Experimental Robot Inverse Dynamics Identification Using Classical and Machine Learning Techniques

Vinzenz Bargsten, Robotics Research Group, University of Bremen, bargsten@uni-bremen.de, Germany
José de Gea Fernández, Robotics Innovation Center, DFKI, jose.de_gea_fernandez@dfki.de, Germany
Yohannes Kassahun, Robotics Innovation Center, DFKI, yohannes.kassahun@dfki.de, Germany

Abstract

This paper shows the experimental identification of the inverse dynamics model of a KUKA iiwa lightweight robot. We use experimental data from optimal identification experiments to evaluate and compare two different identification approaches: a classical method using a parametrized robot dynamical model and a machine learning method. Both methods accurately estimate the dynamics model and this paper will discuss the pros and cons of each method.

1 Motivation

The main focus of this paper is on comparing two different estimation techniques for the motion dynamics of a robotic arm using the same experimental data. A 'classical' technique that uses a parametrized robot dynamic model based on rigid body physics, and a machine learning technique which does not require the prior knowledge and thus has the potential to further automate the identification process and to capture otherwise unmodeled dynamics.

2 Experimental Setup

The 7-joint robotic arm shown in Figure 1 is connected to a controller by the manufacturer. It executes a small relay program to receive commands to be executed in the manufacturer-specific domain and to transmit the status data of the arm through a network connection. An external control system generates the joint commands and handles the status data within a component-based software framework ROCK [1].

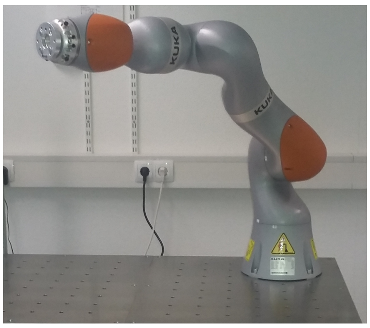

Figure 1: KUKA iiwa R820 lightweight robotic arm used for the experiments

[1] http://www.rock-robotics.org

3 Dynamics Identification

An inverse model of the robot motion dynamics is the mapping from the motion of the robot, given by the joint positions $q(t) \in \mathbb{R}^n$, joint velocities $\dot{q}(t)$ and joint accelerations $\ddot{q}(t)$ to the actuation torques $\tau(t) \in \mathbb{R}^n$ dependent on time t:

$$\tau(t) = f(q(t), \dot{q}(t), \ddot{q}(t)). \tag{1}$$

Knowledge of this model is a basis for dynamic control schemes and estimation of contact forces from joint torques.

3.1 Classical Method

Modeling Assuming the robot consists of n rigid bodies connected by the actuated joints, we can apply an algorithm such as the Recursive-Newton-Euler algorithm or the Lagrange-Formalism to derive (1) from known physical laws for the a-priori known geometry of the robot. This yields a theoretical model, still including unknown dynamic parameters such as the mass m_i, the three first moments of inertia $m_i c_{[x|y|z],i}$, or the six second moments of inertia (inertia tensor) $I_{[xx|xy|xz|yy|yz|zz],i}$, for each body i of the robot. Since joint friction usually cannot be neglected, we add two additional parameters, $F_{c,i}$ and $F_{v,i}$, as coefficients of a Coulomb and viscous friction model. Expressing the set of parameters for each link in a coordinate system fixed to the link, results in a set of constant inertial parameters. Moreover, we can obtain a linear relationship, advantageous for the further processing, between these parameters and the rest of the theoretical model. The resulting rigid-body model thus has the form

$$\tau(t) = \mathbf{Y}\left(q(t), \dot{q}(t), \ddot{q}(t)\right)\, \theta, \tag{2}$$

where $\theta \in \mathbb{R}^{12n}$ denotes the parameter vector with the n sets of parameters θ_i,

$$\theta_i = \left(m_i\; m_i c_{x,i}\; m_i c_{y,i}\; m_i c_{z,i}\; I_{xx,i}\; I_{xy,i}\; I_{xz,i}\; I_{yy,i}\; I_{yz,i}\; I_{zz,i}\; F_{c,i}\; F_{v,i}\right)^T.$$

Identification For a reference trajectory sampled at $t = kT_s, k \in 1...K$ with sampling time T_s, an *identification matrix* $\varPhi$,

$$\varPhi = \begin{pmatrix} \mathbf{Y}\left(q(T_s), \dot{q}(T_s), \ddot{q}(T_s)\right) \\ \dots \\ \mathbf{Y}\left(q(kT_s), \dot{q}(kT_s), \ddot{q}(kT_s)\right) \\ \dots \\ \mathbf{Y}\left(q(KT_s), \dot{q}(KT_s), \ddot{q}(KT_s)\right) \end{pmatrix}, \tag{3}$$

can be created. Using stiff position controllers, close tracking of the reference trajectory by the robot yields the required torques, $\tau_m(kT_s)$. A similar procedure as in [4] has been applied, to obtain a sufficiently *rich*, periodic, band-limited excitation trajectory through optimization of the parameters of a Fourier-Series. The dynamic parameters $\hat{\theta}$ are estimated by minimization of the squared error between measured τ_m and computed torque $\varPhi\hat{\theta}$, while physical consistency is handled by constraints such as positive masses, positive definite inertia tensor and positive friction coefficients.

3.2 Machine Learning Method

The machine learning method uses a committee of neural networks for learning the inverse model of the robot. The neural networks belong to the class of networks where the first layer is generated randomly and kept constant, and only the output layer is optimized during learning. Examples of such networks are Echo State Networks (ESNs) [2] and Extreme Learning Machines (ELMs) [1]. These types of networks can easily be used for either off-line or on-line learning and a number of optimization techniques can be applied for optimizing the output weights. In our particular implementation, we used a committee of ten ELMs. Each ELM has 500 hidden neurons and a rectifier activation function $f(a) = \max(0, a)$, where a is the activation of a hidden unit node. Each ELM learns the function given by Equation (1) directly from the training data with an input vector $\mathbf{u}(t) = [q(t), \dot{q}(t), \ddot{q}(t), 1]^T \in \mathbb{R}^{3n+1}$ and an output vector $\hat{\boldsymbol{\tau}}(t)$. Assume that $T_h = \{\mathbf{h}(t_1), \dots, \mathbf{h}(t_K)\}$ is a set of vectors of hidden unit activations calculated from $\mathbf{u}(t)$ and $\{\boldsymbol{\tau}(t_1), \dots, \boldsymbol{\tau}(t_K)\}$ is the corresponding set of vectors of the measured torque values used for training. The solution to the output matrix $\mathbf{W}_o$ is given by

$$\mathbf{W}_o = (\mathbf{H} + \beta\mathbf{I})^{-1}\mathbf{T}, \tag{4}$$

where $\beta = \frac{1-p}{p}, p \in (0, 1]$, $\mathbf{H} = \mathbb{E}[\mathbf{h}\mathbf{h}^T]$ and $\mathbf{T} = \mathbb{E}[\boldsymbol{\tau}\boldsymbol{\tau}^T]$. In our experiment, $p = 0.5$. The time complexity for training is comparable to the classical method.

4 Discussion

Preliminary results are shown in Figure 2. Both methods are able to fit the measured (training) data closely. Since the measured data is the same as the training data here, a validation experiment is still to be carried out, where e.g. random point-to-point motions of the robotic arm is used to generate independent measurement data for validation. It is clear from the paper that the classical method requires a great deal of knowledge to apply it for system identification. On the other hand the machine learning method requires a sufficiently large amount of data that has all the necessary information needed for modeling. An estimate of the amount and quality of data required, e.g. the performance with less optimized identifiaction experiments, for both methods will be discussed. We plan to include results on on-line-learning of an inverse model for the machine learning method [3] in the full paper.

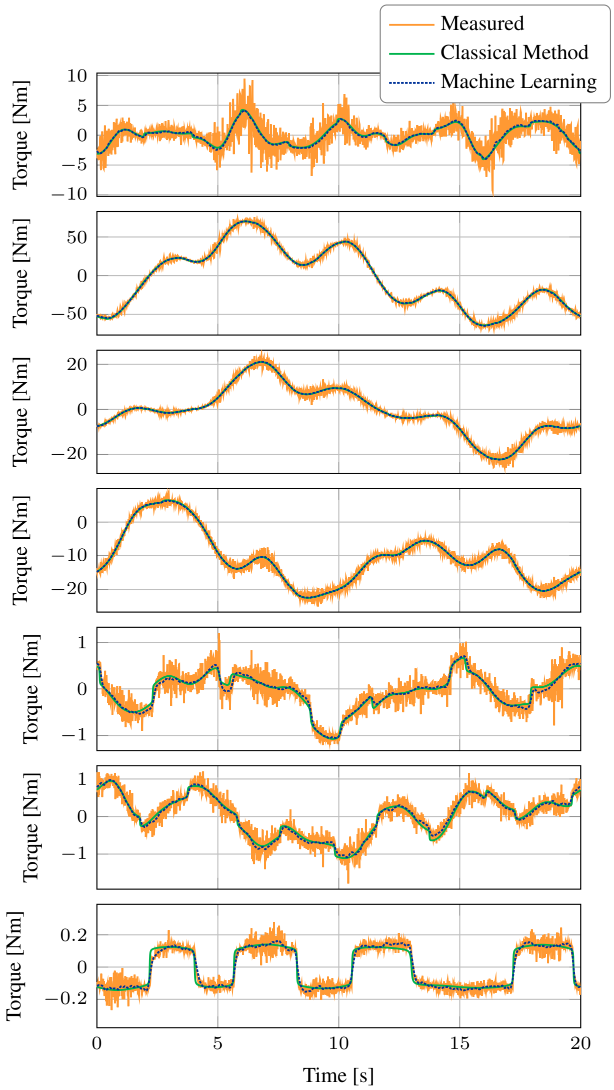

Figure 2: Comparison of measured and model-fitted torques. From top to bottom: joints 1 to 7.

References

[1] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006.

[2] Herbert Jaeger and Harald Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, pages 78–80, 2004.

[3] Franziska Meier and Stefan Schaal. Drifting gaussian process regression for inverse dynamics learning. In IROS 2015 Workshop on Machine Learning in Planning and Control of Robot Motion, 2015.

[4] J. Swevers, W. Verdonck, and J. De Schutter, "Dynamic Model Identification for Industrial Robots," *IEEE Control Syst. Mag.*, vol. 27, no. 5, pp. 58–71, 2007.