2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Accurate Kinematic Modeling using Autoencoders on Differentiable Joints

Nikolas Wilhelm[1,2], Sami Haddadin[1], Rainer Burgkart[2], Patrick van der Smagt[3] and Maximilian Karl[3]

***Abstract*— In robotics and biomechanics, accurately determining joint parameters and computing the corresponding forward and inverse kinematics are critical yet often challenging tasks, especially when dealing with highly individualized and partly unknown systems. This paper unveils a cutting-edge kinematic optimizer, underpinned by an autoencoder-based architecture, to address these challenges. Utilizing a neural network, our approach simulates inverse kinematics, converting measurement data into joint-specific parameters during encoding, enabling a stable optimization process. These parameters are subsequently processed through a predefined, differentiable forward kinematics model, resulting in a decoded representation of the original data. Beyond offering a comprehensive solution to kinematics challenges, our method also unveils previously unidentified joint parameters. Real experimental data from knee and hand joints validate the optimizer's efficacy. Additionally, our optimizer is multifunctional: it streamlines the modeling and automation of kinematics and enables a nuanced evaluation of diverse modeling techniques. By assessing the differences in reconstruction losses, we illuminate the merits of each approach. Collectively, this preliminary study signifies advancements in kinematic optimization, with potential applications spanning both biomechanics and robotics.**

I. Introduction

In the fields of robotics and biomechanics, determining joint positions and computing forward and inverse kinematics are essential. A significant challenge arises when multiple systems assess the same rigid body without knowledge of their relative positioning, resulting in missing transformation functions in forward kinematics [1]. While traditional methods have been foundational, they often face limitations in speed and precision, especially in intricate robotic systems [2].

Recent studies underscore the efficacy of neural network-based methods in kinematic challenges. Köker et al. [3] as well as Polyzos et al. [4] introduced a neural network solution for inverse kinematics in a three-joint robotic manipulator, demonstrating the capability of neural networks in determining accurate joint angles from cartesian coordinates. Duka [5] and Jiang et al. [2] further showcased the versatility of neural networks in robotic arm trajectory tracking and the integration of particle swarm optimization with back propagation networks, respectively, for enhanced kinematic solutions. Segota et al. [6] emphasized the role of multilayer perceptrons in industrial robotic manipulators. In biomechanics, deep learning has illuminated human movement nuances, with works like Henry et al. [7] on foot deformities and Sun et al. [8] on the synergy of neural circuits and body mechanics.

Within the context of these advancements, differentiable kinematics has emerged as a promising avenue. Ono et al. [9] presented kinematics for a Stewart platform using moving frames, emphasizing the importance of the special orthogonal and special Euclidean groups for effective matrix computations. Furthermore, Fang et al. [10] introduced an efficient learning-based method to address the inverse kinematic problem in soft robots, highlighting the potential of Jacobian-based iterations and sim-to-real transfer strategies. Some promising software solutions have emerged as well, such as Meier et al. [11] with their software solution for differentiable robotics and Mölschl et al. [12] with their solution for differentiable kinematics using tensorflow 2 [13].

Within this context, autoencoders have emerged as a pivotal tool. Their ability to encode and decode complex data positions them as ideal candidates for kinematic optimization. Kubovčík et al. [14] demonstrated this by using an autoencoder for novelty detection in robotic control. Similarly, Nagano et al. [15] employed a convolutional variational autoencoder for categorizing videos captured by mobile robots. Midhun and Kurian [16] further showcased the potential of autoencoders in task-level learning in robotics.

This research introduces an autoencoder-based kinematic optimizer, processing measurement data (X_{mea}) to model inverse kinematics and derive joint positions (Q), which are decoded to reconstruct $\hat{X}_{\mathrm{mea}}$ specifically designed for unknown biomechanical systems where the axis are undefined or different modelling strategies can be compared easily. Validations using knee and hand joint measurements underscore the optimizer's potential in kinematic optimization across biomechanical and robotic spheres. This preliminary study uniquely integrates differentiable kinematics software with autoencoder-based inverse kinematics, offering a comprehensive solution validated with hand and knee joint data. We release the source code for differentiable forward kinematics and optimization in the GitHub repository `https://github.com/NikonPic/DiffAutoKin`.

*This work was not supported by any organization

[1]Nikolas Wilhelm and Sami Haddadin are with the Munich Institute of Robotics and Machine Intelligence, Technical University of Munich, 80992 Munich, Germany `nikolas.wilhelm@tum.de`

[2]Nikolas Wilhelm and Rainer Burgkart are with the Department of Orthopedics and Sports Orthopedics, Klinikum rechts der Isar, School of Medicine, 81675 Munich, Germany

[3]Maximilian Karl and Patrick van der Smagt are with the Volkswagen Group, Machine Learning Research Lab, 80805 Munich, Germany `karlma@argmax.ai`

II. Materials and Methods

A. Differential Kinematics

The analysis of kinematic data often employs transformation matrices to define spatial relationships between coordinate frames. These matrices typically comprise a rotation matrix, R, and a translation vector, p. Vector differentiation is pivotal in discerning relative velocity and acceleration in kinematics, as highlighted by Josephs et al. [17]. Quaternions offer a compact representation of the rotation matrix, facilitating descriptions of orientations and rotations in three-dimensional space [18].

The transformation matrix linking coordinate frames 0 and 1 is expressed as:

$$X_{0,1} = \begin{bmatrix} R_{0,1} & p_{0,1} \\ 0 & 1 \end{bmatrix}; X_{1,0} = \begin{bmatrix} R_{0,1}^T & -R_{0,1}^T p_{0,1} \\ 0 & 1 \end{bmatrix} \quad (1)$$

where $X_{0,1}$ and $X_{1,0}$ denote the transformation matrices from frame 0 to 1 and vice versa, respectively. The rotation matrix, $R_{0,1}$, can be efficiently represented using quaternions. Our approach to differential kinematics harnesses these matrices, offering parallelizability, full differentiability, and a structure conducive to algorithmic formalization.

In fields like robotics and biomechanics, a rigid body's position in 3D space is often represented by a transformation matrix, T_{o-b_i}, indicating the position of body b_i relative to an origin frame o. Such positions can be sourced from tracking systems like OptiTrack [19] or Vicon [20], or derived from a forward kinematics model given known joint angles, q: $T_{o-b_i}(q)$. For a kinematics model based solely on transformation matrices, we distinguish between transformations of rigid bodies, T_{b_i}, and joint-related transformations dependent on joint positions, q_i, denoted as $T_j(q_i)$.

a) Rigid Body Transformations: For transformations pertaining to rigid bodies, the parametrization is rooted in quaternions and offset vectors. A quaternion, symbolized as $\text{quat}_{b_i} = [w, x, y, z]$, characterizes the spatial orientation of the rigid body. The offset vector, $\text{offset}_{b_i} = [x, y, z]$, provides the translational element, indicating the position of the rigid body relative to a reference frame. From these parameters, the transformation matrix T_{b_i} is derived, encapsulating both the rotational data from the quaternion and the translational information from the offset vector.

b) Joint Transformations for Hinge Joints: Hinge joints primarily undergo rotational transformations. The position of the rotational axis is given by $\text{position}_i = [x, y, z]$, and its orientation by the axis vector $\text{axis}_i = [a_x, a_y, a_z]$, summarized by the parameter vector θ_i. The transformation matrix for such a joint is determined through the following steps:

1) **Translation to the Joint Axis:** The system is first translated to the rotational axis using $T_{\text{trans}_{ji}}$, derived from the position vector.
2) **Rotation about the Joint Axis:** At the joint axis, the system undergoes a rotation defined by $T_{\text{rot}_{ji}}(q_i)$. This rotation matrix, derived using Rodrigues' rotation formula, is a function of both the axis vector and the joint angle q_i. Specifically, for an angle q about the axis, the rotation matrix R is:

$$R = I + \sin(q) \times K + (1 - \cos(q)) \times K^2 \quad (2)$$

where I is the identity matrix and K is the skew-symmetric matrix:

$$K = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \quad (3)$$

3) **Inverse Translation from the Joint Axis:** The system is then translated back to its original position using $T_{\text{trans}_{ji}}^{-1}$.

The overall transformation matrix for the hinge joint is:

$$T_j(q_i) = T_{\text{trans}_{ji}} \cdot T_{\text{rot}_{ji}}(q_i) \cdot T_{\text{trans}_{ji}}^{-1} \quad (4)$$

This matrix encapsulates the joint's spatial position and orientation, considering its rotational axis, position, and joint angle.

c) Holistic Kinematics Structure: The kinematic structure integrates both rigid bodies and joints to define the transformation from the base to the end effector, denoted as T_{eff}. This transformation can be represented as:

$$T_{\text{eff}} = \prod_{i=1}^{n} T_{b_i} \cdot T_j(q_i) \quad (5)$$

where T_{eff} is the cumulative transformation from the base to the end effector, encompassing all rigid bodies and joints in the kinematic chain. Each term in the product represents the transformation associated with a specific body or joint in the structure. The final kinematics chain is visualized in Figure 1.

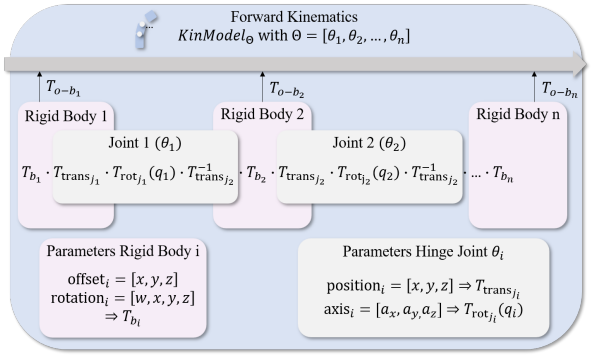

Fig. 1. Forward Kinematics Model Visualization. The model illustrates the dependency on joint parameters Θ and showcases the transformations T_{o-b_i} derived from the kinematic chain.

d) Differentiable Kinematics Software Solution: In addition to the theoretical foundations and methodologies discussed, we have developed a comprehensive software solution that facilitates the generation of a differentiable kinematics model. This software is designed to automatically parse a Mujoco-based XML file [21] obtained from the biomechanical simulator, streamlining the process of model creation based on the PyTorch framework [22]. The source code is publicly available and can be accessed at `https://github.com/NikonPic/DiffAutoKin`.

B. Autoencoder-based Kinematic Optimizer

Leveraging the principles of differentiable forward kinematics, we present a state-of-the-art kinematics optimizer anchored in autoencoder neural network architectures. This section elucidates the architecture's design, operational mechanics, and the inherent advantages of this approach for kinematic optimization.

Autoencoders are neural networks tailored for unsupervised learning, aiming to learn efficient data encodings. At its core, an autoencoder compresses input data into a condensed representation and subsequently reconstructs the original data from this compressed form. Given an input x, it is encoded to a representation y through $y = f(x)$. This representation is then decoded to produce a reconstruction $\hat{x}$ such that $\hat{x} = g(y)$. The training goal is to minimize the reconstruction error, often measured by the mean squared error (MSE): $L(x, \hat{x}) = \|x - \hat{x}\|_2$ [23].

We can leverage the auto-encoder architecture by interpreting the encoder and decoder as inverse and forward kinematics, respectively. Consider a dataset X_{mea} from a tracking system, comprising m pose measurements represented as transformation matrices: $X_{\text{mea}} = [T_{\text{mea}}(t_1), \ldots, T_{\text{mea}}(t_m)]$. Notably, each matrix $T_{\text{mea}}(t_i)$ can be further decomposed into n distinct transformation matrices T_{o-b_j}, signifying the poses of the tracked rigid bodies. The encoder, represented as $Q : Q = \text{Inv}(X_{\text{mea}})$, translates this data into joint angles $Q = [\overrightarrow{q_1}, \ldots, \overrightarrow{q_m}]$, where each vector $\overrightarrow{q_m}$ presents the joint angles of the kinematic system and is defined as $\overrightarrow{q} = [q_1, \ldots, q_n]$. The inverse kinematics solution is modeled by a neural network NN_Ψ. These joint angles are then decoded using the differentiable forward kinematics model, parameterized by $\Theta = [\theta_1, \ldots, \theta_n]$, to produce the reconstructed data $\hat{X}_{\text{mea}} = \text{KinModel}_\Theta(Q)$. This approach ensures the kinematics model's mathematical integrity is preserved, and the gradient is maintained throughout the autoencoder, facilitating backpropagation-based updates.

The optimization goal is to minimize the difference between X_{mea} and $\hat{X}_{\text{mea}}$. This reconstruction error forms our loss function:

$$\mathcal{L}(X_{\text{mea}}, \hat{X}_{\text{mea}}) = \frac{1}{m} \sum_{i=1}^{m} \|T_{\text{mea}}(t_i) - \hat{T}_{\text{mea}}(t_i)\|_2 \tag{6}$$

Using gradient-based optimization, parameters Ψ and Θ are iteratively refined to minimize $\mathcal{L}$. The choice of optimization algorithm can be adapted based on the problem specifics and computational considerations.

$$\Psi^*, \Theta^* = \arg\min_{\Psi,\Theta} \mathcal{L}(X_{\text{mea}}, \hat{X}_{\text{mea}}) \tag{7}$$

Through this optimization, the autoencoder adeptly addresses both inverse and forward kinematics challenges, offering a holistic solution for kinematic analyses. The integrated workflow is depicted in Figure 2.

C. Regularization Techniques

To improve the robustness and generalizability of our autoencoder-based kinematic optimizer, we integrated two advanced regularization techniques: Independent Component Analysis (ICA) and Variational Autoencoder (VAE).

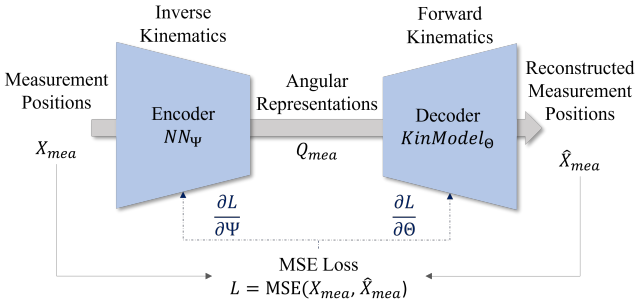

Fig. 2. Overview of the autoencoder architecture for kinematics. The encoder translates measurement data X_{mea} to joint angles Q, while the decoder reconstructs this data using the differentiable kinematics model. Gradients, shown as dotted arrows, flow through the entire architecture during optimization, ensuring minimized reconstruction error and integrating solutions for both inverse and forward kinematics.

Independent Component Analysis (ICA): ICA traditionally decomposes a multivariate signal into additive, independent non-Gaussian components [24]. In the context of our model, the objective is to encourage the components of the encoded representation to be as independent as possible. To achieve this, we introduce an ICA-inspired regularization term, L_{ICA}, which is defined as:

$$L_{\text{ICA}} = \sum_{i \neq j} \text{cov}(q_i, q_j)^2 \tag{8}$$

where q_i and q_j are the components of the encoded representation q. This loss function computes the penalty based on the off-diagonal elements of the covariance matrix of the encoded representations. By penalizing the squared values of these off-diagonal elements, the loss encourages the encoded features to be more independent of each other. The L_{ICA} term is then combined with the primary reconstruction loss to train the model.

Variational Autoencoder (VAE): VAEs extend the autoencoder paradigm by encoding input data, such as X_{mea}, into a probabilistic distribution, thereby introducing a stochastic dimension to the autoencoder framework [25]. The VAE loss is a combination of the reconstruction loss and the KL-divergence, L_{KL}. The KL-divergence ensures that the encoded representations are regularized towards a standard normal distribution:

$$L_{\text{KL}} = -\frac{1}{2} \sum_i (1 + \log(\sigma_i^2) - \mu_i^2 - \sigma_i^2) \tag{9}$$

where μ_i and σ_i^2 represent the mean and variance of the encoded distribution for the i^{th} sample.

The reconstruction loss is modeled considering the decoder's output as a Gaussian distribution. The mean of this distribution is determined by the output of the neural network, NN_Ψ, while its variance is parameterized by a learnable log standard deviation, initialized to $\log(0.05)$:

$$L_{\text{rec}} = -\mathbb{E}_{Q \sim q_\Phi(Q|X_{\text{mea}})}[\log p_\Theta(X_{\text{mea}}|Q)] \tag{10}$$

where $p_\Theta(X_{\text{mea}}|Q)$ denotes the Gaussian distribution defined by the decoder. The overall VAE loss, L_{VAE}, is then given by:

$$L_{\text{VAE}} = L_{\text{rec}} + \lambda L_{\text{KL}} \tag{11}$$

with λ serving as a weighting factor.

Both ICA and VAE were utilized independently, offering unique benefits: ICA promotes feature independence, ideal for models with redundant axes requiring decoupling, whereas VAE enhances the smoothness and generalizability of the latent space, thus increasing robustness against noisy measurement data. By integrating these approaches, our goal was to develop a kinematic optimizer that achieves a balance between accuracy, robustness, and interpretability.

D. Experimental Validation

To rigorously evaluate the autoencoder-based kinematic optimizer, especially when augmented with ICA and VAE techniques, we conducted experiments using kinematic data from two real-world setups, as depicted in Figure 3.

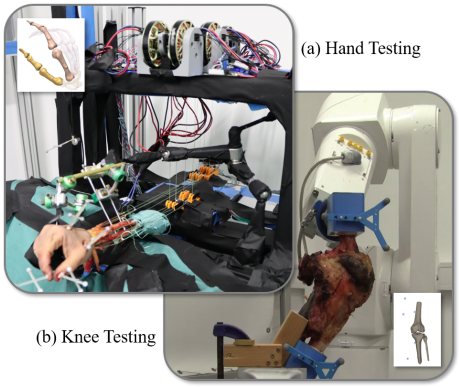

Fig. 3. Experimental setups for kinematic data acquisition: (a) Hand testbench capturing movements of the index finger's MCP, PIP, and DIP joints, and the thumb's CMC, MCP, and IP joints (authors' unpublished data). (b) Data from Wilhelm et al. [26] representing knee joint motions across multiple DOFs.

The base models originate from the primary MuJoCo models of the respective datasets and are provided in the GitHub repository. These models are first converted into kinematic representations and subsequently reformatted to MuJoCo's format for evaluation. For the hand testbench, we analyzed joint positions of the index finger (MCP, PIP, DIP) and thumb (CMC, MCP, IP) during pincer grip motion. The knee joint, given its multiple DOFs, required testing on three distinct models (1 DOF, 2 DOFs, 3 DOFs) using the dataset from Wilhelm et al. [26]. Our algorithm aimed to identify optimal representations, especially for the 1 and 2 DOF models.

Further analysis was conducted using VAE and ICA techniques, with derived motion axes visualized. The primary evaluation metric was the reconstruction loss (MSE), applied to an independent validation subset (10% of total data).

III. Results

This section presents the performance evaluation of the autoencoder-based kinematic optimizer, emphasizing the enhancements provided by ICA and VAE regularization techniques, using data from a hand testbench and a knee joint dataset with the task to find all joint axis and solve the corresponding inverse kinematics problem.

A. Hand Testbench Analysis

The hand testbench experiments were conducted to assess the performance of three autoencoder variants: the standard Autoencoder (AE), the Autoencoder augmented with Independent Component Analysis (AE+ICA), and the Variational Autoencoder (VAE). These experiments had a dual objective. Firstly, simulated data was generated based on predefined anatomical axis positions, providing the initial dataset for validation. Secondly, actual measurement data from the testbench was employed for kinematic identification. For both scenarios, the decoder model's axis parameters were initialized at random.

Figure 4 presents the validation reconstruction loss, quantified by MSE, over three independent training iterations for each variant and dataset. Adjacent to the loss plots, hand models depict the trained joint axes, highlighted using dotted lines corresponding to the color of each model's loss curve.

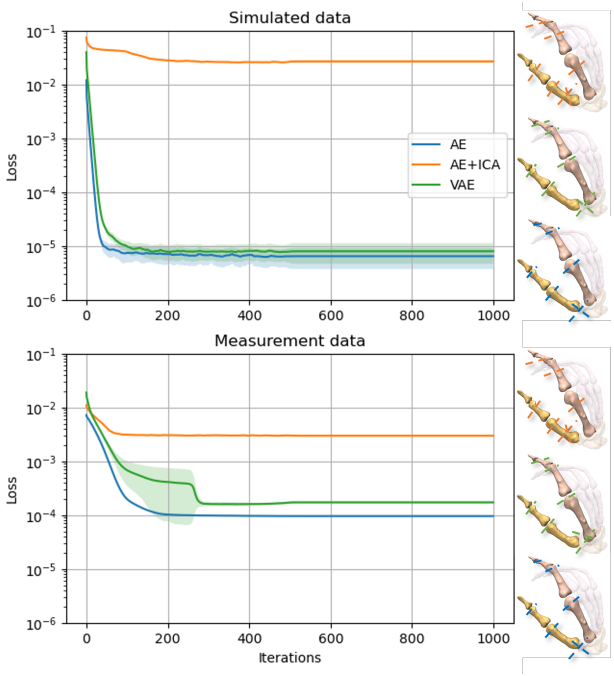

Fig. 4. Training progression for hand model joint optimization across three independent runs for the simulated data (top) and the actual measurement data (bottom). The MSE loss on the validation dataset as the mean over all joints from the hand-testbench data is plotted for AE (blue), AE+ICA (orange), and VAE (green). Beside each loss curve, hand models illustrate the trained joint axes, emphasized using dotted lines in the respective model colors.

In terms of performance, the AE+ICA variant exhibited the highest error with losses exceeding 2×10^{-2} for simulated and 3×10^{-3} for measurement data. The VAE variant

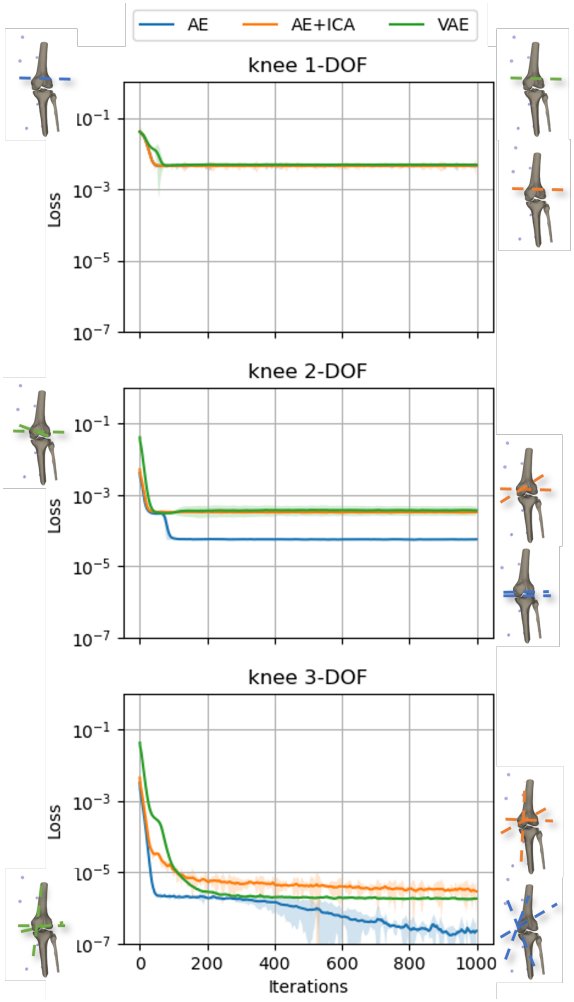

Fig. 5. Training progression for three model variations: 1-DOF (top), 2-DOF (middle), and 3-DOF (bottom), evaluated using the MSE loss on the validation dataset from Wilhelm et al. [26] over three independent training runs. The models encompass the standard autoencoder (AE, blue), the ICA-enhanced autoencoder (AE+ICA, orange), and the Variational Autoencoder (VAE, green). Adjacent knee models illustrate the derived joint axes, emphasized using dotted lines.

followed with losses greater than 7×10^{-6} for simulated and 1×10^{-4} for measurement data. In contrast, the AE variant showcased the best performance, achieving losses below 9×10^{-7} for simulated and 6×10^{-6} for measurement data.

B. Knee Joint Analysis

The knee joint, characterized by its intricate structure and multiple degrees of freedom (DOFs), necessitated a nuanced evaluation. We assessed our optimizer's performance across three distinct models, each representing 1 DOF, 2 DOFs, and 3 DOFs. Figure 5 visualizes the reconstruction loss for each model.

For the 1 DOF model, all variants (AE, AE+ICA, VAE) reported errors exceeding 10^{-3}. In the 2 DOF model, the AE+ICA and VAE exhibited MSE greater than 3×10^{-4}, while the AE was the most accurate, registering below 10^{-4}. The AE model attained its objective by aligning the two available axes to be nearly parallel, utilizing high angular values. This alignment resulted in low reconstruction errors, as evidenced by the significant decrease observed in the loss curve. For the 3 DOF configuration, the AE+ICA was the highest with a loss above 2×10^{-6}, while VAE achieved 1.7×10^{-6} and AE recorded errors below 1×10^{-6}. A summary of the Mean Squared Error (MSE) values for each model across different Degrees of Freedom (DOF) is provided in Table I.

TABLE I

MEAN MSE VALUES REPRESENTING THE RECONSTRUCTION LOSS OF THE MEASUREMENT DATA FOR EACH VARIANT ACROSS DIFFERENT MODELS, AVERAGED OVER THREE INDEPENDENT TRAINING RUNS ON THE REAL DATASETS.

Model	AE	AE+ICA	VAE
knee 1-DOF	4.5×10^{-3}	4.5×10^{-3}	4.6×10^{-3}
knee 2-DOF	5.6×10^{-5}	3.2×10^{-4}	3.1×10^{-4}
knee 3-DOF	1.6×10^{-7}	2.8×10^{-6}	1.7×10^{-6}
hand 7-DOF	9.6×10^{-5}	3.0×10^{-3}	1.6×10^{-4}

C. Decoupling Joint Trajectories: 3-DOF AE Compared to 3-DOF AE+ICA

All model variants, including AE (error 1×10^{-6}), AE+ICA (error 1×10^{-5}), and VAE (error 1×10^{-6}), offer valid solutions for both forward and inverse kinematics due to their low reconstruction errors for the presented problem. For the following analysis we will however only focus on the resulting joint trajectories for the AE and AE+ICA versions for comparison to focus on the decoupling effect of the AE+ICA variant only. The joint trajectories, derived from Wilhelm et al.'s dataset [26], are illustrated in Figure 6.

The AE+ICA method distinctly separates the flexion axis (orange) from the other two axes: internal-external rotation (green) and varus-valgus (blue). In contrast, the AE method shows coupled trajectories, especially during high flexion angles (timesteps 80,000 to 100,000), resulting in significant angles across all axes.

IV. Discussion

This section delves into the implications of the experimental results, evaluating the performance and adaptability of the autoencoder-based kinematic optimizer, especially when enhanced with ICA and VAE regularization techniques.

A. Hand Testbench

The hand testbench results highlight the AE and VAE variants' commendable performance in reconstructing simulated data, both registering errors around 1×10^{-5}. In terms of the identified joint axes, they align closely with the predefined anatomical axes used for data generation. For the measurement data, we observed slightly elevated errors, which are consistent with the measurement inaccuracies inherent in the data. The axes defined by the optimizer also correlate well with the expected anatomical axes. In contrast, the ICA variant struggled with multiple one-dimensional axes, attempting to further decouple these inherently independent axes.

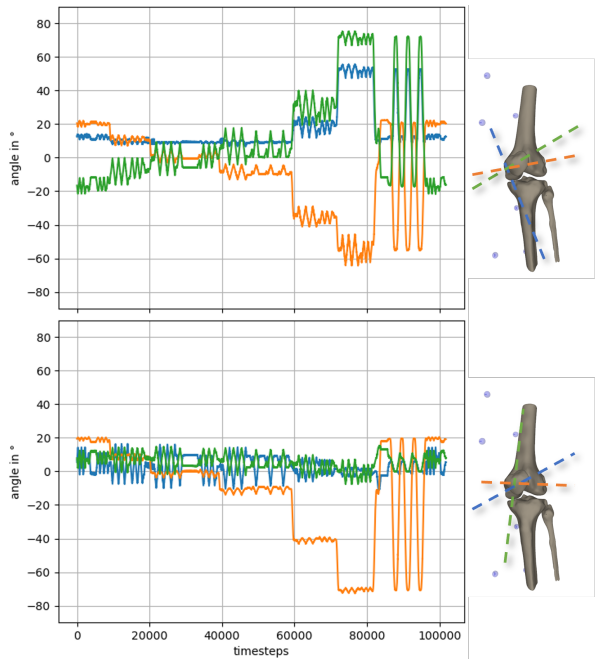

Fig. 6. Joint trajectory comparison for the 3-DOF knee joint using Wilhelm et al.'s data [26] sampled at 100 Hz. The top graph represents the AE-based method, while the bottom graph depicts the AE+ICA approach. Corresponding joint axes are color-coded on the right.

B. Knee Joint Analysis: Model Adaptability Across DOFs

- **1 DOF:** All models identified the knee joint's flexion axis as the primary kinematic feature, reinforcing their capability to discern critical biomechanical attributes.
- **2 DOF:** The AE model, despite its low error, derived nearly parallel axes, suggesting a potential trade-off between error minimization and anatomical accuracy.
- **3 DOF:** The AE+ICA model aligned with established biomechanical knowledge, identifying expected orthogonal axes. In contrast, the AE model's approach, while precise, might be less interpretable followed by the VAE variant with similar issues.

For models with multiple DOFs along a singular joint, the AE+ICA variant stands out by providing anatomical interpretability through its effective decoupling of joint axes. In contrast, while the AE variant excels in precision, it may compromise on interpretability. Although the VAE variant may not be the most interpretable in this context, it consistently exhibits low reconstruction errors. Its robustness is particularly evident for noisy datasets, attributed to its probabilistic treatment of both input and output spaces.

C. Joint Trajectories: AE vs. AE+ICA

The AE and AE+ICA methods' trajectory differences for the 3-DOF knee joint model highlight their distinct optimization strategies. The AE+ICA's differentiation between the flexion axis and other axes aligns with anatomical expectations. Conversely, the AE's trajectory coupling strategy, while effective in error reduction, may not always align with conventional biomechanical understanding.

D. Comparison with Existing Literature

The integration of autoencoders in our kinematic optimization study aligns with and diverges from prevailing literature in notable ways. Li et al. [27] highlighted the importance of precise trajectory planning in complex environments, a principle that is congruent with our objectives, even though their primary focus was on autonomous parking. Sundaralingam and Hermans [28] underscored the adaptability in kinematic trajectory optimization, a trait that resonates with the versatility observed in our AE and AE+ICA models. In contrast, Mari'c et al. [29] introduced a distinct Riemannian optimization method for the inverse kinematics problem, emphasizing a different approach but sharing the broader goal of addressing kinematic challenges. Li et al. [30], while operating in a disparate domain of mining truck trajectory planning, echoed the pervasive challenges in kinematic optimization. Our research offers novel insights, particularly the efficacy of ICA and VAE regularization techniques, suggesting potential avenues for refining kinematic optimization models.

E. Limitations

This study's primary limitation is its dependence on a singular validation dataset, potentially overlooking the variability inherent in joint movements across varied populations. Furthermore, inherent assumptions in the models, particularly concerning joint constraints and movement redundancies, may constrain their applicability to alternative datasets or practical contexts. It's essential to recognize that the optimized kinematic model in general only serves as a surrogate to the actual ground truth model, focusing primarily on minimizing reconstruction errors. A more comprehensive evaluation is warranted to contrast the derived joint parameters with the true parameters in established systems.

V. CONCLUSION

This research introduced a versatile autoencoder-based kinematic optimizer capable of concurrently solving both forward and inverse kinematics for diverse models and datasets using a predefined and differentiable structure for forward kinematics. The methodology demonstrated its efficacy by successfully identifying surrogate kinematic models with low reconstruction errors for two anatomically distinct joints: the hand and the knee. Notably, the optimizer aids in interpreting and identifying compromise axis, as evidenced by its performance with the knee joint under reduced degrees of freedom. The architecture, characterized by its simplicity and efficiency, stands as a notable contribution in the realm of kinematic optimization. The study's findings underscore the potential of this approach as a robust tool for biomechanics, robotics, and related disciplines, offering a promising avenue for future advancements in the field.

REFERENCES

[1] Peng Gao, Hyeonseung Lee, Chan-Woo Jeon, Changho Yun, Hak-Jin Kim, Weixing Wang, Gaotian Liang, Yufeng Chen, Zhao Zhang, and Xiongzhe Han. Improved position estimation algorithm of agricultural mobile robots based on multisensor fusion and autoencoder neural network. *Sensors*, 22(4), 2022.

[2] Guanwu Jiang, Minzhou Luo, Keqiang Bai, and Saixuan Chen. A precise positioning method for a puncture robot based on a PSO-optimized BP neural network algorithm. *Applied Sciences*, 7(10):969, September 2017.

[3] Raşit Köker, Cemil Öz, Tarık Çakar, and Hüseyin Ekiz. A study of neural network based inverse kinematics solution for a three-joint robot. *Robotics and Autonomous Systems*, 49(3):227–234, 2004. Patterns and Autonomous Control.

[4] Konstantinos D. Polyzos, Peter P. Groumpos, and Evangelos Dermatas. Solving the inverse kinematics of robotic arm using autoencoders. In Alla G. Kravets, Peter P. Groumpos, Maxim Shcherbakov, and Marina Kultsova, editors, *Creativity in Intelligent Technologies and Data Science*, pages 288–298, Cham, 2019. Springer International Publishing.

[5] Adrian-Vasile Duka. Neural network based inverse kinematics solution for trajectory tracking of a robotic arm. *Procedia Technology*, 12:20–27, 2014.

[6] Sandi Baressi Segota, Nikola Andjelic, Vedran Mrzljak, Ivan Lorencin, Ivan Kuric, and Zlatan Car. Utilization of multilayer perceptron for determining the inverse kinematics of an industrial robotic manipulator. *International Journal of Advanced Robotic Systems*, 18(4):172988142092528, July 2021.

[7] Jensen K. Henry, Jeffrey W. Hoffman, Jaeyoung Kim, Brett D. Steineman, Daniel R. Sturnick, Constantine A. Demetracopoulos, Jonathan T. Deland, and Scott J. Ellis. The impact of progressive collapsing foot deformity on foot &amp ankle kinematics and plantar pressure during simulated gait. *Foot and Ankle Orthopaedics*, 7(1):2473011421S0023, jan 2022.

[8] Xiyang Sun, Yingtao Liu, Chang Liu, Koichi Mayumi, Kohzo Ito, Akinao Nose, and Hiroshi Kohsaka. A neuromechanical model for drosophila larval crawling based on physical measurements. *BMC Biology*, 20(1), jun 2022.

[9] Takeyuki Ono, Ryosuke Eto, Junya Yamakawa, and Hidenori Murakami. Analysis and control of a stewart platform as base motion compensators - part i: Kinematics using moving frames. *Nonlinear Dynamics*, 107(1):51–76, November 2021.

[10] Guoxin Fang, Yingjun Tian, Zhi-Xin Yang, Jo M. P. Geraedts, and Charlie C. L. Wang. Efficient jacobian-based inverse kinematics with sim-to-real transfer of soft robots by learning. *IEEE/ASME Transactions on Mechatronics*, 27(6):5296–5306, December 2022.

[11] Franziska Meier, Austin Wang, Giovanni Sutanto, Yixin Lin, and Paarth Shah. Differentiable and learnable robot models, 2022.

[12] Lukas Mölschl, Jakob J. Hollenstein, and Justus Piater. Differentiable forward kinematics for tensorflow 2, 2023.

[13] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[14] Martin Kubovčík, Iveta Dirgová Luptáková, and Jiří Pospíchal. Signal novelty detection as an intrinsic reward for robotics. *Sensors*, 23(8):3985, April 2023.

[15] Masatoshi Nagano, Tomoaki Nakamura, Takayuki Nagai, Daichi Mochihashi, and Ichiro Kobayashi. Spatio-temporal categorization for first-person-view videos using a convolutional variational autoencoder and gaussian processes. *Frontiers in Robotics and AI*, 9, September 2022.

[16] Midhun M S and James Kurian. Task level disentanglement learning in robotics using vae. *International journal of electrical and computer engineering systems*, 13(7):561–568, September 2022.

[17] H Josephs, RL Huston, and K Anderson. Dynamics of mechanical systems. *Applied Mechanics Reviews*, 56(2):B22–B23, March 2003.

[18] Leo Dorst, Chris Doran, and Joan Lasenby, editors. *Applications of Geometric Algebra in Computer Science and Engineering*. Birkhäuser Boston, 2002.

[19] OptiTrack. Optitrack, 2023. Available online: `https://optitrack.com/` (accessed on 1 September 2023).

[20] Vicon. Vicon, 2023. Available online: `https://www.vicon.com/` (accessed on 1 September 2023).

[21] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[23] Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures. In Isabelle Guyon, Gideon Dror, Vincent Lemaire, Graham Taylor, and Daniel Silver, editors, *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, volume 27 of *Proceedings of Machine Learning Research*, pages 37–49, Bellevue, Washington, USA, 02 Jul 2012. PMLR.

[24] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4):411–430, 2000.

[25] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013.

[26] Nikolas Wilhelm, Constantin von Deimling, Sami Haddadin, Claudio Glowalla, and Rainer Burgkart. Validation of a robotic testbench for evaluating biomechanical effects of implant rotation in total knee arthroplasty on a cadaveric specimen. *Sensors*, 23(17), 2023.

[27] Bai Li, Tankut Acarman, Youmin Zhang, Yakun Ouyang, Cagdas Yaman, Qi Kong, Xiang Zhong, and Xiaoyan Peng. Optimization-based trajectory planning for autonomous parking with irregularly placed obstacles: A lightweight iterative framework. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):11970–11981, 2022.

[28] Balakumar Sundaralingam and Tucker Hermans. Relaxed-rigidity constraints: kinematic trajectory optimization and collision avoidance for in-grasp manipulation. *Autonomous Robots*, 43(2):469–483, June 2018.

[29] Filip Maric, Matthew Giamou, Adam W. Hall, Soroush Khoubyarian, Ivan Petrovic, and Jonathan Kelly. Riemannian optimization for distance geometric inverse kinematics. *CoRR*, abs/2108.13720, 2021.

[30] Bai Li, Yakun Ouyang, Xiaohui Li, Dongpu Cao, Tantan Zhang, and Yaonan Wang. Mixed-integer and conditional trajectory planning for an autonomous mining truck in loading/dumping scenarios: A global optimization approach. *IEEE Transactions on Intelligent Vehicles*, 8(2):1512–1522, 2023.