Inertia Parameter Identification with machine learning

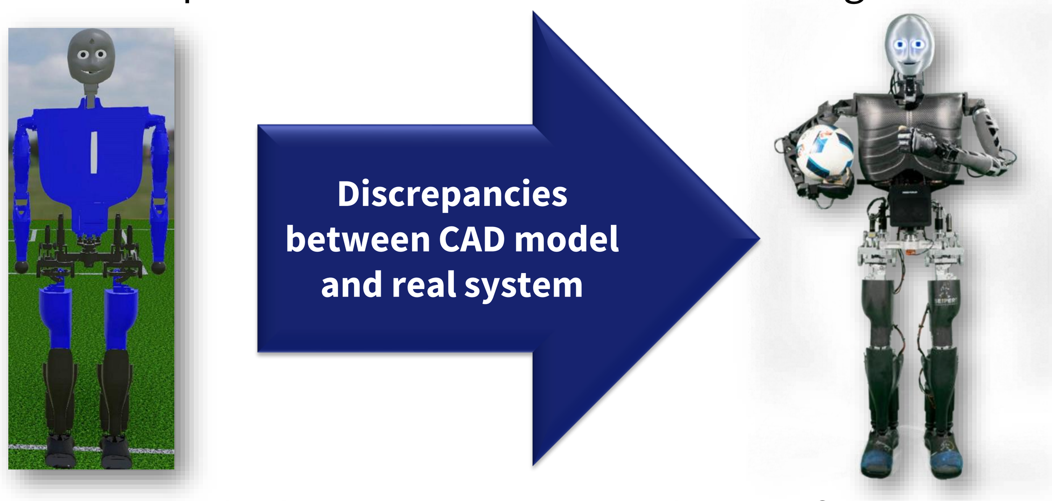
HOCHSCHULE OFFENBURG

Exploration of Neural Network Architectures for inertia parameter identification of a robotic arm

Background & Problem Description

Background

- Inertia parameters are crucial for control algorithms and precise calculations



Inertia parameters per body-segment:
- **mass (m_i)**
- **center of mass vector ($\vec{r_i}$)**
- **inertia-tensor-entries ($I_{xx}, I_{xy}, \dots$)**

➢ **10 – parameters per segment!**

- Commonly the Lagrangian Formulation of the equations of motion are used
  ➢ n-Equations on joint level (one equation per joint)
  represented by a set of n equations for the generalized forces — specifically the joint torques(τ_i , i = 0..n) of the joints connecting the links

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

  ➢ 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 θ

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

Problem Description

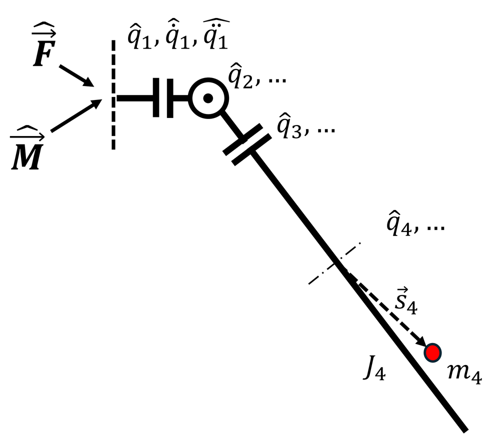
- Joint motor encoder signals are used to calculate the joint-angular velocities and accelerations – using numerical differentiation – which is prone to errors because of noisy sensor data; this is aggravated due to the numerical differentials
- The formulation of the equations of motions on joint level has problems with friction terms and other potential non-linearities

Overall Approach

We use a four-degrees of freedom – robotic arm from the humanoid robot "Sweaty" from Offenburg University as Example Problem

Newton-Euler equations with direct measurements

- We use a newly designed sensor from Offenburg University to **directly measure** the kinematic joint parameters: $\hat{q}_i, \hat{\dot{q}}_i, \hat{\ddot{q}}_i$
- Additionally, we use a force-torque sensor at the base of the robotic arm, to measure the reaction forces and moments resulting by the overall movement of the arm: $\hat{F}, \hat{M}$



Steps

1. Formulating the Newton-Euler equations (6-equations) at the base of the robot
2. Generating random robot configurations (Inertial-parameter sets), under physical constraints
3. Generating excitation frames ("snapshot" of physical sensible kinematic parameter values) ⇒ circumventing the optimal excitation trajectory optimization
4. Use robot configurations and excitation frames to calculate resulting ground truth reaction forces and moments
5. Perform Numerical optimization and NN-Training on generated data sets.

Numerical Optimization and NN-Architecture Exploration

- Numerical Optimization for both the nonlinear and linearized problem
- Exploration of different MLP architectures with growing complexity as well as Siamese Network architectures
- Implementation of a custom physics inspired loss function

Neural Network Architectures

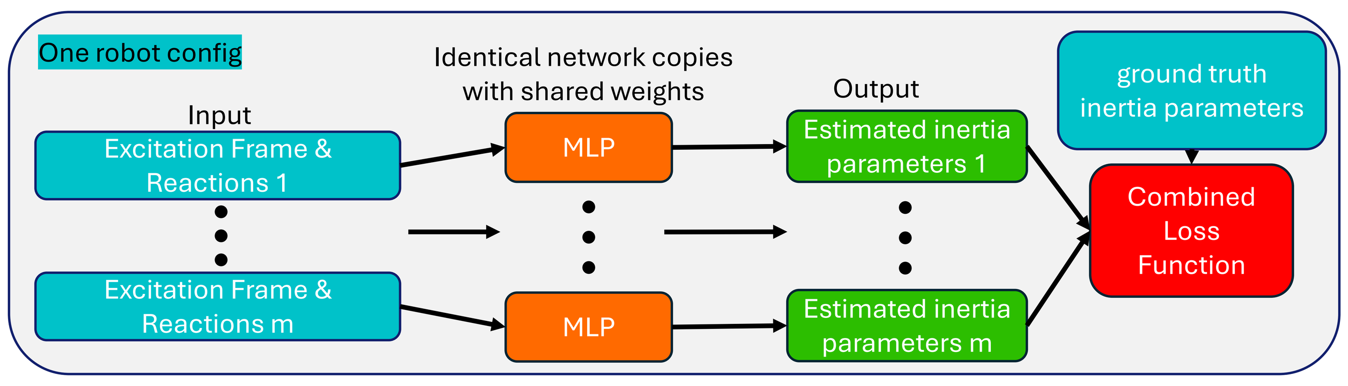
Exploration of Fully-Connected Networks

After testing the general data generation pipeline with a minimal possible MLP, we explored in multiple iteration steps different architectures
Furthermore, we expanded the network dimensions, to use multiple excitation frames per inference – mimicking the numerical approach using least square estimators.

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	$W = [64, 32, 16, 8, 4, 2]$

Siamese-Network Architectures

Identical MLPs with different excitations and reactions, but with combined loss-function.
➢ Minimize sum of squared errors of all estimates



Physics inspired loss function

Use of the equations to enforce physically consistent outputs by adding an additional term to the loss function. Using the current **excitation frame** and **estimated inertia parameters** to calculate the **estimated reaction forces and moments** – minimizing the difference between estimated reactions and ground truth reactions – in addition to minimizing the Mean Squared Errors of the single predictions compared to the ground truth.

Results

Results

- Numerical Optimization
  ➢ Accurately identified **mass** and **center of mass** for linear and nonlinear cases.
  ➢ Linear approach accurately estimated additional Inertia-Tensor parameters; nonlinear solver showed variable accuracy (50 - 95%).
- Neural Network Exploration
  ➢ Limitations of AI Methods Precision limited by data, training time, and TensorFlow's handling of custom loss functions.
  ➢ Best AI Model:
  Siamese network with eight hidden layers performed best
  but did not match classical optimization.

Conclusion and Future Work

- Directly measured kinematic parameter values are promising and should be compared to a traditional approach using Lagrangian formulation with numerical differentiation of motor encoder values
- With real world noisy sensor data, the numerical approach should be compared with the best AI approach
- The incorporation of the equation in the loss function are promising but need further optimization to be viable

Prof. Dr. Stefan Hensel | Prof. Dr. Bernd Waltersberger | Stefan Glaser | Maximilian Giessler | Thomas Granser

stefan.hensel | bernd.waltersberger | stefan.glaser | maximilian.giessler | thomas.granser

@hs-offenburg.de