2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

Parameter Identification for Dynamical Systems Using Optimal Control Techniques

Kai Schäfer[1,2], Margareta Runge[1], Kathrin Flaßkamp[1], and Christof Büskens[1]

Abstract— **We apply direct methods from numerical optimal control to solve nonlinear dynamic parameter identification problems. The focus is on comparing shooting methods and a full discretization approach regarding approximation quality and efficiency. For an idealized robotic system, we show that the full discretization approach is beneficial, since it is most robust to minimizing the fitting error. The methods are finally validated using data from a real-world robotic system.**

I. INTRODUCTION

In many fields of engineering and natural sciences, modeling real-world systems and technical processes is crucial for simulation and prediction, as well as for optimization and control design. Oftentimes, a system's dynamical behavior is qualitatively known from first principles, e.g. physical laws in mechanics. It is then possible to make reasonable assumptions on the structure of differential equations. Therefore, system identification comes down to identifying the unknown parameters in order to generate quantitatively valid models. To give an example, consider a mechanical system such as a multi-link robot. There are several well established methods to generate equations of motions (e.g. Euler-Lagrange, Newton-Euler, or Hamilton equations) for its dynamics. Still, the modeling equations depend on parameters: the masses, lengths, and friction coefficients of the robot links and joints, for instance. These parameters have to be determined before the model can be used for simulation, optimization, or control. This is done by performing suitable experiments on the real system and collecting data via sensors.

Alternatively, it is also possible to build models from data only, i.e. using an arbitrary function space representation. This is less in the scope of this paper though. However, for complex systems with interacting subsystems, there will always be parts for which less sophisticated model structures are known and it is therefore necessary to rely on generic approaches such as linear ODE models.

The aim of parameter identification is to find model parameters which lead to a model output that approximates given measurements best. There exist classical approaches to this type of estimation problem, e.g. Kalman filtering or maximum-likelihood estimation [1], [2]. In general, i.e. for an arbitrary choice of optimization objective and model representation, parameter identification can be performed by solving nonlinear optimization (NLP) problems [3]. We follow the NLP approach in our paper in order to be flexible in choosing the cost function and the nonlinear dynamic system models. However, first and foremost, we are interested in studying the influence of the transcription method that has to be applied in order to transform the model equations into NLP constraints (see Section III for details).

a) Related work: Several related problem settings, Kalman filtering and maximum-likelihood estimation, among others, are presented in the textbook [2]. Identification of nonlinear dynamical systems is considered in [3], [4]. For multi-link robots, a sequential estimation of link parameters is common [5]. A focus on inertial parameters can be found in [6]. Since we are interested in general parameter identification techniques, we perform simultaneous identification for our robotic test systems in this work. Nonlinear systems may show qualitatively different dynamical behavior in different scenarios. Thus, finding good, i.e. informative trajectories (described for robotics e.g. in [7]), is an important aspect, which can also be addressed by optimization [8].

b) Contributions: In this paper, we apply transcription methods known from optimal control, namely full discretization and single or multiple shooting (see e.g. [9], [10], [11]) to parameter identification problems. We compare the resulting NLP problems analytically regarding complexity, i.e. size and sparsity structure. For numerical comparisons of the methods, we use an academic example of a two-link robot. The NLP problem formulations are solved by WORHP [12]. We show that full discretization with a first-order integration scheme is an interesting alternative to shooting techniques, despite its higher number of optimization parameters. In many cases, full discretization indeed yields better identification results, i.e. lowest fitting error, compared to single and multiple shooting with few nodes, for which the optimizer oftentimes returns bad local minima. The methods are validated in our experiments with a real industrial robot.

c) Outline: The remainder of this paper is organized as follows: A formal problem setting is introduced in Section II. Section III provides an overview of transcription methods from optimal control. We discuss the different optimization problems together with the NLP solver WORHP, which we use, in Section IV. Then, the methods are applied first to the academic example of an idealized robot and secondly to a real-world system in Section V. Section VI gives concluding remarks and an outlook to future work.

II. PROBLEM FORMULATION

Let measurements $x_i^d \in \mathbb{R}^{n_d}$ of a real-world system be given at discrete time points $t_0 = t_1 < t_2 < \cdots < t_N = t_f$ and assume a corresponding dynamic model is given by

$$\dot{x}(t) = f(t, x(t), p), \quad t \in [t_0, t_f],$$

[1]Center for Industrial Mathematics, University of Bremen, Germany
[2]Corresponding author, kai.schaefer@uni-bremen.de

978-3-9524-2699-9 ©2018 EUCA

where $x(t) \in \mathbb{R}^{n_x}$ denotes the states, $f : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ the vector field and $p \in \mathbb{R}^{n_p}$ a set of unknown parameters. We assume f to be sufficiently regular (e.g. Lipschitz-continuous in x), such that there exist unique solutions for all initial values x_0 and all parameter values $p \in \mathbb{R}^{n_p}$.

The goal of parameter identification is to fit the model output to the data by identifying parameters p which lead to state trajectories $x \in C^1([t_0, t_f], \mathbb{R}^{n_x})$ that approximate the given measurements best (in least squares sense). We formulate this task in the following way:

Nonlinear dynamic parameter identification problem (NLDPIP):

$$\underset{x(\cdot),x_0,p}{\text{Minimize}} \qquad \frac{1}{N}\sum_{i=1}^{N} \|c(x(t_i)) - x_i^d\|_2^2 \tag{1}$$

$$\text{subject to} \qquad \dot{x}(t) = f(t, x(t), p), \quad t \in [t_0, t_f], \tag{2}$$

$$x(t_0) = x_0, \tag{3}$$

where $c\colon \mathbb{R}^{n_x} \to \mathbb{R}^{n_d}$ denotes the system output, e.g. a projection of states to the ones for which measurements are available. Here, we seek to minimize the Euclidean distance between the simulated model output and the observed measurements. Since our method does not require error models of certain type, we do not make specific assumptions on measurement noise at this point. Future work might take into account special error structures (e.g. normal distribution).

An NLDPIP is an *infinite dimensional optimization problem*, since the system state $x \in C^1([t_0, t_f], \mathbb{R}^{n_x})$ is part of the optimization variables. Therefore, the problem has to be approximated such that standard NLP solvers can be applied. This is analogously to optimal control problems, in which states and controls form infinite dimensional optimization variables that have to be approximated by time discretization (see e.g. [9]). Therefore, we apply different methods which are well known in numerical optimal control to our parameter identification problem, as described in the following section. In the end, this yields an NLP problem of this general form:

Nonlinear constrained optimization problem (NLP):

$$\begin{aligned} &\underset{Y}{\text{Minimize}} && r(Y) \\ &\text{subject to} && h(Y) \leq 0, \\ &&& g(Y) = 0. \end{aligned}$$

In addition to equality constraints $g(Y) = 0$ one can formulate inequality constraints $h(Y) \leq 0$, if desirable for specific problem instances.

III. METHODS

In general, it is not possible to solve the ODE (2) of an NLDPIP analytically. Thus, it has to be approximated, in other words, simulated, by numerical integration techniques. This is equivalent to a discretization in time, i.e. defining a grid $\{t_i\}_{i=1}^N$ on the interval $[t_0, t_f]$ and approximating $x \in C^1([t_0, t_f], \mathbb{R}^{n_x})$ by a set of vectors $\{x_i\}_{i=1}^N$, it should hold that $x_i \approx x(t_i)$ for all $1 \leq i \leq N$.

A. Single shooting

A simple and intuitive way of discretization is the single shooting or initial value approach. The idea is to combine NLP and ODE solvers: Whenever the NLP solver needs to evaluate $x(t)$ at some time point t, the ODE is solved by an external integration method, e.g. by a Runge-Kutta scheme.

The NLP can then be formulated like this:

$$\min_{x_1,p} \quad \frac{1}{N}\sum_{i=1}^{N} \|c(x(t_i; p)) - x_i^d\|_2^2.$$

Since a solution of the ODE is uniquely determined by fixed parameters and a given initial value, only these values are defined as optimization variables. To evaluate the cost function, the state at the data time points is needed though. Thus, $x(t_1; p) = x_1$ and $x(t_i; p)$ is the solution from integrating the ODE up to time t_i, $2 \leq i \leq N$.

This method has the advantage of a small number of optimization variables, as we will later see when comparing it to the other methods. Further, one can make use of high-order (adaptive) integration methods. However, this method lacks robustness (as indicated in [4]), since the solution of the ODE might depend heavily on the parameters. A poor initial guess of those might lead to divergence of the integration during the optimization.

In other words, this unconstrained optimization problem exhibits a highly nonlinear objective function due to the complex calculation of $x(t_i; p)$. One way of reducing this nonlinear behavior and overcoming numerical instabilities is introducing nodes at which the integration is restarted, as it is presented in the following.

B. Multiple shooting

The idea of (direct) multiple shooting (see e.g. [11] for optimal control or [10], [4] for parameter identification) is to subdivide the time interval $[t_0, t_f]$ into subintervals by introducing so called *multinodes* τ_j, $j = 1, \ldots, M$. The ODE is still, as in single shooting, solved by an external integration method, but only on the smaller subintervals, restarting on every multinode. The additional initial values $x(\tau_j)$ become optimization variables x_{τ_j} $(j = 1, \ldots, M)$, therefore the parameter space of the NLP increases (dependent on the chosen number of multinodes). In order to receive a continuous solution of the ODE at the end of the optimization process, continuity constraints (cf. Eq. (5)) are added to the problem which guarantee that end points of subintervals match subsequent initial values. The problem can thus be formulated as

$$\min_{Y} \quad \frac{1}{N}\sum_{i=1}^{N} \|c(x(t_i; p, x_{\tau_{\sigma(i)}})) - x_i^d\|_2^2 \tag{4}$$

$$s.t. \quad x(\tau_{j+1}; p, x_{\tau_j}) - x_{\tau_{j+1}} = 0 \quad \text{for } 1 \leq j \leq M-1 \tag{5}$$

with $Y = [x_{\tau_1}, \ldots, x_{\tau_M}, p]$. The expression $x(t_i; p, x_{\tau_{\sigma(i)}})$ describes the solution from integrating the ODE up to time t_i with initial value $x_{\tau_{\sigma(i)}}$, in which $\sigma(i) := \max_{1 \leq j \leq M}\{j : \tau_j \leq t_i\}$. Due to the structure of the constraints, their derivatives show a characteristic sparsity structure, which we are going to discuss in Section IV-B.

C. Full discretization

In full discretization, contrarily to shooting methods, the process of numerically solving the ODE is formulated as additional equality constraints for the NLP. This turns the states at each discretization point into optimization variables, such that the problem then reads

$$\begin{aligned} \min_{Y} \quad & \frac{1}{N}\sum_{i=1}^{N} \|c(x_i) - x_i^d\|_2^2 \\ s.t. \quad & x_i + h \cdot \Phi(t_i, t_{i+1}, x_i, x_{i+1}, p) - x_{i+1} = 0 \\ & \qquad\qquad\qquad \text{for } 1 \le i \le N-1 \end{aligned}$$

with $Y = [x_1, \ldots, x_N, p]$. Here, $\Phi(t_i, t_{i+1}, x_i, x_{i+1}, p)$, for ease of notation, serves as a place holder for any explicit or implicit one-step method, whereas other choices, such as multi-step schemes, could be applied as well. For portrayal reasons, we choose the number of data and discretization points N to coincide. The NLP becomes large-scale, but the corresponding Jacobian or Hessian matrix becomes sparse due to the specific structures in the problem formulation.

For comparison, we list the number of variables and the number of constraints of all three methods in Table I. Recall that n_x is the state dimension, n_p the number of parameters, N the number of time discretization points and M the number of multinodes.

IV. OPTIMIZATION

NLP solvers are typically based on sequential quadratic programming (SQP) or interior point (IP) techniques. State-of-the-art implementations, which handle high-dimensional problems by sparsity exploitation, are IPOPT ([13]), KNITRO ([14]), SNOPT ([15]), and WORHP, for instance. Our method of choice is WORHP ("We Optimize Really Huge Problems", cf. [12]).

A. WORHP

WORHP has been developed as the official NLP solver for ESA, the European Space Agency, and can be obtained via `www.worhp.de`. It provides an SQP and an IP algorithm, where we make use of the former. Thus, derivatives of r and g (and h, if applicable) of the NLP have to be provided either by numerical approximation or analytical expressions.

The software module *TransWORHP* offers numerical methods for optimal control (cf. [16]) using full discretization or single and multiple shooting techniques (cf. Section III) for various integration schemes. As it is crucial for large scale optimization problems, WORHP exploits sparsity structures in the derivatives.

TABLE I: Comparison of methods.

Method	$n := \dim(Y)$	$m := \dim(g)$
Single shooting	$n_x + n_p$	0
Multiple shooting	$n_x \cdot M + n_p$	$n_x \cdot M$
Full discretization	$n_x \cdot N + n_p$	$n_x \cdot N$

B. Sparsity exploitation for parameter identification

The sparsity structure of an NLP strongly depends on the chosen transcription method for the NLDPIP. As it can be directly seen, every component of the set of equality constraints of the multiple shooting and the full discretization method only depends on few variables. This defines the sparsity of the constraint's Jacobian Dg and from similar arguments, the structure of the gradient Dr can be deduced. The sparsity structure then takes over to second order derivatives, i.e. to the NLP-Lagrangian's Hessian DH.

The multiple shooting technique raises the question whether there is an optimal number of shooting nodes. Here, "optimal" might refer to all kinds of criteria, but we are in particular interested in those that are independent of the example system. Thus, in [17], the *number of nonzero elements* (nnz) of the gradient Dr, the Jacobian Dg, and the Hessian DH of optimal control problems is analyzed and an equation for the number of multinodes that lead to a minimum of nnz in Dg and DH, respectively, is given. Following, an increased number of multinodes can lead to an improvement regarding computing times, while this is not correct for parameter identification, since the problem formulations qualitatively differ in this regard. This is due to the absence of control variables. Details are omitted due to space limitations, but will be included in future work.

For NLPs of type (4), one can show that nnz is strictly monotonically increasing when the number of multinodes is increased. Therefore, we consider the *relative density* rd of the derivatives, i.e.

$$rd(Dr) = \frac{nnz(Dr)}{n}, \quad rd(Dg) = \frac{nnz(Dg)}{n \cdot m},$$

and since the Hessian matrix is symmetric, we have

$$rd(DH) = \frac{nnz(DH)}{0.5 \cdot n \cdot (n+1)}.$$

We illustrate the relative density for increasing numbers of multinodes in Section V-A for an idealized robotic system.

Typically, the problem size, i.e. the time-discretization step size and, if applicable, number of multinodes, has to be chosen as a tradeoff between computational complexity and accuracy of the solution. As we show numerically in Section V-A the shape of the results for the robot indicates that there is no need to increase the number of multinodes up to its maximum (defined by integration step size).

We exclude the influence of integration schemes from this analysis. That is, we choose the same integration schemes (with identical fixed step sizes) for defining the constraints in full discretization as we use for forward integration in the shooting methods. Future work will include the order and step size of different integration schemes in our analysis.

V. APPLICATION

At first, we compare the methods numerically regarding an idealized robotic system with two degrees of freedom. The quality of the data fitting serves as main criterion for comparison. Here, we make use of simulated data generated by chosen parameters. In a next step, first results regarding the parameter identification for the 6-link industrial robot DENSO VS-050 are presented.

A. Idealized robotic system

In order to demonstrate and analyze the methods presented in Section III we use an academic example of an idealized robotic system with two degrees of freedom. This robot's two links have cylindric shapes and are connected via joints, which can rotate around the z-axis (joint 1) and around the y-axis (joint 2), as depicted in Fig. 1. We assume that the joints can be controlled externally, e.g. by torques or currents. For the derivation of the equations of motion, we use the Euler-Lagrange approach and we choose the joint angles $\theta = (\theta_1, \theta_2)^T$ to serve as generalized coordinates and the corresponding joint angular velocities $\dot{\theta} = (\dot{\theta}_1, \dot{\theta}_2)^T$ as their derivatives. We can formulate the Lagrange function as difference between kinetic and potential energy in the following way:

$$L(\theta, \dot{\theta}) = \frac{1}{2}\sum_{i=1}^{2} m_i \|\dot{S}_i\|^2 + \frac{1}{2}\sum_{i=1}^{2} \omega_i^T \mathcal{I}_i \omega_i - \sum_{i=1}^{2} m_i g S_{i,z},$$

in which S_i is the center of mass of link i relative to the base frame, ω_i is the angular velocity of link i relative to the base frame, $\mathcal{I}_i$ the inertia tensor of link i and g the gravitational constant. However, we abstain from a detailed derivation of the dynamics and instead refer to [18]. The dynamic behavior is then described by forced Euler-Lagrange Equations

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} = f_L$$

with an external forcing term

$$f_L(\dot{\theta}, p, I) = -\kappa_1 \tanh(a\dot{\theta}) - \kappa_2 \dot{\theta} + \kappa_3 I,$$

in which I represents the vector of currents flowing through the joint motors. These equations can finally be written in the common form

$$M(\theta, p)\ddot{\theta} = F(\theta, \dot{\theta}, p) + f_L(\dot{\theta}, p, I).$$

It is notable that these equations are highly nonlinear due to the complex dynamics in robotics. The model terms are dependent of time-independent (physical or modeling) parameters p. All in all, there are 19 parameters which we want to identify simultaneously. In order to create artificial measurements we choose an arbitrary set of parameters (see Table II) and arbitrary inputs I for the currents. We then simulate the system for a given time interval with a high-order integration method in combination with a small step size. As optimization criterion we choose to approximate the simulated joint angles θ^d and joint angular velocities z^d. The task is to find states $\theta(\cdot)$ and $z(\cdot)$, initial values θ_0 and z_0, and parameters p that

$$\begin{aligned}
\text{minimize} \quad & J = \frac{1}{N}\sum_{i=1}^{N} \left(\|\theta(t_i) - \theta_i^d\|_2^2 + \|z(t_i) - z_i^d\|_2^2 \right) \\
\text{subject to} \quad & \dot{\theta}(t) = z(t), \\
& \dot{z}(t) = M^{-1}(\theta(t), p) \\
& \qquad \cdot \left[F(\theta(t), z(t), p) + f_L(z(t), p, I) \right], \\
& \theta(t_0) = \theta_0, \quad z(t_0) = z_0
\end{aligned}$$

for $t \in [t_0, t_f]$. This can be reformulated as an NLP with the methods presented in Section III and solved with WORHP. We additionally set box constraints for the parameters ($p_i^{min} \leq p_i \leq p_i^{max} \;\forall i \in \{1, \dots, n_p\}$). Since the measurements are known, we can use them as initial guesses for the state variables and we set the initial parameters arbitrarily, but in their respective order of magnitude. For these specific calculations, second-order information is approximated by using BFGS methods (see [12]). In order to keep calculations comparable among each other, we use the explicit Euler method with a constant step size in all of the applied methods (although TransWORHP allows the use of higher order integration schemes).

Fig. 2 shows the simulated data and the state trajectories resulting from the solution of the NLP with a multiple shooting using five multinodes and a full discretization. Both of the methods converge to a local minimum, but the solution from full discretization obviously ($J = 6.5 \cdot 10^{-2}$ against $J = 5.016$) finds a better one.

When having a closer look at the performance of multiple shooting regarding the fitting quality, we are mainly interested in the number of multinodes we have to introduce in order to find solutions that approximate the data well. Therefore we solved the NLP by using M multinodes with $M \in \{2, \dots, 301\}$ (distributed equidistantly on the time interval). In Fig. 3 we see the respective objective values

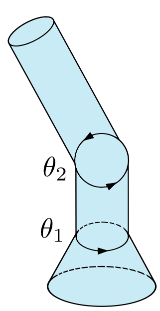

Fig. 1: Illustration of the idealized robot.

TABLE II: Parameters used to create measurements.

Description	Symbol	Link 1	Link 2	Unit
Mass	m	6	23	kg
Center of mass: x	s_x	0	0	m
Center of mass: y	s_y	0	0	m
Center of mass: z	s_z	/	0.2875	m
Moment of inertia: x	$\mathcal{I}_x$	/	0.6912	kgm^2
Moment of inertia: y	$\mathcal{I}_y$	/	0.6912	kgm^2
Moment of inertia: z	$\mathcal{I}_z$	0.03	0.115	kgm^2
Factor (viscous friction)	κ_1	1	2	Nm/s
Factor (dry friction)	κ_2	8	3	Nm
Current factor	κ_3	0.7	0.4	–
Modeling factor	a	50	50	–

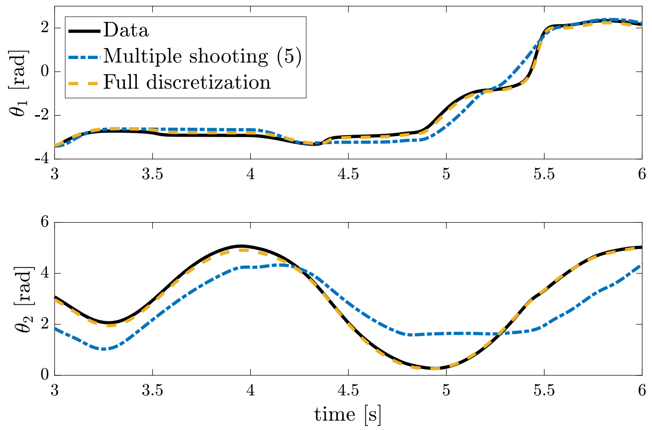

Fig. 2: Data and solution trajectories of joint angles θ.

together with the values from single shooting and full discretization. The figure shows a typical behavior (in the sense of multiple experiments using different initial parameters): By introducing additional multinodes (i.e. increasing the number of variables and constraints, but thereby decreasing the objective function's complexity) the solver is more likely to find better local minima compared to using few multinodes. This observation encourages the search for an "optimal" number of multinodes (as already mentioned in Section IV). It is notable that solutions with similar objective values do not correspond to identical values in parameter space. In fact, we observed that different sets of parameters can lead to very similar motions.

As already described in Section IV, the solver WORHP is able to exploit sparsity structures originating from different transcription methods. By increasing the number of multinodes, we increase the problem dimensions of the NLP, but due to the specific structure in the new constraints and objective function, the relative densities of the Jacobian and Hessian matrix (as introduced in IV-B) decrease. This is illustrated in Fig. 4. We can observe that there is a high change in the relative densities when using only a few multinodes, which might be one reason for the method to find strongly varying local minima (see Fig. 3).

B. Real-world application

We consider the 6-link robot DENSO VS-050 (see Fig. 5), which finds applications in many industrial fields. The aim is to model its dynamic behavior with the methods presented in Section III, i.e. to find parameters in the model formulation for which model output and measurements coincide best. In contrast to our idealized robot, this one does not have nice shapes (in a modeling sense) and especially a higher number of links and joints, respectively, which (among other factors) makes the parameter identification a challenging task. We focus on parameters of the first two links and therefore assume that the links three to six serve as an extension of link two, i.e. the respective joints are fixed. Still, the resulting problem formulation differs from our idealized one by some additional terms and parameters due to modeling. We are further able to measure real data via the robot's sensors, which makes our methods applicable as before.

Thus, we formulate the NLDPIP, apply the methods and solve the corresponding NLP with WORHP. Due to the very general formulation as an NLP, we can easily include box-constraints limiting the physical movement or add parameter constraints representing physical dependencies. Here, we use the joint angles and joint angular velocities as fitting criteria and apply the full discretization method, in which the ODE is solved with a trapezoidal rule. Fig. 6 shows the obtained optimal state trajectories, which approximate the measurements qualitatively sufficient.

Since we are interested in finding a set of parameters that is applicable not only for one scenario, but in a more general case, we validate the identified parameters by simulating the system for another experimental measurement, which is shown in Fig. 7. The identified parameters lead to a model output (using again the trapezoidal rule) that describes the real data well. However, the validation fails for qualitatively different measurements, which might be explained by providing too few information in the data (see also Section VI). Nevertheless, these first results validate our approach in principle.

VI. CONCLUSION

This paper considers parameter identification problems of nonlinear dynamic systems. That is, we assume a system model is known up to a set of parameter values which has to be determined based on experimental measurement data. The

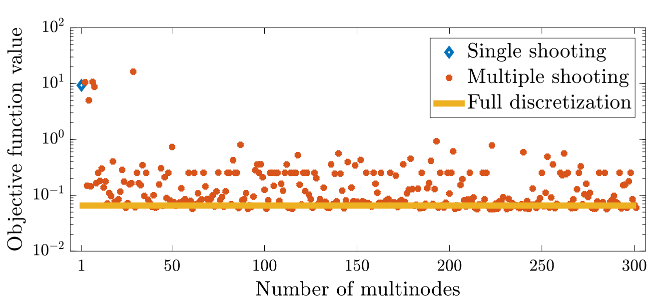

Fig. 3: Semilogarithmic representation of objective values for an increasing number of multinodes.

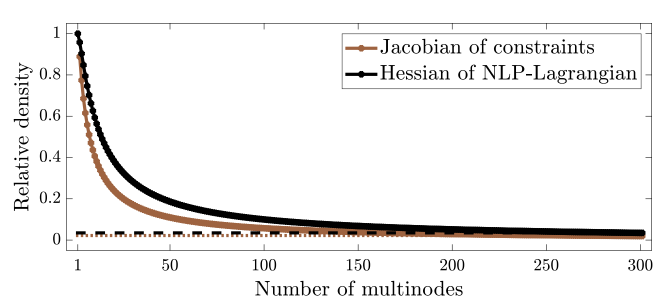

Fig. 4: Relative densities of NLP matrices. Dashed/dotted lines: Relative densities for full discretization.

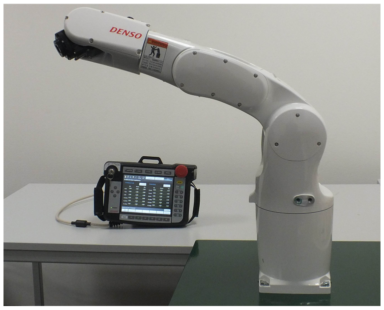

Fig. 5: Industrial robot DENSO VS-050.

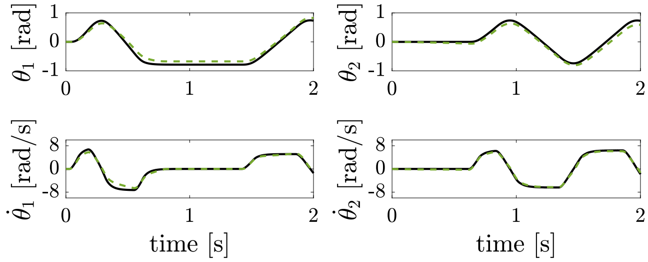

Fig. 7: Validation results: Real measurements (black) and model output (green dashed).

parameter fitting problem is cast into a constrained optimization problem with least-squares-type cost function and the system equations as constraints. Since the problem strongly resembles an optimal control problem, direct methods from numerical optimal control, i.e. single/multiple shooting and full discretization, are applied. Unlike in optimal control, there is no optimal number of multinodes if only sparsity is considered. The case studies for an academic system and a real multi-link robot indicate that full discretization is preferable to shooting, when both accuracy of solutions and sparsity structure exploitation are considered. Future work has to address locality of our current approach, i.e. we observe that sometimes, well fitted parameters for one scenario lead to poor results on other, qualitatively different scenarios. This issue can be addressed by searching for trajectories, which provide optimal information about the system's dynamics (cf. e.g. [7], [8]). Moreover, the role of the integration method within shooting and full discretization, as well as redundancy in parameter space and the role of multiple local optima have to be studied more closely.

ACKNOWLEDGMENT

K. Schäfer and M. Runge acknowledge funding by the Deutsche Forschungsgemeinschaft (DFG) within the framework of GRK 2224/1.

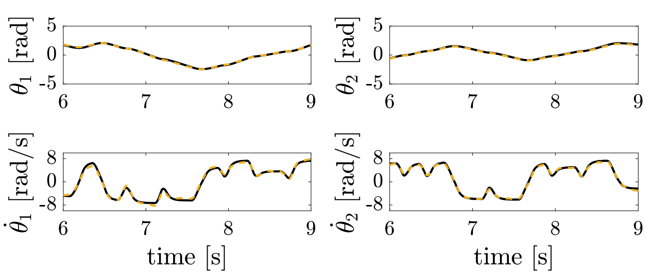

Fig. 6: Parameter identification results: Real measurements (black) and obtained trajectories (yellow dashed).

REFERENCES

[1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[2] T. K. Yaakov Bar-Shalom, X.-Rong Li, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. John Wiley & Sons, Inc., 2002.

[3] K. Schittkowski, *Numerical Data Fitting in Dynamical Systems*. Kluwer Academic Publisher, 2002.

[4] H. U. Voss, J. Timmer, and J. Kurths, "Nonlinear dynamical system identification from uncertain and indirect measurements," *International Journal of Bifurcation and Chaos*, vol. 14, no. 06, pp. 1905–1933, 2004.

[5] P. Khosla and T. Kanade, "Parameter identification of robot dynamics," in *Proceedings of the 24th IEEE Conference on Decision and Control*, 1985, pp. 1754–1760.

[6] M. Gautier and W. Khalil, "Direct calculation of minimum set of inertial parameters of serial robots," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 3, pp. 368–373, June 1990.

[7] J. Swevers, C. Ganseman, D. B. Tükel, J. D. Schutter, and H. V. Brussel, "Optimal robot excitation and identification," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 5, 1997.

[8] A. D. Wilson, J. A. Schultz, and T. D. Murphey, "Trajectory optimization for well-conditioned parameter estimation," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 1, 2015.

[9] M. Gerdts, *Optimal Control of ODEs and DAEs*. De Gruyter, 2011.

[10] H. Bock, "Numerical treatment of inverse problems in chemical reaction kinetics," in *Modelling of Chemical Reaction Systems*, ser. Springer Series in Chemical Physics, K. Ebert, P. Deuflhard, and W. Jäger, Eds. Heidelberg: Springer, 1981, vol. 18, pp. 102–125.

[11] H. Bock and K. Plitt, "A Multiple Shooting algorithm for direct solution of optimal control problems," in *Proceedings of the 9th IFAC World Congress*. Budapest: Pergamon Press, 1984, pp. 242–247.

[12] C. Büskens and D. Wassel, "The ESA NLP solver WORHP," in *Modeling and optimization in space engineering*. Springer, 2012, pp. 85–110.

[13] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[14] R. H. Byrd, J. Nocedal, R. Waltz, and A. Zaslavski, *Knitro: An Integrated Package for Nonlinear Optimization*. Springer, 2006, pp. 35–59.

[15] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Rev.*, vol. 47, no. 1, pp. 99–131, 2005.

[16] C. Büskens and M. Knauer, "From WORHP to TransWORHP," in *Proceedings of the 5th International Conference on Astrodynamics Tools and Techniques*, May 2012.

[17] C. Büskens, "Optimization methods and sensitivity analysis for optimal control problems with control and state constraints," Ph.D. dissertation, University of Münster, 1998.

[18] R. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 1994.