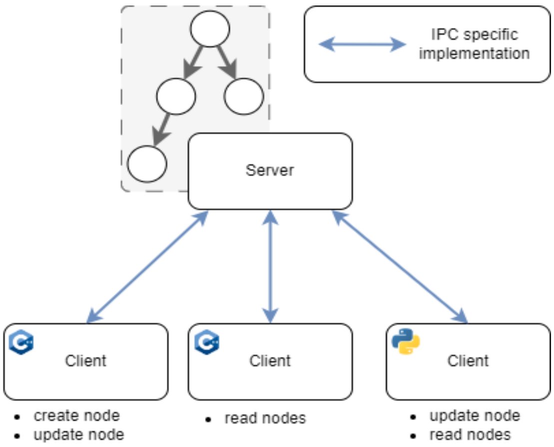

Fig. 3. Illustration of an exemplary server-client architecture with different API implementations.

IV. Evaluation

In this section, we evaluate the proposed framework to assess its overall performance and effectiveness. The evaluation focuses on two primary aspects: computation time and functionality. Computation time analysis examines the efficiency of the framework in performing spatial transformations and uncertainty propagation. This is compared to the naive implementation of robot operation system (ROS) *tf* [40] and the sampling-based implementation of *uncertain tf* [28]. Functionality evaluation focuses on the individual properties of each framework and their applicability to the robotic domain.

A. Computation Time

In this evaluation, we focus on the computation time required to retrieve the concatenated transform for a full path. We define a transformation tree with four levels, similar to the tree illustrated in Fig. 2. For each implementation, we query the combined path from one leaf node to another and measure the execution time. This process is repeated $N = 100'000$ times to gather statistically significant data on the execution time for each implementation.

We use the naive implementation of ROS *tf* as the baseline. While *tf* is widely used in robotic applications, it does not support the inclusion of covariances in the calculations. To address this limitation, an uncertainty-aware approach called *uncertain tf* has been developed. *Uncertain tf* employs a Monte Carlo analysis that samples transforms based on the covariance of each node and propagates them along the path. Although this method can model non-linearities, the sampling-based approach does not scale well to larger kinematic systems, as the number of samples grows exponentially.

Our approach introduces Lie Algebra, offering a closed-form solution for modeling the propagation of uncertainty. This method transforms poses into the tangent space, as detailed in Section III.B, which demands significant computational resources.

The results are shown in Fig. 4. Computation times are normalized to the median execution time of ROS *tf*. The median execution time for our approach is approximately 24.5 times slower than the baseline. In comparison, the implementation of *uncertain tf* takes 60.9 times longer than the baseline, making it 2.5 times slower than our approach.

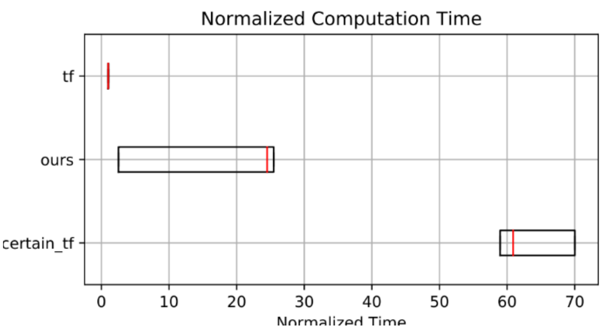

Fig. 4. Normalized computation time of ROS *tf*, our approach and *uncertain tf*.

This demonstrates that our model-based approach significantly outperforms the sampling-based approach while still accounting for uncertainty.

B. Functionality

As discussed earlier, integrating uncertainty into the calculation of transformation trees impacts the computational resources required for processing. Therefore, it is crucial to analyze the properties of each implementation to identify the best fit for each targeted application. The important properties of each approach are listed in Table I.

TABLE I. Implementation Properties

Property	ROS tf	uncertain tf	ours
Uncertainty Model	None	Gaussian	Gaussian
Error Propagation Approach	None	Sampling	Lie Algebra
Integration	Low	Medium	Medium
Computational Cost	Low	High	Medium
Robustness	Low	High	High
IPC-agnostic	No	No	Yes

The naive implementation for ROS *tf* does not incorporate any uncertainty model into the calculation. In contrast, both *uncertain tf* and our approach extend the system to include covariances. While both can propagate errors along the transformation path, our method uses a closed-form solution based on Lie Algebra instead of a sampling approach.

The *tf* package is already part of the default ROS installation, resulting in negligible integration overhead. *Uncertain tf* builds on top of the ROS implementation and requires one additional package for integration. However, our approach, which is not yet integrated into the ROS ecosystem, requires the developer to integrate the system completely. This, however, offers a completely IPC-agnostic approach that does not rely on ROS as a middleware and can work with any communication framework. Lastly, ROS *tf* assumes all measurements are exact, so any deviations will decrease the robustness of the system.

V. Application

To demonstrate the practical utility and broad applicability of the proposed framework, two application examples will be illustrated in the following sections. An in-depth analysis of applying Lie Algebra to the configuration modeling problem has been presented in [13]. Therefore, we will focus on the scene-graph implementation in this discussion.

The first application showcases the integration of the framework on a robotic arm, which is affected by bending induced by the gravitational pull of the Earth. This example highlights how the system compensates for real-world physical effects that deviate from ideal models. By applying the proposed methodology, we can accurately model and correct for these deviations, ensuring the robotic arm operates with high precision despite the bending.

The second application illustrates a mapping task on a system with an uncertain RECS, formulated as a graph optimization problem. This example demonstrates how the framework handles uncertainties in the robotic configuration space, ensuring accurate and reliable mapping. By using a robust scene-graph implementation, the system can dynamically adjust to changes and uncertainties in the environment, maintaining the integrity of the mapping process.

These examples are chosen to underscore the versatility and robustness of the proposed framework in handling various practical challenges in robotics. They provide concrete evidence of how the framework can be applied to real-world scenarios, demonstrating its effectiveness in improving the accuracy and reliability of robotic systems. Through these applications, we aim to showcase the framework's potential for widespread use in diverse robotic applications, highlighting its capability to address complex problems with innovative solutions.

A. *Uncertain Robotic and Environmental Configuration State*

As part of the European Space Agency (ESA) project for a sample transfer arm breadboard study, the German Aerospace Center (DLR) developed the TINA manipulator [41] as a compact, modular, and torque-controlled robotic system designed to meet the requirements of the Mars Sample Return mission. Fig. 5 illustrates the robotic arm in its initial position mounted on a lander.

Fig. 5. TINA arm bending due to gravity.

Upon closer inspection, it becomes evident that the manipulator, even in its initial configuration, experiences moderate deformations attributable to its own weight and joint play, particularly in the axial direction. This can be seen in Fig.5. The computed position, designated as $\boldsymbol{T'}$, represents the theoretical location without accounting for uncertainties. The real position of the end effector is denoted as $\boldsymbol{T}$, and it falls within an anticipated region by incorporating the effects of uncertainties. These deformations introduce uncertainties in the pose of the end effector, which can be effectively modeled using the proposed framework.

By incorporating the expected variance parameters into the transformation tree, the state of the robot configuration can be predicted probabilistically. This allows the position of the end effector to be constrained within an anticipated uncertainty region. Considering these uncertainties provides a more realistic depiction of the arm's pose by acknowledging the impact of various factors, including gravitational forces. This approach enhances spatial awareness and enables more robust manipulations.

The selection of appropriate probabilistic parameters heavily depends on the specific characteristics of the associated system and requires specialized technical knowledge. If necessary, experimental evaluations must be conducted to validate and fine-tune these parameters. This approach ensures that the manipulator's performance remains robust and reliable, even in the presence of inherent uncertainties.

B. *Environmental Maping*

To enable more intricate manipulations and interactions between the robot and its environment, a significant challenge lies in achieving precise registration of the robot relative to its surroundings. This entails aligning various world representations generated for different types of tasks to ensure coherence and accuracy in the robot's perception of its environment.

As depicted in Fig. 6, Rollin' Justin [42] is mapping a Smart Payload Unit (SPU) in Martian surroundings. In addition to the unknown state of the environmental configuration, further challenges arise from within the robot itself. Although the upper body assembly is rigidly connected to the base platform, the wire rope construction in different parts of the torso is inherently less precise than the rigid joints of the arms, introducing uncertainties into the robot's configuration state.

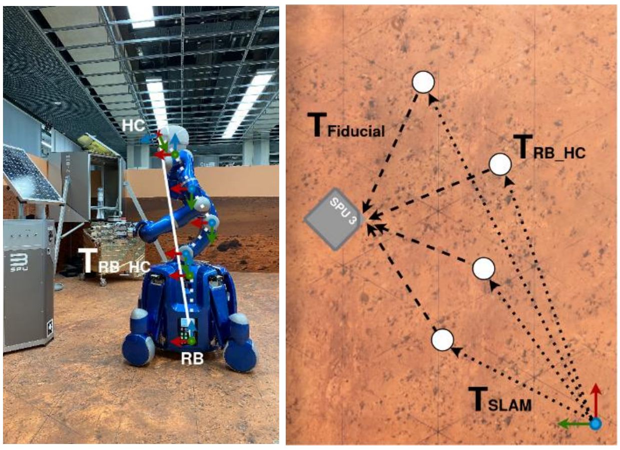

Fig. 6. Rollin' Justin maps an instrument in a Martian environment.

Effectively managing and mitigating this uncertainty is crucial since information for navigation purposes is collected from sensors in the base, while other higher-level tasks, such as object recognition and manipulation, rely on information from the camera mounted in Justin's head. Therefore, modeling the spatial relations of the robot's configuration state, including uncertainties, is essential and can be addressed by the proposed framework. This framework simplifies the handling of transformations and their associated uncertainties by summarizing them into a single step.

In the context of environmental mapping, the transformation from the robot base to the head camera becomes particularly critical as it serves as the foundation for registering fiducial markers linked to the SPU. Combined with the spatial relationship to the registered fiducials and information regarding the global reference provided by MROSLAM [3], an optimization graph can be constructed, as illustrated in Fig. 6. The optimization problem can be effectively addressed using GTSAM [43] or comparable algorithms, leading to an optimized estimation of the SPU's pose.

This comprehensive approach significantly improves the reliability and quality of environmental mapping outcomes in the robot's operational context. By integrating precise registration techniques and robust uncertainty modeling, the framework enhances the robot's ability to interact accurately and efficiently with its environment, ensuring higher levels of performance in complex tasks.

VI. Conclusion

This paper introduces a robust framework for representing uncertain spatial transformations in robotic systems, leveraging Lie Algebra for a structured and probabilistic approach. Traditional deterministic methods often fall short in accounting for the inherent inaccuracies and environmental factors that affect robotic operations. Our proposed framework addresses these limitations by incorporating uncertainty into transformation trees, providing a more realistic and reliable computation of spatial transformations.

The framework models inaccuracies arising from sensor decalibration, joint position errors, mechanical stress, and gravitational influences, as well as environmental uncertainties from perception limitations. By integrating probabilistic models into the transformation calculations, we offer a robust and adaptable solution for various robotic applications, enhancing the system's ability to handle real-world complexities.

We evaluated the approach based on computational time with the naive implementation of Robot Operating System (ROS) *tf* and its extension *uncertain tf* and compared functionalities across these approaches.

We are able to show that our method outperforms the current state-of-the-art approach for uncertain transformation trees in terms of computational complexity. Further, we demonstrate the practical utility of the proposed framework through two application examples. The first example involves a robotic arm affected by gravitational bending, showcasing how the system considers for real-world physical effects that deviate from ideal models. The second example illustrates a mapping task on a system with an uncertain Robot and Environmental Configuration State (RECS), formulated as a graph optimization problem. These applications highlight the framework's effectiveness in improving positional accuracy and enabling precise manipulations.

The hierarchical transformation tree structure not only simplifies complex kinematic chains but also provides a comprehensive understanding of the robot's spatial relationships. This approach reduces computational complexity and enhances the scalability and adaptability of the system. Additionally, the IPC-agnostic design allows for easy integration into different robotic frameworks, further enhancing the system's versatility.

Future work includes extending the framework to model temporal deviations, enabling configuration retrieval from previous time steps. We also aim to align the interface with ROS's *tf* implementation for seamless integration.

In summary, this contribution significantly advances the management of spatial transformation uncertainties in robotics, providing a versatile and robust tool that enhances the reliability and performance of robotic systems in diverse applications. The source code for this framework is accessible online https://github.com/DLR-RM/tf-dude.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Idea, architecture and integration has been prepared by MS, mathematical background has been provided by LB, support in integration and experimental evaluation by XL. Support in integration by LD. Organizational and theoretical support by RT. Manuscript preparation and organization by MS. All authors had approved the final version.

Funding

This work was supported by the Bavarian Ministry of Economic Affairs, Regional Development and Energy (StMWi) by means of the project SMiLE2gether (LABAY102).

References

[1] U. Seibold, B. Kubler, T. Bahls, R. Haslinger, and F. Steidle, "The DLR MiroSurge surgical robotic demonstrator," in *The Encyclopedia of Medical Robotics*, vol. 1: Minimally Invasive Surgical Robotics, World Scientific, 2019, pp. 111–142.

[2] Nguyen *et al.*, "On the covariance of X in AX = XB," *IEEE Transactions on Robotics*, 2018.

[3] Sewtz *et al.*, "Robust approaches for localization on multi-camera systems in dynamic environments," in *Proc. 2021 7th International Conference on Automation, Robotics and Applications (ICARA)*, 2021.

[4] Stoiber *et al.*, "A sparse gaussian approach to region-based 6dof object tracking," in *Proc. the Asian Conference on Computer Vision*, 2020.

[5] L. Burkhard *et al.*, "Robust probabilistic robot arm keypoint detection exploiting kinematic knowledge," in *Proc. 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems, Workshop on Probabilistic Robotics in the Age of Deep Learning*, 2022.

[6] A. B. W. Kennedy and F. Reuleaux, *The Kinematics of Machinery*, New York: D. Van Nostrand, 1881.

[7] J. J. Uicker *et al.*, *Theory of Machines and Mechanisms*, Cambridge University Press, 2023.

[8] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *J. Appl. Mech.*, vol. 22, no. 2, pp. 215–221, 1955.
[9] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC Press,1994.
[10] T. Low and S. Calinon, "Geometric algebra for optimal control with ¨ applications in manipulation tasks," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3586–3600, 2023.
[11] J. Lu, T. Zou, and X. Jiang, "A neural network based approach to inverse kinematics problem for general six-axis robots," *Sensors*, vol. 22, 2022.
[12] A. Malik, Y. Lischuk, T. Henderson, and R. Prazenica, "A deep reinforcement-learning approach for inverse kinematics solution of a high degree of freedom robotic manipulator," *Robotics*, vol. 11, 2022.
[13] L. Burkhard *et al.*, "The probabilistic robot kinematics model and its application to sensor fusion," in *Proc. 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.
[14] M. Kaess *et al.*, "isam2: Incremental smoothing and mapping using the bayes tree," *The International Journal of Robotics Research*, vol. 31, issue 2, pp. 216–235, 2012.
[15] R. Kummerle *et al.*, "g2o: A general framework for graph optimization," in *Proc. 2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, May 9–13, 2011.
[16] S. Choudhary, L. Carlone, C. Nieto, J. Rogers, H. I. Christensen, and F. Dellaert, "Distributed trajectory estimation with privacy and communication constraints: A two-stage distributed gauss-seidel approach," in *Proc. 2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 5261–5268.
[17] C. Li, G. Guo, P. Yi, and Y. Hong, "Distributed pose-graph optimization with multi-level partitioning for multi-robot slam," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 4926–4933, 2024.
[18] S. F. Su *et al.*, "Manipulation and propagation of uncertainty and verification of applicability of actions in assembly tasks," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. vol. 22, no. 6, pp. 1376–1389,1992.
[19] M. Stoiber, M. Sundermeyer, and R. Triebel, "Iterative corresponding geometry: Fusing region and depth for highly efficient 3d tracking of textureless objects," in *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6855–6865.
[20] I. Ali, O. Suominen, A. Gotchev, and E. R. Morales, "Methods for simultaneous robot-world-hand-eye calibration: A comparative study," Sensors, vol. 19, no. 12, 2837, 2019.
[21] I. Enebuse, B. Ibrahim, M. Foo, R. S. Matharu, and H. Ahmed, "Accuracy evaluation of hand-eye calibration techniques for vision-guided robots," *PLoS One*, vol. 17, no. 10, e0273261, 2022. doi: 10.1371/journal.pone.0273261
[22] C. Carlsson and O. Hagsand, "Dive—A platform for multi-user virtual environments," *Computers & Graphics*, vol. 17, no. 6, pp. 663–669, 1993.
[23] H. Tramberend, "Avocado: A distributed virtual reality framework," in *Proc. IEEE Virtual Reality (Cat. No. 99CB36316)*, IEEE, 1999, pp. 14–21.
[24] B, Browning and E. Tryzelaar, "Ubersim: A multi-robot simulator for robot soccer," in *Proc. the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, 2003, pp. 948–949.
[25] E. Drumwright, J. Hsu, N. Koenig, and D. Shell, "Extending open dynamics engine for robotics simulation," in *Simulation, Modeling, and Programming for Autonomous Robots*, N. Ando, S. Balakirsky, T. Hemker, M. Reggiani, O. von Stryk, Eds. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2010, vol 6472. https://doi.org/10.1007/978-3-642-17319-6_7, 2010
[26] T. Foote, "tf: The transform library," in *Proc. IEEE Conference on Technologies for Practical Robot Applications (TePRA)*, IEEE, 2013.
[27] E. M. Coelho, B. MacIntyre, and S. J. Julier, "OSGAR: A scene graph with uncertain transformations," in *Proc. IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2004.
[28] T. Ruehr. (2013). uncertain tf. [Online]. Available: https://github.com/ruehr/uncertain tf
[29] T. D. Barfoot *et al.*, "Associating uncertainty with three-dimensional poses for use in estimation problems," *IEEE Transactions on Robotics*, vol. 30, issue 3, June 2014.
[30] Y. Wang *et al.*, "Error propagation on the euclidean group with applications to manipulator kinematics," I*EEE Transactions on Robotics*, vol. 22, no. 4, 2006
[31] Y. Wang and G. S. Chirikjian, "Nonparametric second-order theory of error propagation on motion groups," *The International Journal of Robotics Research*, vol. 27, no. 11–12, pp. 1258–1273, 2008.
[32] J. Sol *et al.*, "A micro Lie theory for state estimation in robotics," arXiv preprint, arXiv:1812.01537, 2018.
[33] G. S. Chirikjian, "Information theory on Lie groups and mobile robotics applications," in *Proc. 2010 IEEE International Conference on Robotics and Automation*, IEEE, 2010.
[34] P. Coelho and U. Nunes. (2004). Lie algebra application to mobile robot control: A tutorial. *Robotica*. [Online]. 22(5), pp. 553–563. Available: https://www.cambridge.org/core/journals/robotica/article/lie-algebra-application-to-mobile-robot-control-a-tutorial/
[35] Boost. (2023). Boost C++ Libraries. [Online]. Available: http://www.boost.org/
[36] M. Quigley, B. Gerkey, and W. D. Smart. (2009). Ros: an open-source robot operating system. [Online]. Available: http://www.ros.org
[37] R. Contributors. (2018). Ros2: The robot operating system. [Online]. Available: https://index.ros.org/doc/ros2/
[38] O. M. G. (OMG). (2004). Data Distribution Service (DDS). [Online]. Available: https://www.omg.org/spec/DDS/
[39] F. Schmidt. (2024). Links and nodes. [Online]. Available: https://gitlab.com/links_and_nodes/links_and_nodes
[40] Stanford Artificial Intelligence Laboratory *et al.* Robotic operating system. [Online]. Available: https://www.ros.org
[41] M. Maier *et al.*, "TINA: The modular torque controlled robotic arm—A study for mars sample return," in *Proc. 2021 IEEE Aerospace Conference (50100)*, 2021.
[42] M. Fuchs *et al.*, "Rollin' justin—Design considerations and realization of a mobile platform for a humanoid upper body," in *Proc. 2009 IEEE International Conference on Robotics and Automation*, 2009.
[43] Dellaert *et al.* (May 2022). borglab/gtsam. [Online]. Available: https://github.com/borglab/gtsam