2026 IEEE/SICE International Symposium on System Integration (SII)
January 11-14, 2026. Cancún, México

Data-Driven Dynamic Parameter Learning of manipulator robots

Mohammed Elseiagy[1,3], Tsige Tadesse Alemayoh[2], Ranulfo Bezerra[2], Shotaro Kojima[2], Kazunori Ohno[2]

Abstract— **Bridging the sim-to-real gap remains a fundamental challenge in robotics, as accurate dynamic parameter estimation is essential for reliable model-based control, realistic simulation, and safe deployment of manipulators. Traditional analytical approaches often fall short when faced with complex robot structures and interactions. Data-driven methods offer a promising alternative, yet conventional neural networks such as recurrent models struggle to capture long-range dependencies critical for accurate estimation. In this study, we propose a Transformer-based approach for dynamic parameter estimation, supported by an automated pipeline that generates diverse robot models and enriched trajectory data using Jacobian-derived features. The dataset consists of 8,192 robots with varied inertial and frictional properties. Leveraging attention mechanisms, our model effectively captures both temporal and spatial dependencies. Experimental results highlight the influence of sequence length, sampling rate, and architecture, with the best configuration (sequence length 64, 64 Hz, four layers, 32 heads) achieving a validation R^2 of 0.8633. Mass and inertia are estimated with near-perfect accuracy, Coulomb friction with moderate-to-high accuracy, while Damping coefficient and distal link center-of-mass remain more challenging. These results demonstrate that combining Transformers with automated dataset generation and kinematic enrichment enables scalable, accurate dynamic parameter estimation, contributing to improved sim-to-real transfer in robotic systems.**

I. Introduction

Robotic arms are essential in today's industries and research, carrying out tasks such as precision manufacturing, surgical support, and exploration of dangerous environments [1]. The performance and safety of these systems depend directly on the accuracy of their dynamic models, which are defined by parameters such as mass, inertia, friction, and damping. Estimating these dynamic parameters correctly is crucial for enabling advanced robotic functions including model-based control, realistic simulation, digital twins, and sim-to-real transfer [2].

One of the main challenges in robotics is the reality gap in sim-to-real transfer. Training control policies or reinforcement learning agents directly on physical robots is often difficult due to safety risks, limited time, and hardware costs. Simulations provide a safe and efficient alternative, but controllers trained in simulation often fail in the real world because of differences between simulated and real dynamics [3], [4], [5]. Accurate dynamic parameter estimation is crucial to close this gap. By carefully identifying a robot's physical properties, we can build simulations that closely match real physics, making sim-to-real transfer more reliable and enabling advanced control and learning methods on real systems [6], [7], [8]. Similarly, accurate models form the basis of digital twins—real-time virtual copies of physical systems used for monitoring, predictive maintenance, and optimization [9], [10], [11], [12].

Despite its importance, estimating these parameters remains challenging. Traditional analytical methods, such as least-squares estimation, often struggle with the complex and coupled dynamics of multi-joint arms [1]. These methods are sensitive to noise and can produce results that are not physically meaningful [2]. Moreover, dynamic parameters can change over time because of wear, temperature shifts, or changes in payload, requiring frequent re-calibration. Data collection in real robots adds further difficulty due to sensor noise and the challenge of obtaining accurate measurements.

To overcome these issues, data-driven methods, especially those based on deep learning, have become strong alternatives. In this context, data-driven refers to learning system dynamics directly from collected kinematic and dynamic information, including joint positions, velocities, accelerations, and applied joint torques during robot motion. These measurements exhibit strong temporal dependencies, as the current state of the robot is inherently influenced by its previous states and the history of applied forces. Capturing these temporal relationships is crucial for accurate parameter estimation. While Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks have shown success in capturing time-based patterns for dynamic parameter identification [13], [14] and dynamics modeling [15], recurrent architectures are known to struggle with very long-range dependencies due to vanishing gradients and sequential processing constraints, which can limit their generalization across diverse robot configurations.

To address these challenges and push forward dynamic parameter estimation, we propose a complete framework with three main phases. First, in the dataset generation phase, dynamically varying simulated manipulators are executed to follow pre-defined trajectories in Gazebo. To enrich the dataset, Jacobian matrices are incorporated as additional features. Finally, in the training phase, different classical and transformer-based models are trained for dynamic parameter estimation and their results are analyzed. By simulating a wide range of conditions—including changes in friction, damping, and inertia—our method aims to generalize across diverse robot configurations. The scope of this study is lim-

This research was performed by the commissioned research fund provided by F-REI (JPFR23010101).

[1] Graduate School of Information Sciences, Tohoku University, Sendai, Japan.
[2] Tough Cyberphysical AI Research Center, Tohoku University, Sendai, Japan.
[3] Egypt-Japan University of Science and Technology (E-JUST), New Borg El-Arab, Alexandria, Egypt.
mohammed.abelaziz@ejust.edu.eg
https://github.com/MohamedAlsiagy/dynamic_parameter_est

ited to simulation-based evaluation, focusing on establishing a robust and scalable methodology for dynamic parameter estimation within a controlled simulation framework, providing a foundation for subsequent real-world deployment.

The main contributions of this work are:

- An automated framework with multiple pipelines for generating diverse robot models and trajectory data in simulation, incorporating gravity-aware PID control to produce rich datasets for parameter estimation.
- A comprehensive comparison of multiple neural network architectures (MLP, RNN, LSTM, TCN, and Transformer) with equivalent parameter counts to identify the most suitable model for dynamic parameter estimation, demonstrating that Transformers achieve superior generalization performance.
- An ablation study investigating the impact of kinematic feature enrichment, revealing that Jacobian matrix elements provide the most compact and informative representation for parameter estimation, outperforming the raw dataset and combined features including positions, quaternions, and velocities.

II. Related Work

The problem of estimating robot dynamic parameters has been studied from classical mechanics to modern deep learning. This section places our work in this context, showing the progress of approaches and clarifying our unique contribution.

A. Model-Driven Identification Methods

Traditional model-driven approaches are based on rigid body dynamics, often expressed with Euler-Lagrange equations. Weighted least-squares (WLS) methods are common, where carefully designed excitation trajectories are used to estimate base parameters [1]. While effective for simple systems, these methods face challenges including the need for specialized excitation trajectories, high sensitivity to noise, and difficulty modeling complex non-linear effects such as friction [2]. Grey-box methods, which combine physical models with data-driven techniques, have recently been proposed to improve robustness [16], especially for digital twins, but they still rely on simplified friction models or large experimental setups.

B. Data-Driven Methods for Robot Dynamics

With deep learning, purely data-driven methods have become strong alternatives. They can learn dynamics directly from sensor data, removing the need for explicit physical equations. Early work used feed-forward networks, but the sequential nature of motion data led to RNNs and LSTMs. For example, [13] and [14] used LSTMs for robot dynamics learning, showing they could capture time-based patterns. [15] studied machine learning methods based on synthetic data for dynamics modeling. However, recurrent architectures have inherent limitations in capturing very long-range dependencies, which can affect their generalization across different robots and conditions.

C. Transformers in Robotics and Time-Series Analysis

The Transformer [17] has recently proven powerful for sequence modeling beyond language, due to its self-attention mechanism. This mechanism allows it to consider all past states when predicting the next, overcoming the limitations of RNNs. In robotics, Transformers have been used for state prediction and dynamics modeling [18]. Recent research has adapted Transformers for multivariate time-series, with new position encoding methods especially useful for robotic data [19], [20]. More broadly, meta-learning with neural networks has been explored to quickly adapt models to new robots with little data, aligning with our goal of generalizable parameter estimation [21]. Our work differs by focusing specifically on estimating physical parameters and by introducing an automated data generation and enrichment pipeline. This places our research at the intersection of deep learning and robot system identification, aiming for a scalable and reliable solution for tasks like sim-to-real transfer and adaptive control.

III. Methodology

This research uses a multi-stage framework consisting of multiple pipelines (Figure 1) to generate, simulate, preprocess, and analyze robotic data with deep learning. The core objective is to create a dataset of robots with maximally diverse dynamic parameters to enable robust generalization of the learned parameter estimation model. The pipeline starts with the Robot Generator, which systematically varies inertial and frictional properties to create a diverse set of *R Robots* by generating 3D meshes with different geometries, computing their corresponding inertial parameters, and assembling URDF models. This diversity in dynamic parameters—including mass distribution, friction coefficients, and inertia tensors—is essential for training models that can generalize across different robot configurations. In parallel, the Trajectory Generator produces *N waypoints* per robot. Each waypoint is validated to avoid collisions by solving the forward kinematics, and interpolation ensures that intermediate points are also collision-free. These robots and trajectories are then integrated into a ROS–Gazebo simulation for realistic physical data collection. The simulated data undergoes Dataset Preprocessing, including kinematic enrichment, feature filtering, timestep unification, and sampling with caching. The final preprocessed dataset is then used for Deep Learning to extract insights and train predictive models capable of estimating dynamic parameters across diverse robotic systems.

A. URDF Generator

To construct the dataset, a custom URDF generator script was developed to produce large collections of robots. For each robot instance, the kinematic configuration (link lengths, joint axes, and joint offsets) was held constant. However, the following structural and dynamic properties were systematically varied: link cross-section shape, link diameter, link center of mass position, joint Coulomb friction coefficients, and joint Damping coefficients. These variations

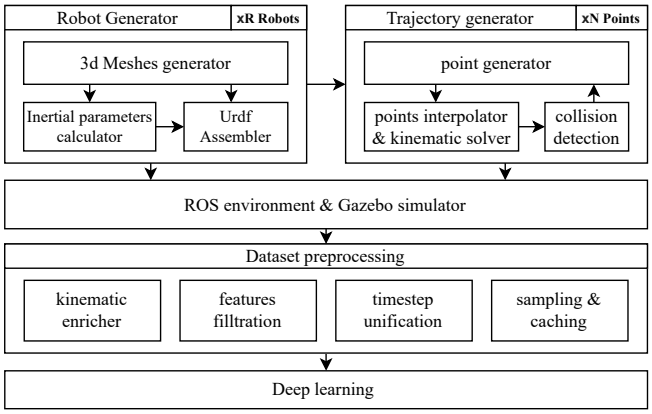

Fig. 1: The multi-stage pipeline for robotic data generation, simulation, preprocessing, and deep learning analysis.

directly influence the robots' inertial and frictional dynamics while preserving kinematic similarity. A total of 8,192 robots were generated using this approach, examples of which are shown in Figure 2.

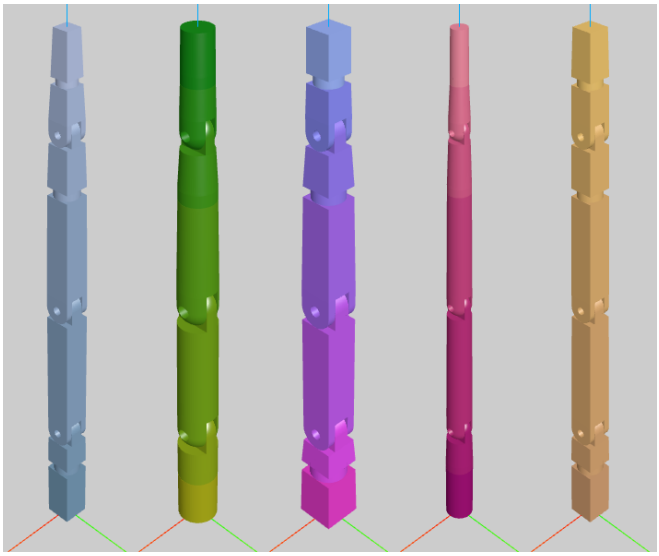

Fig. 2: Examples of robots generated by the URDF script. All robots share the same kinematic configuration (link lengths, joint axes, and offsets) but differ in their inertial and frictional dynamics due to changes in link cross-section, diameter, center of mass, and joint friction coefficients.

All robots share the same kinematic configuration: a 6-DOF serial manipulator with 7 links (including base L_0), all revolute joints, and joint axes Z-Y-Y-Z-Y-Z from base to end-effector. Link lengths are 15, 7.5, 30, 25, 10, 7.5, and 10 cm respectively, yielding approximately 105 cm total reach. The kinematic structure remains constant while dynamic properties vary due to changes in link cross-sections, diameters, mass distributions, and friction coefficients.

B. Dynamic Parameters

The dynamic parameters considered in this work are:

- **Coulomb Friction**: This force opposes motion and is modeled as:

$$F_{\text{Coulomb}} = \mu_c \cdot \text{sign}(\dot{q}),$$

  where μ_c is the Coulomb coefficient and $\dot{q}$ is the joint velocity.
- **Damping**: This resistance is proportional to velocity and is modeled as:

$$F_{\text{Damping}} = \mu_v \cdot \dot{q},$$

  where μ_v is the damping coefficient.
- **Inertia Matrix**: This represents the mass distribution of each link and its resistance to acceleration.

These parameters were selected because they are challenging to measure directly and are susceptible to variations caused by environmental factors (e.g., temperature, humidity) and mechanical wear. In contrast, parameters such as link mass are relatively straightforward to measure and remain stable. The manipulator's dynamics, including frictional effects, can be expressed by the following equation:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F_{\text{friction}} = \tau,$$

where $M(q)$ is the inertia matrix, $C(q,\dot{q})$ represents the Coriolis and centrifugal forces, $G(q)$ is the gravity vector, τ denotes the applied joint torques, and the total friction force F_{friction} is defined as:

$$F_{\text{friction}} = \mu_c \cdot \text{sign}(\dot{q}) + \mu_v \cdot \dot{q}.$$

Geometric: Link diameters $d \in [6, 12]$ cm with circular or square cross-sections. Mass/Inertia: Calculated from STEP geometry assuming aluminum shells ($\rho = 600$ kg/m^3), yielding 0.1–2.5 kg per link.Center of Mass: Biased due to randomly setting radius difference between the base of the link and the tip , this difference is chosen randomly between [-0.06,0] cm. Friction: Coulomb coefficients $\mu_{c,i} \sim \mathcal{U}(0, 0.5) \times (0.625)^i$ with exponential decay from base to distal joints; Damping coefficients $\mu_v \in [0.0, 0.1]$ N·m·s·rad^{-1} uniformly per joint. These ranges produce 8,192 robots with diverse, physically plausible dynamics.

C. Trajectory Generation

For each robot, 16 workspace waypoints were randomly selected. A per-joint PID controller was then utilized to guide the manipulator toward these waypoints. The controller was specifically designed and calibrated to achieve smooth and rapid transitions while preventing overshoot.

The simulator was configured to log raw data (joint positions, velocities, and applied joint torques) at a frequency of 1000 Hz. To enhance robustness, the script incorporated anti-failure detection mechanisms to identify and automatically discard corrupted data resulting from collisions, simulator instabilities, or timeouts.

A standard PID controller without gravity compensation was initially used. However, to mitigate gravitational effects and improve control smoothness, a novel gravity-aware PID modification was introduced:

$$u(t) = \left(K_p + K_G \frac{dj_i}{dz}\right) e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{de(t)}{dt}, \tag{1}$$

where K_G is a constant tuned to help the joint overcome gravity, and $\frac{dj_i}{dz}$ is an element derived from the Jacobian matrix that indicates the effect of the motion of the i-th joint in the z-direction (the direction of gravity).

This modification introduces a proportional torque adjustment based on the gravitational influence of each joint, leveraging Jacobian elements. The resulting hybrid compensation results in improved stability in trajectory tracking, while maintaining a lightweight controller design.

D. Post-Processing and Dataset Composition

Since the raw dataset was collected at 1000 Hz, it was too dense for direct use in model training. To make learning tractable, the trajectories were downsampled by selecting data at a smaller, fixed sampling frequency, thereby compressing the temporal information into manageable sequences. However, naive downsampling discards a significant portion of the collected dataset. To address this, a secondary offset-based sampler was implemented. This method extracts multiple sequences at the same sampling rate but with uniformly stepping temporal offsets, effectively reusing the raw data more efficiently.

Additionally, the dataset was augmented with kinematic features:

- **Jacobian Matrix Elements**: capturing the influence of joint motion on link positions. Its effect is shown in results section.
- **Pruned Redundancies**: removal of constant or irrelevant terms (e.g., translational components for link 1 that only rotates).

E. Final Dataset Composition

The final dataset comprises:

- **8,192 robots** with varied inertial and frictional properties.
- **16 trajectories per robot**, defined by randomly chosen workspace waypoints.
- Augmented kinematic features (filtered Jacobian elements per link).

This methodology ensures that the dataset captures a wide spectrum of dynamic behaviors.

F. Transformer-Based Model Architecture

The proposed model adopts the original transformer architecture introduced by Vaswani et al. [17], initially developed for natural language processing (NLP) and subsequently extended to sequential data tasks. Transformers are particularly effective for robotic time-series data due to their ability to capture long-range dependencies and temporal relationships, making them well-suited for dynamic parameter estimation. Our model consists of three key components: input embedding, joint-specific transformers, and output decoding.

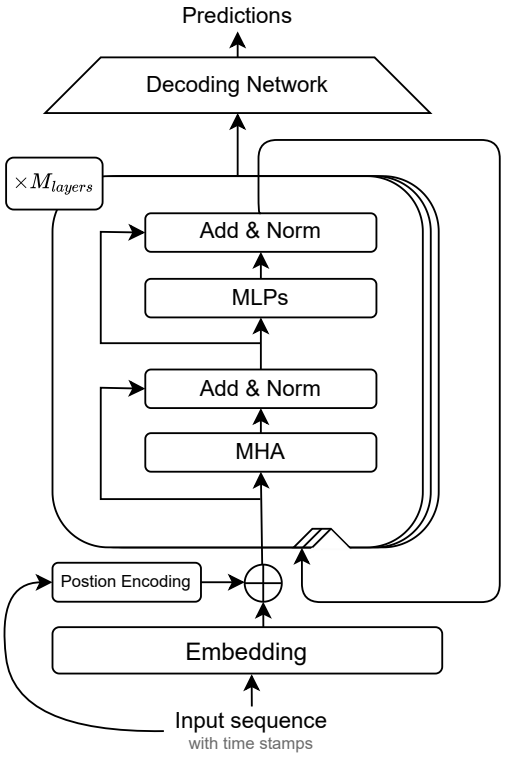

Fig. 3: The architecture of the proposed model, based on the original transformer [17].

Input features, including joint positions, velocities, torques, and kinematic elements, are embedded into a fixed-dimensional space for transformer processing. Adhering to the original transformer design [17], positional encoding is used to represent temporal information, rather than specialized encodings for irregular time series. To ensure compatibility, the raw dataset (originally sampled at 1000 Hz) was resampled onto a fixed frequency grid using interpolation, thereby enabling the use of standard positional encodings.

While alternative encoding schemes such as tAPE and eRPE [19] have been proposed to better represent time in irregularly sampled sequences, and methods like [20] directly address irregular sampling using attention mechanisms, our focus was on maintaining fidelity to the original transformer formulation. Therefore, preprocessing steps (interpolation and resampling) were applied to regularize the data before input, preserving compatibility with the canonical architecture while still capturing essential temporal dynamics.

IV. Results

This section evaluates the proposed approach across dataset configuration, architecture choices, baseline comparisons, kinematic enrichment, and parameter-wise performance. The optimal model achieves validation R^2 of 0.8633.

A. Effect of Dataset Configuration

Table I shows how sequence length, sampling rate, and secondary sampling rate affect performance. Increasing sequence length improves performance up to 64 steps.

TABLE I: Experimental study on dataset configurations using fixed transformer architecture. SSR is secondary sampling rate.

Seq Len	Sampling Rate	SSR	Time (s)	R^2	RMSE
16	32	8	0.512	0.7928	0.1379
16	64	16	1.024	0.8160	0.1299
16	128	32	2.048	0.8357	0.1226
16	256	32	4.096	0.8488	0.1174
32	32	8	1.024	0.8134	0.1308
32	64	16	2.048	0.8428	0.1199
32	128	128	16.384	0.8507	0.1167
64	32	8	2.048	0.8328	0.1237
64	32	16	2.048	0.8331	0.1236
64	64	16	4.096	**0.8633**	**0.1116**
64	64	32	4.096	0.8592	0.1133
64	64	64	4.096	0.8571	0.1141
128	16	8	2.048	0.8181	0.1290
128	32	8	4.096	0.8451	0.1188
128	32	16	4.096	0.8459	0.1185
128	64	16	8.192	0.8479	0.1173

B. Transformer Architecture Comparison

Table II compares models with varying layers, attention heads, and embedding dimensions. The optimal configuration—4 layers, 32 heads, 128 embedding dimension—achieved the highest validation R^2 of 0.8633.

TABLE II: Transformer architecture comparison on a fixed dataset.

Layers	Heads	Embedding Dim	R^2	RMSE
2	16	256	0.8460	0.1185
4	16	128	0.8555	0.1148
4	32	128	**0.8633**	0.1116
8	32	128	0.8410	0.1200

C. Comparison with Baseline Models

Table III compares the transformer against baselines with similar parameter counts (410K-416K) on the Jacobian-enriched dataset. The Transformer achieved the lowest validation RMSE of **0.1116**, representing a 19% error reduction compared to LSTM, 17% over TCN, 30% over RNN, and 31% over MLP. While TCN achieved second-best performance, it required substantially longer training time (34.5 vs 5.0 minutes per epoch).

TABLE III: Comparison of the proposed transformer model against baseline architectures with comparable parameter counts.

Model	Trainable Parameters	R^2	Time/Epoch (mins)
MLP	415,466	0.7114	1.0
TCN	409,894	0.8010	34.5
RNN	411,942	0.7193	2.0
LSTM	411,942	0.7909	1.7
Transformer (Ours)	413,990	**0.8606**	5.0

D. Effect of Kinematic Enrichments

Table IV presents an ablation study on different feature combinations: J (Jacobian), P&Q (positions and quaternions), V&AV (velocities), and JP (joint positions). The Jacobian-only dataset consistently achieved the highest final validation R^2 (0.8606) and fastest convergence. Adding more features to the Jacobian did not yield improvements and occasionally led to degradation.

TABLE IV: Ablation study on kinematic dataset enrichments (Sorted by binary representation ✓=1, ✗=0 for J, P&Q, V&AV, JP).

J	P&Q	V&AV	JP	@13	@25	@38	@50 (Final)
✗	✗	✗	✗	0.7853	0.8026	0.8179	0.8281
✗	✗	✗	✓	0.7907	0.8220	0.8401	0.8529
✗	✗	✓	✗	0.7883	0.8024	0.8151	0.8275
✗	✗	✓	✓	0.7844	0.8145	0.8284	0.8359
✗	✓	✗	✓	0.7941	0.8226	0.8387	0.8493
✗	✓	✓	✓	0.7871	0.8197	0.8353	0.8408
✓	✗	✗	✗	**0.8124**	**0.8423**	**0.8496**	**0.8606**
✓	✗	✗	✓	0.8123	0.8360	0.8488	0.8513
✓	✗	✓	✗	0.8002	0.8227	0.8416	0.8501
✓	✗	✓	✓	0.8002	0.8274	0.8412	0.8427
✓	✓	✗	✓	0.8113	0.8358	0.8494	0.8563
✓	✓	✓	✓	0.8024	0.8275	0.8428	0.8478

E. Parameter Estimation Results

Tables V-VII show detailed results. Mass and inertia achieve near-perfect accuracy (RMSE < 0.03). Coulomb friction varies by joint (RMSE 0.08–0.20), with distal joints performing better. Damping is most challenging (RMSE 0.18–0.31). Distal link COM estimation remains difficult due to coupling effects.

TABLE V: Friction parameter estimation results.

Joint	Coulomb Friction		Damping	
	R2	RMSE	R2	RMSE
J_0	0.8279	0.1211	0.2138	0.2582
J_1	0.4479	0.1965	-0.1945	0.3003
J_2	0.5834	0.1873	-0.1333	0.3060
J_3	0.8419	0.1167	0.6521	0.1753
J_4	0.6927	0.1674	0.5949	0.1789
J_5	0.9101	0.0796	0.8817	0.0939

TABLE VI: Mass and center of mass (COM) parameter estimation results.

Link	Mass		COM	
	R2	RMSE	R2	RMSE
L_2	0.9713	0.0370	0.8624	0.0153
L_3	0.9869	0.0242	0.6876	0.0201
L_4	0.9865	0.0220	0.6699	0.0307
L_5	0.9834	0.0235	0.6478	0.0378
L_6	0.9778	0.0250	0.2440	0.0323

TABLE VII: Inertia matrix parameters estimation results.

Link	Ixx		Iyy		Izz	
	R2	RMSE	R2	RMSE	R2	RMSE
L_1	-	-	-	-	0.9215	0.0285
L_2	0.9720	0.0326	0.9716	0.0328	0.9559	0.0055
L_3	0.9861	0.0222	0.9860	0.0225	0.9767	0.0058
L_4	0.9802	0.0231	0.9810	0.0218	0.9814	0.0089
L_5	0.9777	0.0216	0.9770	0.0232	0.9658	0.0137
L_6	0.9750	0.0226	0.9749	0.0226	0.9688	0.0191

V. Discussion

Why Jacobian Features Work Best. The Jacobian directly encodes the relationship between joint torques and motion. When using joint positions and torques as inputs, the Jacobian provides the missing link: how each joint's motion affects configuration and torque requirements. Adding extra features doesn't help because the network must re-learn these

relationships, whereas the Jacobian gives this information explicitly.

Parameter Estimation Accuracy. *Mass and Inertia* (RMSE < 0.03) succeed because they directly affect joint torques during acceleration. *Coulomb Friction* (RMSE 0.08–0.20) varies by joint—distal joints perform better due to more frequent motion with less load variation. *Damping* (RMSE 0.18–0.31) is most difficult because its velocity-dependent contribution is small compared to inertia and gravity. *Distal Link COM* (L_6 RMSE 0.32) is challenging because distal links are affected by all proximal joints, making many COM positions produce similar torque patterns.

Implications. For well-estimated parameters (mass, inertia, proximal COM), the current approach is sufficient. For Coulomb friction in proximal joints, adding constant-velocity trajectories would help. For Damping and distal COM, data-driven estimation alone may be insufficient—these could instead use physical priors or dedicated identification procedures.

Limitations and Future Work. This study was conducted entirely in simulation. Real-world validation is the critical next step to determine sim-to-real transfer. Future priorities include: (i) testing on physical robots, (ii) generating trajectories to improve observability of difficult parameters, (iii) incorporating physical constraints into the architecture, and (iv) extending to more diverse robot morphologies.

VI. Conclusion

Accurate dynamic parameter estimation for robotic manipulators is critical for closing the sim-to-real gap, yet traditional analytical methods struggle with complex dynamics while recurrent neural networks fail to capture long-range temporal dependencies across diverse configurations.

We addressed this through a three-phase approach: (i) automated generation of 8,192 diverse robot models executing 16 trajectories each using gravity-aware PID control, (ii) dataset enrichment with Jacobian matrix elements as the most informative features, and (iii) transformer-based architecture with attention mechanisms for temporal-spatial dependency modeling.

Our best configuration (64 sequence length, 64 Hz, 4 layers, 32 heads) achieved validation R^2 of 0.8633 (RMSE 0.1116)—19% error reduction vs. LSTM and 17% vs. TCN. Mass and inertia achieved near-perfect accuracy (RMSE $<$ 0.03), Coulomb friction moderate-to-high accuracy (RMSE 0.08–0.20), while Damping (RMSE 0.18–0.31) and distal COM remain challenging due to weak torque signatures and kinematic coupling.

This study's simulation-only scope is a critical limitation. Future work must prioritize: (i) physical robot validation to assess sim-to-real transfer, (ii) excitation trajectories improving observability of difficult parameters, (iii) incorporating physical constraints into network architecture, and (iv) testing on diverse morphologies beyond 6-DOF. Despite these limitations, our method demonstrates strong potential for scalable parameter estimation in robotic systems.

References

[1] Huapeng Wu, Yaxin Lu, Zhaojie Li, Jing Zhao, and Loris Roveda. A review of dynamic parameters identification for manipulator control. *Cobot*, 1(1):1–17, 2022.

[2] Taeyoon Lee, Jaewoon Kwon, Patrick M Wensing, and Frank C Park. Robot model identification and learning: A modern perspective. *Annual Review of Control, Robotics, and Autonomous Systems*, 7:311–334, 2024.

[3] Jakob Jonas Rothert, Sebastian Lang, and Magnus Hanses. Sim-to-real transfer for a robotics task: Challenges and lessons learned. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024.

[4] W. Zhao, J. P. Queralta, and T. Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. *2020 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–7, 2020.

[5] J. Lu, F. Richter, and M. C. Yip. Pose estimation for robot manipulators via keypoint optimization and sim-to-real transfer. In *2022 IEEE Robotics and Automation Letters*, volume 7, pages 3709–3716, 2022.

[6] D. Bargellini. Sim2real transfer for reinforcement learning in robotic arm control: a closed-loop optimization approach for parameter estimation. 2023.

[7] J. Ren, Y. Xu, Y. Wang, X. Chen, and H. Li. Adaptsim: Task-driven simulation adaptation for sim-to-real transfer. *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems*, pages 1786–1794, 2023.

[8] S. Garg, A. Kumar, and A. Gupta. Dynamics as prompts: In-context learning for sim-to-real system identification. *arXiv preprint arXiv:2410.20357*, 2024.

[9] Senthamarai Kannan Chinnasamy, Hari Prasaanth Sura, Amanullah Saleem, Akash Kathirvel, and Prashanna Rangan. Digital twin of robot manipulator using ROS. *AIP Conference Proceedings*, 2864(1), 2023.

[10] Y. Liu, X. Zhang, and Y. Wang. A hybrid digital twin scheme for the condition monitoring of robotic manipulators. *Procedia Computer Science*, 232:124–133, 2024.

[11] L. Chen, H. Wang, and J. Li. Designing digital twins of robots using simscape multibody. *Sensors*, 23(4):62, 2023.

[12] Y. Wang, X. Li, and X. Zhang. Research on parameter compensation method and control strategy of mobile robot dynamics model based on digital twin. *Sensors*, 24(24):8101, 2024.

[13] Shoujun Wang, Xingmao Shao, Liusong Yang, and Nan Liu. Deep learning aided dynamic parameter identification of 6-dof robot manipulators. *IEEE Access*, 8:138102–138116, 2020.

[14] A. S. Polydoros and L. Nalpantidis. Real-time deep learning of robotic manipulator inverse dynamics. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1098–1103, 2015.

[15] S. Baressi Šegota, N. Anđelić, M. Šercer, and H. Meštrić. Dynamics modeling of industrial robotic manipulators: A machine learning approach based on synthetic data. *Mathematics*, 10(7):1174, 2022.

[16] X. Li, Y. Wang, and X. Zhang. Evaluating the use of grey-box system identification for digital twins of robotic manipulators. *International Journal of Advanced Robotic Systems*, 21(1):1–10, 2024.

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[18] Alec Reed, Doncey Albin, Anuh Pasricha, Alessandro Roncone, and Christoffer Heckman. Transformer-based learning models of dynamical systems for robotic state prediction, 02 2024.

[19] Navid Mohammadi Foumani, Chang Wei Tan, Geoffrey I. Webb, and Mahsa Salehi. Improving position encoding of transformers for multivariate time series classification. *Data Mining and Knowledge Discovery*, 38(1):22–48, September 2023.

[20] Satya Narayan Shukla and Benjamin Marlin. Multi-time attention networks for irregularly sampled time series. In *International Conference on Learning Representations*, 2021.

[21] Manuel Bianchi Bazzi, Asad Ali Shahid, Christopher Agia, John Alora, Marco Forgione, Dario Piga, Francesco Braghin, Marco Pavone, and Loris Roveda. RoboMorph: In-context meta-learning for robot dynamics modeling. *arXiv preprint arXiv:2409.12345*, 2024.