# Exploration of Neural Network Architectures for Inertia Parameter Identification of a Robotic Arm

Thomas Granser, Maximilian Giessler, Stefan Glaser, Bernd Waltersberger and Stefan Hensel

Offenburg University of Applied Sciences
{thomas.granser, maximilian.giessler, stefan.glaser, bernd.waltersberger, stefan.hensel}@hs-offenburg.de

**Abstract.** We propose a machine learning based approach for identifying inertia parameters of robotic systems. We evaluate the method in simulation and compare it against classical methods. Specifically, we implement parameter identification based on numerical optimization and test it using ground truth data. For a case study, we set up a physical simulation of a four-degree-of-freedom robot arm, formulating the problem with Newton-Euler equations as opposed to the conventional Lagrangian formulation at the joint level. Additionally, we derive a test methodology for assessing various Artificial Neural Network architectures.

**Keywords:** Inertia parameters identification, robotics, numerical optimization, Newton-Euler, Neural Networks

## 1 Introduction

Inertia parameter identification is essential in robotics for precise motion planning and control [1]. The actual inertia parameters of robots often deviate from those predicted by CAD models due to unmodeled components, production tolerances, or modifications introduced during manufacturing [2]. Several methods have been proposed for inertia parameter identification, as reviewed by Leboutet et al. [3]. The most widely used approach involves modeling the system with an inverse dynamic model, measuring joint torques, and using motor encoder signals to obtain the system's kinematic parameters at specific time points while the robot follows an excitation trajectory.

Traditionally, the equations of motion (EOM) are derived from the Lagrangian formulation, yields linear equations in terms of kinematic parameters (angular position, velocity, and acceleration) at the joint level. However, this method depends heavily on motor encoder signals to estimate dynamic parameters, making it susceptible to noise and errors from numerical differentiation when calculating joint velocities and accelerations. Additionally, the conventional approach relies on torque values from the joint motors, which are often indirectly measured via electrical current and voltage. Since direct torque measurements are often unavailable, accounting for nonlinear friction, thermal losses, and electromagnetic effects becomes challenging, limiting the practical application of these measurements in real-world scenarios [1].

## 2 Background

In the context of *inertia parameter identification* for a multi body-system of $n$ rigid links connected by actuated rotational joints, the classical approach utilizes the Lagrangian formalism to construct an inverse dynamic model. This model is represented by a set of $n$ equations for the generalized forces — specifically the joint torques ($\tau_i, i = 0..n$) of the joints connecting the links:

$$\boldsymbol{\tau}(t) = f_i(q_i(t), \dot{q}_i(t), \ddot{q}_i(t)), \tag{1}$$

This equation can be expressed as a direct relationship between the system's kinematic parameters $\mathbf{Y}(q_i(t), \dot{q}_i(t), \ddot{q}_i(t))$ and the inertia parameters of the system $\boldsymbol{\theta}$:

$$\boldsymbol{\tau}(t) = \mathbf{Y}(q_i(t), \dot{q}_i(t), \ddot{q}_i(t))\boldsymbol{\theta}, \tag{2}$$

with $i$=$0..n$. Here $\boldsymbol{\theta}$ represents the ten unknown inertia parameters for each link. These parameters are comprised by the mass $m_i$, the three first moments (mass times the center of mass vector) $m \cdot r_{\mathrm{com}[x|y|z],i}$ and the six components of the inertia tensor for each of the $n$-links. The first moments thereby represent a nonlinear relation between the masses and center of mass vectors of the links. The first moments introduce a non-linear relationship between the mass and the center of mass vectors of the links, making inertia parameter optimization an inherently nonlinear problem. Additionally, selecting appropriate excitation trajectories poses another challenge, as it presents an optimization problem in itself, as discussed by Lee et al. [4]. For serial robotic systems, such as robotic arms, issues related to non-, low-, or linked-observability of certain inertia parameters can arise, depending on the system's kinematics and excitation. This may result from inadequate excitation or from the nonlinear relationship between the first moments and the weak influence of the off-diagonal elements of the inertia tensor, depending on the robot's configuration [5]. Furthermore, the single-axis actuation of the initial links reduces the system's ability to observe dynamic effects such as Coriolis and centrifugal forces, which are critical for estimating the complete set of inertia parameters. As a result, parameters like the off-diagonal inertia tensor components $I_{xy}, I_{xz}, I_{yz}$ and certain components of the center of mass vectors $\mathbf{r}_{\mathrm{com}}$ often remain unidentifiable [5].

## 3 Methodology

In contrast to the traditional approach, we propose using the Newton-Euler equations in an inertial frame, leveraging direct measurements of angular velocities and accelerations with a newly developed sensor concept at Offenburg University [6, 7]. Additionally, we employ an external force-torque measurement unit mounted at the robot's base to measure the total forces and moments resulting from the system's motion on the robot's fixture. By directly measuring both the kinematic parameters and the reaction forces and moments at the robot's base, we eliminate the need to model complex joint-motor interactions, such as nonlinear friction. This approach also avoids the requirement for numerical differentiation or integration of sensor data. The Newton-Euler equations can be derived using either the recursive Newton-Euler Algorithm or the general formulation in a non-inertial frame for a multi-body system, as described by Giessler et al. [8, 7]. The resulting equations describe the relationship between the motion of the entire system and the resulting forces and moments at the robot's base, rather than the traditional equations of motion (EOM) on joint level.

This approach yields a set of six equations for any given time, compared to the $n$ equations produced by the EOM at the joint level, where $n$ equals the number of joints in the robot. Similar to the Lagrangian formulation, these equations place the forces and moments on the left-hand side, while the right-hand side contains a function representing the interaction between the system's kinematics and inertia parameters [9].

$$\boldsymbol{F}_{\text{ext}} = \xi(q_i(t), \dot{q}_i(t), \ddot{q}_i(t))\boldsymbol{\theta}, \tag{3}$$
$$\boldsymbol{M}_{\text{ext}} = \phi(q_i(t), \dot{q}_i(t), \ddot{q}_i(t))\boldsymbol{\theta}, \tag{4}$$

where $\mathbf{F}_{\text{ext}}$ represents the three external forces and $\mathbf{M}_{\text{ext}}$ represents the three moments measured by the force-torque sensor.

Our contribution consists of two distinct solutions: first, classical numerical optimization; and second, the exploration and development of a machine learning approach employing neural networks. Both methods are evaluated on a synthetic ground-truth dataset, which includes inertia parameters and dynamic states (excitation frames) for various robotic configurations. Using the Newton-Euler equations, we compute the reaction forces and moments corresponding to the movements of a robot configuration.

We perform numerical optimization using solvers for both nonlinear and linearized problems, with the optimization results serving as benchmarks for our artificial intelligence (AI) methods. The AI approach is based on various fully connected feed-forward neural network architectures, which are systematically tested in different configurations. Additionally, we designed a more sophisticated AI approach using Siamese network architectures. Finally, we tested a custom loss function that incorporates physical constraints by embedding the analytical equations of the system into the loss function.

### 3.1 Data Generation

Identifying the inertia parameters of the initial links – such as the base and shoulder – in robotic arms, particularly serial manipulators, presents a specific challenge. These links are typically actuated along only a single axis, resulting in limited excitation of the system's dynamics. This limitation makes it difficult to fully excite the degrees of freedom necessary for identifying all inertia parameters, especially the off-diagonal terms of the inertia tensor and the components of the center of mass vector.

In this work, we use the four link, four degrees-of-freedom robotic arm of the humanoid robot *Sweaty* from Offenburg University as a case study to evaluate the presented approaches. For this robotic arm, we take the inertia parameters of the first two links from the CAD drawings as given base values, which we do not attempt to estimate since the first two segments are not fully actuated. We focus on the remaining two links, after the third joint, to test the outlined strategies. Consequently, there are twenty unknown inertia parameters in the system that we aim to estimate.

As a first step, we formulate the Newton-Euler equations for this system. For the proposed method, it is not necessary to generate physically sensible excitation trajectories since the approach relies on directly measuring all system-relevant information at a given point in time – without the need for numerical differentiation and is currently confined to simulation. While this may lead to suboptimal excitation, it significantly simplifies the data generation process.

The generated data include measurements of angular positions, velocities, and accelerations. Additionally, the generated inertia parameters are checked against boundary constraints to ensure their validity in comparison to real life robots. In particular, we verify that the inertia tensor is positive definite, which is a crucial prerequisite for any real physical inertia tensor.

To generate the data, we first prepared kinematic parameter sets representing physically sensible motions, as well as arbitrary inertia parameter sets for different robotic arm configurations. Our aim was to train the model on varying robot arm configurations to achieve abstraction capabilities and to have multiple datasets for testing the numerical optimization. To this end, we identified boundaries for the kinematic parameters to define the range of values they can take during normal operations of the robotic arm. Furthermore, we used the inertia parameters of the last two links of the arm, collected from the CAD drawings, as base values to create randomly sampled robot configurations.

We decided to generate the data separately and not as part of the training process of the networks – firstly, to save time during training, and secondly, to be able to use the same dataset for the numerical optimization approach. After generating one hundred different inertia parameter sets for each of the two links in question, we applied a full factorial combination of these sets to generate a total of ten thousand unique robot configurations. Each configuration was paired with one hundred randomly generated kinematic parameter sets to create one hundred excitation trajectory frames for each robot configuration. Using the previously derived Newton-Euler equations, we then calculated the instantaneous reaction forces and moments for each of the excitation frames.

### 3.2 Numerical Optimization

For each frame, there are twenty unknown inertia parameters to be estimated but only six equations. To obtain a reasonable estimate for the parameters, we need at least four such observations to have a total of twenty-four equations. For the experiments, we used MATLAB's `lsqnonlin` (least squares nonlinear) solver function to estimate the parameters, treating the first moments as separate constituting parameters – this represents the nonlinear problem. Afterwards, we combined the first moments (the linearized problem) and performed the estimation using the `lsqlin` (least squares linear) solver function – treating the first moments as individual atomic parameters.

### 3.3 Machine learning setup

For the machine learning setup, we used the generated ground-truth data to train different model architectures for this regression problem.

Our work can be divided into multiple iterative steps. Initially, we designed the simplest feed-forward neural network possible for this data. We used only one frame at a time, resulting in twelve measured kinematic parameter values – the angular positions, velocities, and accelerations of all four joints –and six parameter values for the reaction forces and moments, totaling eighteen input parameters $\mathbf{X}$.

Since we aim to estimate the inertia parameters of the last two links of the sequential robotic arm, we try to estimate a combined twenty output values $\mathbf{y}$. In this first step, we aimed to test whether the overall setup is functional and capable of learning the underlying data distribution. To evaluate this, we selected only ten of the generated robot configurations to limit the training time. As shown in Figure 1, the simple neural network structure has a single hidden layer.

After testing this network, we expanded the network architecture by adding additional layers and neurons per layer, as well as exploring different overall shapes of the studied networks. In addition, we tested different activation functions.

To explore the impact of network depth and width on the performance, we evaluated several architectures. We expanded our set of neural networks to include the following configurations:

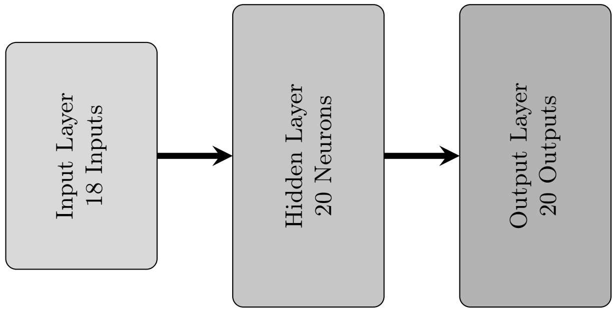


**Fig. 1.** A simple neural network structure with one hidden layer. The diagram shows an input layer with 18 inputs, a hidden layer with 20 neurons, and an output layer with 20 output values.

– **Deeper Networks:** Networks with increased numbers of hidden layers ($L$):
  - $L = 8$
  - $L = 12$
  - $L = 16$
– **Wider Networks:** Networks where each hidden layer has a width ($W$) that is a multiple of the number of desired outputs ($N_{\text{out}}$):
  - $W = k \times N_{\text{out}}$, where $k \in \{1, 2, 4, 8, 16\}$
– **Funnel-Shaped Network:** A network with 6 hidden layers where the number of neurons decreases in each layer:
  - Layer widths: $[64, 32, 16, 8, 4, 2]$

This is summarized in Table 1.

**Table 1.** Extended Neural Network Architectures with Various Depths and Widths

| **Network** | **Description** | **Hidden Layers ($L$)** | **Width Factor ($k$)** | **Layer Width ($W$)** |
|---|---|---|---|---|
| Network 1 | Baseline width | 8 | 1 | $W = N_{\text{out}}$ |
| Network 2 | Double width | 8 | 2 | $W = 2 \times N_{\text{out}}$ |
| Network 3 | Increased depth | 12 | 1 | $W = N_{\text{out}}$ |
| Network 4 | Further increased depth | 16 | 1 | $W = N_{\text{out}}$ |
| Network 5 | Quadruple width | 8 | 4 | $W = 4 \times N_{\text{out}}$ |
| Network 6 | Octuple width | 8 | 8 | $W = 8 \times N_{\text{out}}$ |
| Network 7 | Sixteen times width | 8 | 16 | $W = 16 \times N_{\text{out}}$ |
| Funnel Network | Decreasing layer sizes | 6 | Variable | $W = [64,\ 32,\ 16,\ 8,\ 4,\ 2]$ |

Building upon those experiments, we expanded the network structure to simultaneously receive multiple excitation frames of the same robot configuration as inputs – an approach inspired by the numerical optimization method. As a variation of this technique, we also implemented Siamese Networks. The key characteristic of Siamese Neural Networks is the use of two identical neural networks that share the same architecture and weights [10]. These subnetworks process different inputs in parallel, but because they share the same weights, the overall network is more directly inclined to find outputs that satisfy all given input combinations simultaneously, which is particularly beneficial in our case where the expected output for one robot configuration is an identical set of inertia parameters. Lastly, we attempted to integrate the symbolically formulated Newton-Euler equations into the loss function of the networks, in an effort to create a physics-inspired neural network.

**Custom Loss Function for Physics-Inspired Neural Networks.** To integrate physical consistency into the neural network training, we designed a custom loss function that combines the prediction error of the inertia parameters with the discrepancy in reaction forces and moments computed via the Newton-Euler equations. The loss function is defined as:

$$\mathcal{L}(\hat{\boldsymbol{\theta}}) = \alpha \cdot \frac{1}{n} \sum_{i=1}^{n} \left( \hat{\theta}_i - \theta_i^{\text{true}} \right)^2 + \beta \cdot \frac{1}{N} \sum_{j=1}^{N} \left\| \mathbf{R}_{\text{meas}}^{(j)} - \boldsymbol{\Psi}^{(j)} \hat{\boldsymbol{\theta}} \right\|^2, \tag{5}$$

where:

- $\hat{\boldsymbol{\theta}}$ is the vector of predicted inertia parameters (outputs of the neural network).
- $\theta_i^{\text{true}}$ is the $i$-th ground truth inertia parameter.
- $n$ is the total number of inertia parameters.
- $N$ is the number of excitation frames.
- $\mathbf{R}_{\text{meas}}^{(j)} = \begin{bmatrix} \mathbf{F}_{\text{meas}}^{(j)} \\ \mathbf{M}_{\text{meas}}^{(j)} \end{bmatrix}$ is the vector of measured reaction forces and moments for the $j$-th excitation frame.
- $\boldsymbol{\Psi}^{(j)} = \begin{bmatrix} \boldsymbol{\xi}^{(j)} \\ \boldsymbol{\phi}^{(j)} \end{bmatrix}$ is the regressor matrix derived from the Newton-Euler equations for the $j$-th excitation frame.
- $\alpha$ and $\beta$ are weighting coefficients that balance the importance of the two terms.

In this formulation the first term represents the Mean Squared Error (MSE) between the predicted inertia parameters and the ground truth parameters. The second term, penalizes the discrepancy between the predicted and measured reaction forces and moments, ensuring that the predicted inertia parameters lead to physically consistent behavior according to the Newton-Euler equations.

By minimizing this loss function, the neural network adjusts $\hat{\boldsymbol{\theta}}$ to not only match the ground truth inertia parameters but also to ensure that these parameters result in reaction forces and moments that align with the physical measurements.

The functions $\boldsymbol{\xi}^{(j)}$ and $\boldsymbol{\phi}^{(j)}$ represent the contributions of the kinematic states to the reaction forces and moments, respectively, and are derived from the Newton-Euler equations for the $j$-th excitation frame. This form of the loss function emphasizes the importance of the predicted inertia parameters in reproducing the measured reaction forces and moments according to the Newton-Euler equations, thus ensuring physical plausibility.

For the exploration of the networks we constricted the training on a selection of three-thousand of the ten-thousand generated robot configurations, and trained the models for at least one-hundred epochs each, to limit training time.

## 4 Results

The results of the classical optimization demonstrated that the masses and the center of mass vector values could be correctly identified by this approach for both the linearized and nonlinear problem formulations. The linearized problem could also identify all other parameters correctly, whereas the nonlinear solver produced a wide range of accuracies – between approximately 50% and 95% – for the remaining parameters (inertia entries), depending on the selected robot configuration and especially the selected excitation frames. The iterative exploration of neural network architectures revealed varying degrees of effectiveness among the examined models. A variety of different network architectures were trained using various scaling methods and activation functions, with performance assessed through training loss and prediction accuracy. Under otherwise identical constraints, the *tanh* activation function, the robust scaler, and the SGD optimizer produced the most promising results for the given architectures. Despite extensive testing, the AI-based methods struggled to achieve the precision of classical numerical optimization, given the limited training time and data used to explore the presented network architectures. Although the approach of a physics-inspired neural network worked in principle, it led to unmanageable and impractical training times. This was because TensorFlow could not sufficiently optimize the backpropagation step for the custom loss function, which prevented us from performing this attempt with the limited processing power available at the time. In this preliminary exploration, the most promising result was produced by the Siamese network with eight hidden layers and a width factor of eight. However, it still could not approach the performance of the numerical optimization approach.

## 5 Conclusions

Our findings underscore the potential of combining direct dynamic measurements with numerical optimization and machine learning methods for inertia parameter identification. The performance of the studied neural networks, compared to numerical solvers, reveals both their limitations and their potential when training time is extended and the dataset is expanded. Since the influence of noisy sensor data was not studied in this work, this remains an important area for future research. Incorporating noisy data could provide a more relevant and realistic use case, where a well-designed model can demonstrate its generalization and robustness capabilities.

## References


1. Hollerbach, J., Khalil, W., Gautier, M.: Model Identification. Springer Handbooks. Springer International Publishing (2016)
2. Lages, W.F.: Parametric identification of the dynamics of mobile robots and its application to the tuning of controllers in ros. In Koubaa, A., ed.: Robot Operating System (ROS). Volume 707 of Studies in Computational Intelligence. Springer International Publishing (2017) 191–229
3. Leboutet, Q., Roux, J., Janot, A., Guadarrama-Olvera, J.R., Cheng, G.: Inertial parameter identification in robotics: A survey. Appl. Sci. **11**(9) (May 2021) 4303
4. Lee, T., Lee, B.D., Park, F.C.: Optimal excitation trajectories for mechanical systems identification. Automatica **131** (2021) 109773
5. Gayral, T., Daney, D.: A sufficient condition for parameter identifiability in robotic calibration. In Thomas, F., Perez Gracia, A., eds.: Computational Kinematics, Dordrecht, Springer Netherlands (2014) 131–138

6. Gießler, M., Werth, J., Waltersberger, B., Karamanidis, K.: A wearable sensor and framework for accurate remote monitoring of human motion. Commun. Eng. **3**(1) (Jan. 2024) 20
7. Giessler, M., Waltersberger, B.: Computational advantages in robotics by evaluating newton-euler equations with respect to a moving reference point in a non-inertial frame. In Review (Jun 2022) preprint.
8. Featherstone, R. In: The Recursive Newton-Euler Algorithm. Springer Berlin Heidelberg, Berlin, Heidelberg (2020) 1–5
9. Atkeson, C.G., An, C.H., Hollerbach, J.M.: Estimation of inertial parameters of manipulator loads and links. The International Journal of Robotics Research **5**(3) (1986) 101–119
10. de Rosa, G.H., Papa, J.P.: Chapter 7 - learning to weight similarity measures with siamese networks: a case study on optimum-path forest. In Falcao, A.X., Papa, J.P., eds.: Optimum-Path Forest. Academic Press (2022) 155–173