2013 IEEE International Conference on Control Applications (CCA)
Part of 2013 IEEE Multi-Conference on Systems and Control
Hyderabad, India, August 28-30, 2013

Modeling, Parameter Identification and Model-Based Control of a Lightweight Robotic Manipulator

Vinzenz Bargsten, Pablo Zometa, and Rolf Findeisen

***Abstract*— Nowadays many robotic tasks require close and compliant tracking of desired positions, paths, or forces. To achieve these goals, model-based control schemes provide a possible solution, allowing to directly consider the nonlinear dynamics. One of the key challenges, however, is the derivation of suitable models, which allow sufficiently fast evaluation, as well as the parameterization of these models based on measurements. In this work we outline and review a structured approach for model-based controller design for robots. In a first step we derive suitable models for multi-link robots. In a second step we review how such models can be parameterized and how optimal identification experiments can be designed. Based on the model we then derive a simple model based controller to experimentally validate the results considering a lightweight robot. The single steps of the derivation and controller design are supported by a newly developed freely available model toolbox for the considered lightweight robot.**

I. Introduction

Modern industrial manipulators rely on controllers to execute complex tasks [1], [2], [3]. We outline a structured approach for modeling, parameter identification and model-based control which is, in the case of the considered lightweight robot, supported by a suitable freely available toolbox. The focus is on robotic manipulators with revolute joints actuated by electric motors.

We do not attempt to provide a comprehensive review of parameter estimation or model-based control of robotic manipulators. Instead, we refer the reader to [1] and references therein. However, there are only limited works covering the complete cycle from modeling, estimation of parameters, and controller design. For example, in [4] motion control of a Mitsubishi PA-10 robotic arm is considered. A dynamic model is derived that includes the drive transmission, the effects of gravity, but no inertial effects. Based on the model, a model-based controller is implemented and experimentally verified.

The main contribution of this work is to demonstrate a structured design approach from modeling, parameter estimation, controller design up to the implementation of a model-based controller on a commercially available robotic arm, highlighting the challenges encountered along the way. Additionally, we present a freely available software library useful for simulation and model-based control for the considered *KUKA lightweight robot IV* (*LWR*). The outlined structured approach consists of a series of consecutive steps. In the first step we derive a model which is well suited for parameter identification. Based on this model, we then proceed with the estimation and validation of unknown or uncertain parameters. In the third step we transform the identified model into two forms, one for simulation and the other for model-based controller design. Finally, based on the identified model in these two forms, we can design and implement motion controllers using, for example, feedback linearization methods.

This article is structured as follows: Section II describes a method for modeling of robots which is well suited for parameter identification. We furthermore outline how one can optimally identify the parameters. Section III discusses the design of model-based controllers. Section IV presents the application of the discussed methods to the *LWR*, before we conclude in Section V.

VB is with Robotics Research Group, University of Bremen, Germany. E-mail: bargsten@uni-bremen.de. PZ and RF are with the Institute for Automation Engineering, Otto-von-Guericke-Universität Magdeburg, Germany. E-mail: {pablo.zometa, rolf.findeisen}@ovgu.de.

II. Modeling and Parameter Identification

We consider the robotic arm shown in Figure 1. The arm consists of 7 revolute joints. Each of the joints is actuated by an electric motor. Moreover, each joint is equipped with a position sensor and a link-side torque sensor. We assume the robot links to be rigid bodies and the joints to be rigid transmissions. For a detailed description of the robot we refer to [5].

In a first step we consider a dynamical model that relates the robot motion in terms of the joint positions $q(t) \in \mathbb{R}^n$, joint velocities $\dot{q}(t)$ and joint accelerations $\ddot{q}(t)$ with the respective actuation torques $\tau(t) \in \mathbb{R}^n$. The overall dynamics is given by (dropping for simplicity in the following the time dependency)

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

The dynamics is derived using the Lagrangian formalism [1], using coordinate systems attached to the respective links. This leads to an overall model, which is linear in the yet unknown system parameters. The linearity will be exploited for the parameter identification based on experiments.

We obtain for the equations of motion

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

where π is the vector of unknown parameters consisting of the stapled parameters for each single link i given by

$$\pi_i = \left(m_i \; m_i l_{Cx,i} \; m_i l_{Cy,i} \; m_i l_{Cz,i} \; \hat{I}^i_{i,xx} \; \hat{I}^i_{i,xy} \; \hat{I}^i_{i,xz} \; \hat{I}^i_{i,yy} \; \hat{I}^i_{i,yz} \; \hat{I}^i_{i,zz} \; F_c \; F_v\right)^T.$$

Here m_i denotes the mass of the link. Furthermore π_i contains the three first moments of inertia $m_i l_{Cx,i}, m_i l_{Cy,i}, m_i l_{Cz,i}$, the six components of the

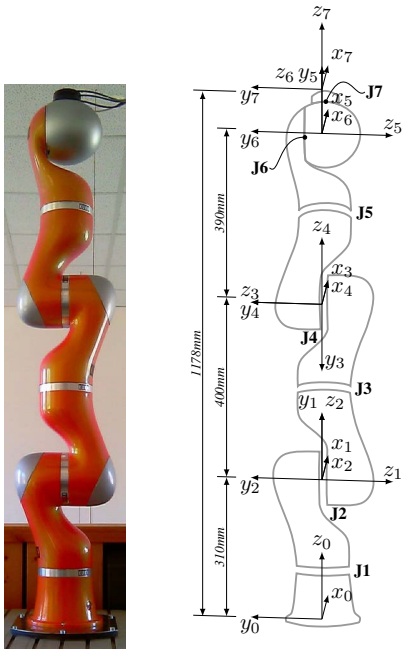

Fig. 1. *KUKA lightweight robot IV*.

inertia tensor (the terms $\hat{I}^i_{i,..}$) as well as Coulomb and viscous friction parameters F_c and F_v, respectively.

A detailed description of the appearing terms and formulations can be found in [6].

A. Optimal Experiment Design

While geometric parameters are measured/estimated directly from physical dimensions, there are many approaches to estimate the dynamic parameters. We focus here on estimation based on experiments. In particular, we use a least squares parameter estimation scheme. To sufficiently excite the robot we use the experimental design approach described in [7] to obtain optimal identification experiments for good estimation results.

For the parameter estimation, the joint torques as well as the joint positions are assumed to be measured, while the joints track specified *excitation trajectories*. To obtain a good estimation of the parameters, the robot motion should sufficiently excite the effects dependent on the parameters to be identified. The excitation trajectories are chosen such that the information content about the unknown parameters in the measurement data is maximized while satisfying constraints of the robot (e.g. maximum velocity, geometric constraints). The inputs are parameterized in terms of Fourier series functions

$$q_i(t) = q_{i,0} + \sum_{k=1}^{n_h} \left(a_{i,k} \sin(k\,\omega_f\,t) + b_{i,k} \cos(k\,\omega_f\,t) \right). \tag{3}$$

Here $q_i(t)$ denotes the joint angles for each joint i as function of the Fourier coefficients $q_{i,0}$, $a_{i,k}$ and $b_{i,k}$. Joint velocities $\dot{q}_i(t)$ and accelerations $\ddot{q}_i(t)$ are respectively obtained by analytic differentiation of (3) with respect to time. Choosing a Fourier basis allows for a band-limited excitation frequency, determined by the fundamental frequency ω_f and the number of harmonics n_h, which is a well suited approach for mechanical systems. This facilitates a trade-off between adequate excitation with respect to the parameters to be estimated and excitation of unmodeled dynamics. Thanks to the periodicity of (3), multiple periods can be measured consecutively. Time domain data averaging of the measured periods of data reduces the effect of random errors.

For the experiment optimization and the parameter estimation process we consider sampled trajectories with K samples and sampling period T_s. The measurements are then condensed in the *identification matrix* Φ, which is given by

$$\Phi = \begin{pmatrix} \mathbf{Y}\left(q(T_s), \dot{q}(T_s), \ddot{q}(T_s)\right) \\ \dots \\ \mathbf{Y}\left(q(kT_s), \dot{q}(kT_s), \ddot{q}(kT_s)\right) \\ \dots \\ \mathbf{Y}\left(q(KT_s), \dot{q}(KT_s), \ddot{q}(KT_s)\right) \end{pmatrix}. \tag{4}$$

It is important to ensure identifiability of the parameters to be estimated. For this purpose, a numerical reduction procedure as presented in [8] is used. It allows the determination of a minimal set of parameters in an automated way by regarding the matrix rank of the identification matrix Φ evaluated for random values. Its columns are examined and grouped depending on the change of the matrix rank. We further refer to $\bar{\mathbf{Y}}$ as the reduced form of the matrix $\mathbf{Y}$. Accordingly, $\hat{\pi}$ refers to the reduced set of parameters and we determine a *reduced* identification matrix $\bar{\Phi}$ by using $\bar{\mathbf{Y}}$ instead of $\mathbf{Y}$ in (4). The reduced form of (2) is given by

$$\tau = \bar{\mathbf{Y}}(q, \dot{q}, \ddot{q})\,\hat{\pi}. \tag{5}$$

In our case the reduction procedure reduced the number of parameters from 84 to 57.

The Fourier coefficients $q_{i,0}$, $a_{i,k}$ and $b_{i,k}$, which determine the sampled trajectory, are optimized with respect to the information content in the measurement data. Specifically, a *d-optimality* criterion (refer to [9], [7], [10, p. 359, 464ff]) is chosen, since it allows to minimize uncertainty on the parameter estimates. The noise of the actuator torque measurements is incorporated using a weighting matrix which is motivated by the covariance matrix Σ of the torque measurements.

The resulting optimization problem for the design of optimal trajectories in terms of the Fourier series coefficients is given by

$$\underset{q_{i,0}, a_{i,k}, b_{i,k}}{\text{minimize}} \; -\log\left(\det\left(\bar{\Phi}^T\,\Sigma^{-1}\,\bar{\Phi}\right)\right), \tag{6}$$

which is subject to the joint or Cartesian position limits, velocity limits, acceleration limits or limits of actuation torques. The latter requires an estimate of the parameters from e.g. previous, slower experiments. The resulting optimal experiments are discussed in Section IV.

B. Parameter Identification

The parameter identification is based on real experiments measuring the tracked trajectories. Based on the measured joint positions, the Fourier coefficients $q_{i,0}$, $a_{i,k}$ and $b_{i,k}$ are

readjusted to account for tracking errors, so that the analytic function reflects the actually measured positions accurately.

To reduce the influence of the noisy torque measurements τ_{msr}, for each experiment multiple repetitions of the motions are performed and the results are averaged. The parameters are then estimated using the averaged measurements by least squares estimation

$$\hat{\pi}_{\text{LS}} = \left(\bar{\Phi}^T \bar{\Phi}\right)^{-1} \bar{\Phi}^T \tau_{\text{msr}}. \tag{7}$$

Here $\hat{\pi}_{\text{LS}}$ denoted the vector of estimated parameters. Alternatively weighted least squares techniques can be used to account for different levels of variance in the torque measurements.

The precision of the resulting model and the parameters are discussed in Section IV-B.

III. Model-Based Controller Design

Tracking of trajectories in operational space is an essential task to be performed. We consider the design of a model-based controller that closely tracks desired trajectories in joint space. For an in-depth description of this topic see [1], [11], [2].

In a first step we derive, based on the identified model, a suitable model in joint space. In a second step we outline a controller based on inverse dynamics. Figure 2 shows the structure of the overall resulting model-based controller.

A. Model Transformation

The model in the form (5) is especially useful for the design of optimal experiments and parameter identification. For the purpose of control, however, it is of advantage to transform (5) into the following equivalent form:

$$\tau = \mathbf{B}(q)\ddot{q} + \mathbf{C}(q,\dot{q})\dot{q} + g(q) + \tau_{\text{F}}(\dot{q}), \tag{8}$$

where $\mathbf{B}(q)$ describes inertial effects, $\mathbf{C}(q,\dot{q})$ describes Coriolis and centrifugal effects, $g(q)$ and $\tau_{\text{F}}(\dot{q})$ describe the influence of gravity and friction respectively, [3], [1].

For implementation and simulation we furthermore consider the following implicit form, which can be efficiently solved by suitable implicit solvers, such as [12]

$$h(t,s,\dot{s}) = 0,\ \ s(t_0) = s_0,\ \ \dot{s}(t_0) = \dot{s}_0.$$

Here s and h are the transformed state vector and dynamic system, respectively. The initial values are given by s_0, $\dot{s}_0$. Accordingly, we can define the state vector $s = (q^T, \dot{q}^T)^T$ and its time derivative $\dot{s} = (\dot{q}_p^T, \ddot{q}_p^T)^T$. Note, that we distinguish the joint velocities of the state vector and its time derivative by an additional index, i.e. $\dot{q}$ and $\dot{q}_p$. Subsequently, we obtain the following model

$$h(t,s,\dot{s}) = \begin{pmatrix} \dot{q} - \dot{q}_p \\ \bar{\mathbf{Y}}(\ddot{q}_p, \dot{q}, q)\hat{\pi} - \tau \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{9}$$

Note that this formulation allows to determine analytically a Jacobian matrix $\mathbf{J}$ without inversion of the inertia matrix $\mathbf{B}(q)$. For this problem it is given by

$$\mathbf{J} = \frac{\partial h}{\partial s} + \alpha \frac{\partial h}{\partial \dot{s}}.$$

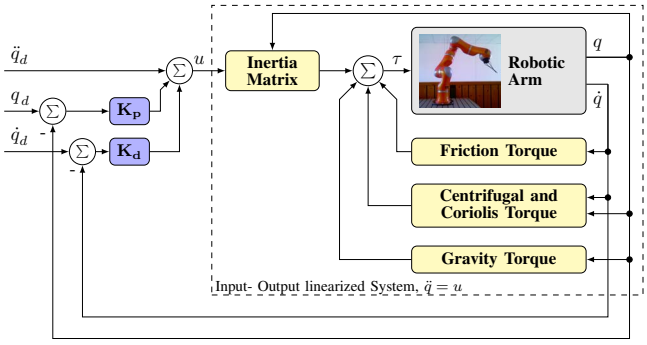

Fig. 2. Structure of the model-based controller consisting of feedback linearization and a linear feedback controller. Note that measurement of $\dot{q}$ is not available and has to be estimated.

B. Inverse Dynamics Control

To solve the trajectory tracking problem, we use a *computed torque controller*, also known as *inverse dynamics controller*. In this controller design a feedback is chosen such that the controlled dynamics are linear, a concept generally known as *exact* or *feedback linearization* [13]. Its main advantage is that we can design linear controllers that stabilize the system. The downside is, however, the requirement for an accurate model of the robot dynamics. Furthermore, the model equations have to be computed in real time and knowledge of the state, i.e. the joint positions and joint velocities, is required.

For the given setup, the feedback linearizing control law is given by

$$\tau = \hat{\mathbf{B}}(q)u + \hat{\mathbf{C}}(q,\dot{q})\dot{q} + \hat{g}(q) + \hat{\tau}_{\text{F}}(\dot{q}).$$

The new input vector is u. $\hat{\mathbf{B}}(q)$, $\hat{\mathbf{C}}(q,\dot{q})$, $\hat{g}(q)$ and $\hat{\tau}_{\text{F}}(\dot{q})$ respectively denote the unknown equivalents of the robot motion dynamics (8). To improve the readability, we define

$$\begin{aligned}\Delta n(q,\dot{q}) := &\left(\mathbf{C}(q,\dot{q}) - \hat{\mathbf{C}}(q,\dot{q})\right)\dot{q} + \left(g(q) - \hat{g}(q)\right) \\ &+ \left(\tau_{\text{F}}(\dot{q}) - \hat{\tau}_{\text{F}}(\dot{q})\right).\end{aligned}$$

Applying the above control law we obtain

$$\hat{\mathbf{B}}(q)u = \mathbf{B}(q)\ddot{q} + \Delta n(q,\dot{q}).$$

We solve for $u - \ddot{q}$, i.e.

$$u - \ddot{q} = \left(\mathbf{I} - \mathbf{B}(q)^{-1}\hat{\mathbf{B}}(q)\right)u + \mathbf{B}(q)^{-1}\,\Delta n(q,\dot{q}).$$

In addition, we use a PD feedback controller to compensate for errors. The input u is then determined by

$$u = \ddot{q}_d + \mathbf{K_p}\tilde{q} + \mathbf{K_d}\dot{\tilde{q}}.$$

This results in the following error dynamics:

$$\begin{aligned}\ddot{\tilde{q}} + \mathbf{K_p}\tilde{q} + \mathbf{K_d}\dot{\tilde{q}} = &\left(\mathbf{I} - \mathbf{B}(q)^{-1}\hat{\mathbf{B}}(q)\right)u + \\ &\mathbf{B}^{-1}(q)\,\Delta n(q,\dot{q}).\end{aligned}$$

For a perfect model, the error system is given by:

$$\ddot{\tilde{q}} + \mathbf{K_d}\dot{\tilde{q}} + \mathbf{K_p}\tilde{q} = 0.$$

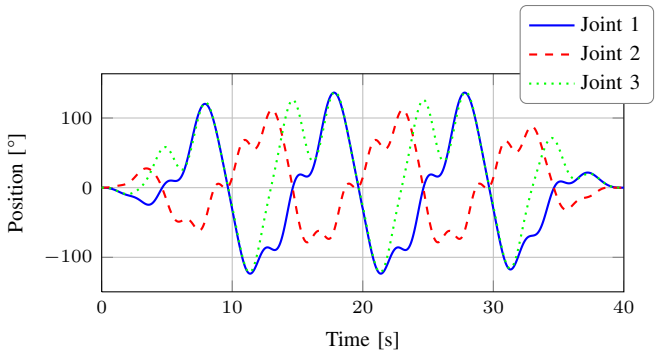

Fig. 3. Four periods of the optimized trajectory for the first 3 joints with smooth fade-in and fade-out of the motion.

An overview of the resulting overall structure of inverse dynamics control is shown in Figure 2.

The matrices $\mathbf{K_p}$ and $\mathbf{K_d}$ are positive definite design matrices. Choosing them diagonal, i.e. $\mathbf{K_p} = \mathrm{diag}\{\omega_1^2, \ldots, \omega_n^2\}$ and $\mathbf{K_d} = \mathrm{diag}\{2\delta_1\omega_1, \ldots, 2\delta_n\omega_n\}$, we obtain decoupled error dynamics for each joint. These matrices determine the stiffness and damping coefficients. The frequency ω_i determines the oscillation frequency of the undamped system i, whereas the corresponding damping coefficient is determined by δ_i. For a single joint the error dynamics are determined by

$$\ddot{\tilde{q}}_i + 2\omega_i\delta_i\dot{\tilde{q}}_i + \omega_i^2\tilde{q}_i = 0. \tag{10}$$

Since the matrices $\mathbf{K_p}$ and $\mathbf{K_d}$ are positive definite, the error converges asymptotically to zero. For a good error compensation without overshooting we chose critical damping, $\delta_i = 1, \ \forall\, i = 1 \ldots n$. The frequency ω_i allows us to choose the stiffness of the system or how fast the error converges.

IV. Experimental Validation

In the following we outline the experimental results achieved employing the design procedure, consisting of modeling, optimal experimental design, parameter estimation, and feedback controller design.

A. Optimal Experiment Design

For the design of experiments (6) we incorporate constraints of joint position q, height of the robot end effector z_e, joint velocity $\dot{q}$ and joint acceleration $\ddot{q}$. Self-collision is prevented by the joint position constraints. Since no payload is attached to the robot for the experiments, the actuation torques are not near their actual limits. Torque constraints are therefore determined only by the acceleration and velocity limits used in the experiments. Values for position and velocity constraints are taken from the robot specifications. The height limits can be determined by measurement of the robot environment. Such environmental constraints in general prevent the robot from colliding with the floor, walls, etc. Acceleration limits have been iteratively determined. Using the existing position control mode of this robot, we execute the experiments. An example motion for an identification experiment is shown in Figure 3. Parameter estimates with a high variance over different experiments were iteratively set to zero, so that we obtain 25 remaining non-zero parameters.

B. Model Validation

Based on the measurement data of about 30 experiments, we obtain an estimate of the parameters according to (7). We did not use the weighted least squares technique to avoid introducing an additional bias on the estimates due to unmodeled effects (such as position dependent friction).

The model quality is estimated computing the joint torques for a given motion, which is then experimentally validated. The quality of the inverse dynamics is especially important for the model-based controller. Specifically we validate the *inverse dynamic* form of the model (8) for a random motion of the robot. For this motion, the joint torques are computed according to (5) using the estimated parameters $\hat{\pi}_{\mathrm{LS}}$ as parameter vector $\hat{\pi}$. The computed torques are then compared to the measured torques for this motion.

Figure 4 compares the model generated and measured torques of each joint. It shows that the model torques are in good agreement with the measured torques, except for joint 7. Note that the range of the applied torques are quite different for each individual joint. While large torques are applied to joints 2, 3 and 4 to withstand gravity, the torques applied to joints 5, 6 and 7 are almost two orders of magnitude smaller, since these joints are at the end of the manipulator and since no payload is used. This also explains the limited quality, since for smaller torques, unmodeled effects and measurement noise have a larger influence.

Additionally, measured torques often have measurement offsets. The torque measurement can be calibrated at a joint position where no torques due to gravity are active. However, effects such as stiction make a calibration difficult and small offsets, e.g. in the range of $\pm 0.1\,\mathrm{Nm}$ to $\pm 0.3\,\mathrm{Nm}$ can remain. In view of these facts, the model accuracy for the joints 5 and 6 is good, while for joint 7 unmodeled friction effects dominate.

Overall the model is in sufficiently good agreement for control.

C. Simulation and Modeling Toolbox

To simplify the usage of the derived model, we developed a software library that provides the model and simulations in two forms. The first one calculates the matrices and vectors of the inverse dynamics (8) required by many model-based motion controller. The second form computes a simulation model (9) (and the respective Jacobian matrix) to be used by an implicit solver of ordinary differential equation, [12]. The library is written in the `C` programming language and is freely available for download [14].

D. Control Results

The outlined trajectory tracking control in the operational space using the methods presented is implemented on the considered lightweight robot.

The basic control of the *LWR* allows to command actuation torques for each individual joint via the Fast Research

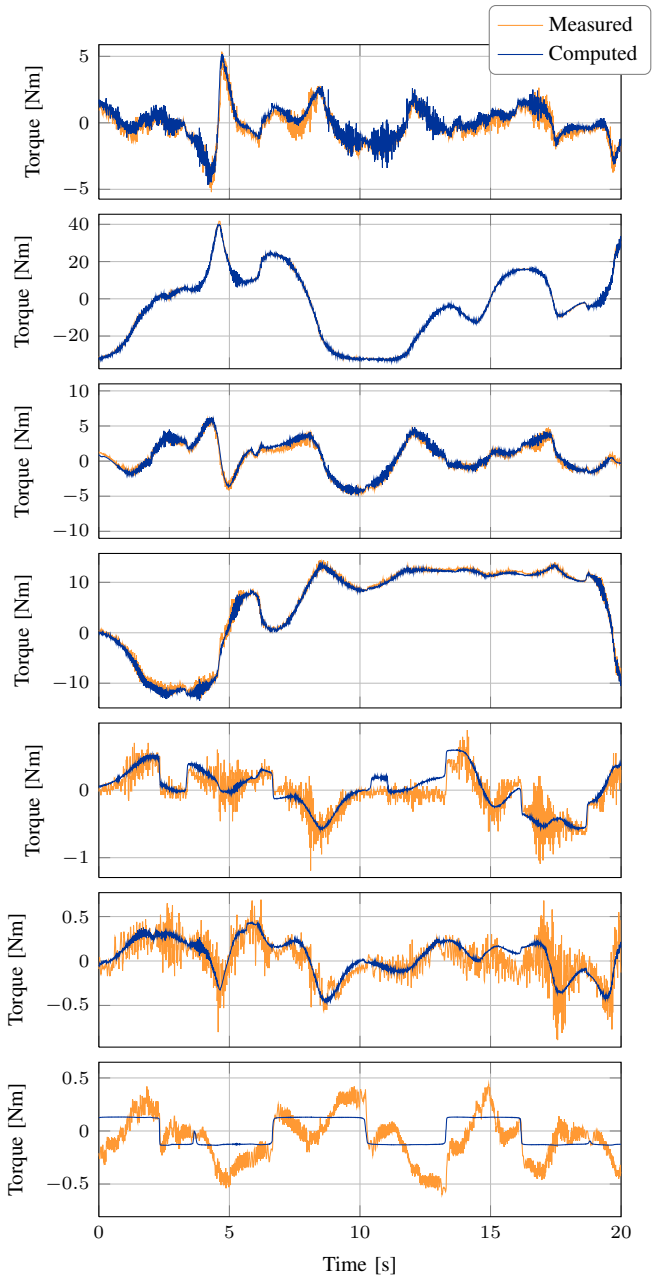

Fig. 4. Comparison of measured and computed torques for random motions. From top to bottom: joints 1 to 7.

Interface (FRI) [15]. A remote computer with an Ethernet connection is used to specify *superposed* joint torques using FRI's *joint impedance control mode*. In our particular setup, this interaction is described by

$$\tau_{\text{cmd}} = \tau_{\text{FRI}} + g_{\text{FRI}}(q), \tag{11}$$

where $\tau_{\text{FRI}} \in \mathbb{R}^n$ is the joint torque computed by the remote computer, $g_{\text{FRI}}(q)$ denotes the position-dependent gravity compensation torques computed by the *LWR*, and τ_{cmd} are the *setpoints* for the individual low-level torque controller embedded in each joint. This enables us to actually control the robot on the torque level and to bypass the internal position controllers of the robot. Note that τ_{cmd} are setpoints

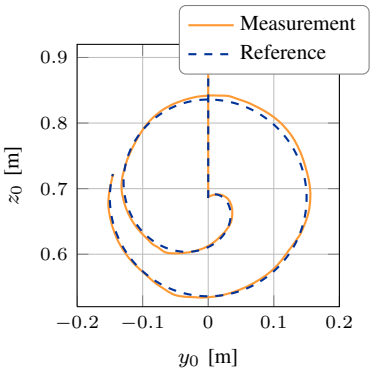

Fig. 5. Comparison of measured and desired positions of the robot end effector for the undisturbed tracking experiment.

and they differ from τ_{msr}, the actual measured torques. Ideally $\tau_{\text{msr}} = \tau_{\text{cmd}}$, however this is in general not the case. This may affect the tracking performance, as shown later.

The inverse dynamics controller is implemented on the remote computer. For the prototype implementation, Python was used, which constrained the sampling period to be not lower than $T_s = 10ms$. Note that the FRI allows to use sampling times as low as $1ms$, which will be exploited in future implementations. Our controller uses the library described in Section IV-C to compute the actuation torques to compensate for the nonlinear effects. For a sufficient approximation of the error system dynamics (10), we chose a stiffness setting of $\omega_i^2 < 1000(rad/s)^2$, [16], [17]. To avoid an additional delay from finite differentiation and low-pass filtering, a discrete Kalman filter has been implemented to estimate the joint velocities $\dot{q}$, [18].

To evaluate the controller implementation we track a circular trajectory in the Cartesian space. The circle lies on a plane parallel to the $y_0 - z_0$-plane of the robot base coordinates (refer to Fig. 1). The reference and actual path followed by the end effector are compared in Figure 5. As shown, there is an observable mismatch between the reference trajectory and the real trajectory followed by the end effector. Figures 6 and 7 show that the low-level torque controllers insufficiently apply the reference torques and that this considerably degrades the tracking performance of joints 5 to 7, in particular.

Using the same trajectory, we manually and heavily deviate the robot's end effector from its Cartesian trajectory at different points in time. The behavior in time of joint 4 under these conditions is shown in Figure 8. We see only little overshooting and the error compensation is critically damped, which goes according to our design goals, cf. (10).

Overall, note that the motion of joints 1 to 4 are very close to the reference. Furthermore, the circular end effector motion is similar to the desired Cartesian trajectory. A better tracking performance of the small joints (5 to 7) might be achieved using more aggressive controller parameters. However, this requires a lower sampling period (e.g. $T_s = 1ms$).

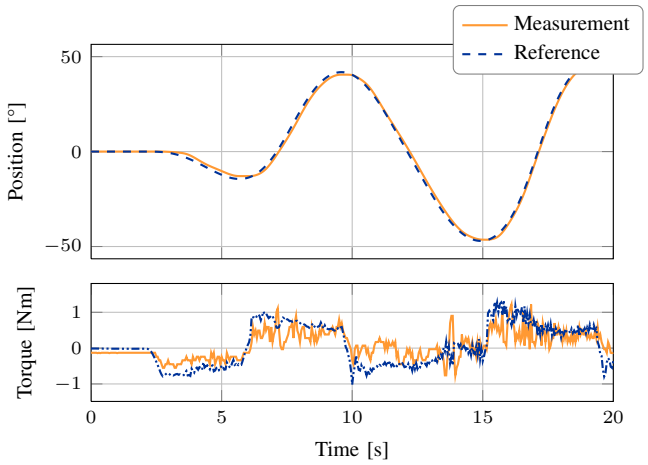

Fig. 6. Comparison of measured and desired positions of joint 1 in the tracking experiment.

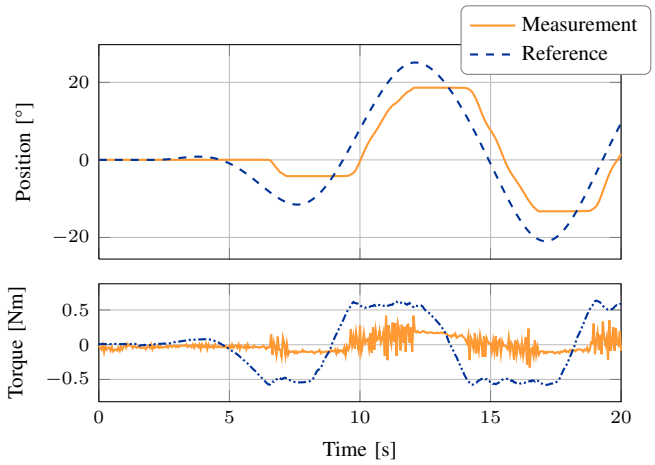

Fig. 7. Comparison of measured and desired positions of joint 6 in the tracking experiment.

V. Conclusion

In this work we outlined in a structured way the application of model-based control approaches to perform complex robotic task. Using a suitable model, we first showed how optimal joint trajectories can be found, such that the parameters of the model can be estimated sufficiently well. We then showed that after a suitable reformulation the model is well suited for model-based controller design. The main contributions are the derivation of suitable models and a corresponding toolbox for the *KUKA lightweight robot IV*, which is available for free download, the description of a suitable optimal experimental design scheme, as well as the experimental validation of a model-based controller utilizing inverse dynamics. The overall scheme of modeling, parameter estimation, and controller design is not limited to the specific robot and can easily be used for other configurations.

Future work will consider the use of the derived model for predictive path following and force feedback control. The developed library and videos about the implementation can be found in [14].

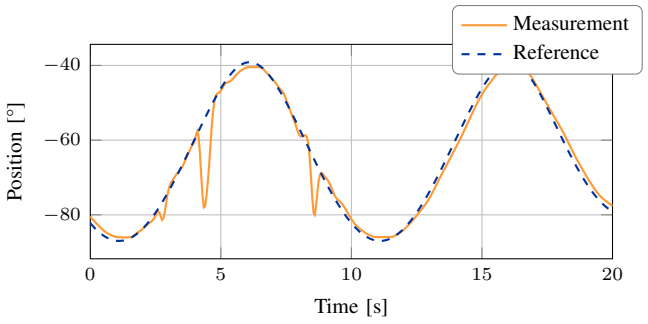

Fig. 8. Error compensation when the robot is disturbed by hand. Comparison of measured and desired position of joint 4.

References

[1] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer, 2011.

[2] R. Kelly, V. S. Davila, and J. A. L. Perez, *Control of robot manipulators in joint space*, 1st ed. Berlin, Heidelberg: Springer, 2005.

[3] W. Khalil and E. Dombre, *Modeling, Identification and Control of Robots (Kogan Page Science Paper Edition)*. Butterworth-Heinemann, 2004.

[4] C. W. Kennedy and J. P. Desai, "Modeling and control of the mitsubishi pa-10 robot arm harmonic drive system," *Mechatronics, IEEE/ASME Transactions on*, vol. 10, no. 3, pp. 263–274, 2005.

[5] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Albu-Schäffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald *et al.*, "The KUKA-DLR lightweight robot arm - a new reference platform for robotics research and manufacturing," in *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*. VDE, 2010, pp. 1–8.

[6] V. Bargsten, "Implementation of a model-based control scheme for a robotic arm," Master's thesis, Otto-von-Guericke Universitat Magdeburg, 2012.

[7] 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.

[8] J. Swevers, D. Ganseman, D. B. Tükel, J. de Schutter, and H. Van Brussel, "Additional remarks related to the paper Optimal robot excitation and identification," no. 95R65, 1997.

[9] J. Swevers, C. Ganseman, D. B. Tükel, J. de Schutter, and H. Van Brussel, "Optimal robot excitation and identification," *IEEE Trans. Robot. Autom.*, vol. 13, no. 5, pp. 730–740, 1997.

[10] J. C. Spall, *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. John Wiley & Sons, 2003.

[11] J. Slotine, W. Li *et al.*, *Applied nonlinear control*. Prentice-Hall Englewood Cliffs, NJ, 1991, vol. 199, no. 1.

[12] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward, "SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers," *ACM Trans. Math. Softw.*, vol. 31, no. 3, pp. 363–396, Sep. 2005. [Online]. Available: http://doi.acm.org/10.1145/1089014.1089020

[13] A. Isidori, *Nonlinear control systems*. Springer-Verlag New York, Inc., 1997.

[14] http://ifatwww.et.uni-magdeburg.de/syst/about_us/people/zometa/.

[15] G. Schreiber, A. Stemmer, and R. Bischoff, "The fast research interface for the KUKA lightweight robot," in *IEEE Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications How to Modify and Enhance Commercial Controllers (ICRA 2010)*, 2010.

[16] J. Lunze, *Regelungstechnik 2: Mehrgrößensysteme, Digitale Regelung*. Springer, 2008.

[17] R. Isermann, *Digitale Regelsysteme: Band 1: Grundlagen, deterministische Regelungen (German Edition)*. Springer, 1988.

[18] G. Bishop and G. Welch, "An introduction to the Kalman filter," *Proc of SIGGRAPH, Course*, vol. 8, pp. 27 599–3175, 2001.