2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Dynamic Parameter Estimation Utilizing Optimized Trajectories

Argtim Tika[1], Jonas Ulmen[2] and Naim Bajcinca[3]

***Abstract*— We suggest a procedure for dynamic parameter estimation of serial robot manipulators. Its basic idea relies on the synthesis of an optimal manipulation trajectory, which is based on properly introduced parameter aggregates to ensure a collection of numerically well-conditioned data-sets, yielding an accurate computation of parameter estimates. The optimal trajectory itself is computed by using a memetic algorithm, which represents a metaheuristic combination of genetic and gradient based algorithms. The algorithm is experimentally verified by estimating the parameters of the UR5 robot by Universal Robots.**

I. INTRODUCTION

In advanced use cases of industrial robots, one is typically enforced to apply model-based control techniques, such as computed torque control that rely on accurate mathematical modeling of the robot manipulator at hand. These approaches are often afflicted with insufficiently known values of dynamical parameters involving inertia elements, friction parameters, etc. Dynamic parameters are generally provided by the robot manufacturers in form of data sheets, but these parameters are usually related to specific usability scenarios but may not cover the versatility of all scenarios, e.g. in advanced application scenarios or are not sufficiently detailed. As a consequence, the estimation of the dynamic parameters is often a mandatory step in the design of proper control policies.

In this paper, we present a general procedure for parameter estimation of robot manipulators, which makes use of a formulation of the dynamic robot models in terms of symbolic parameter aggregates, i.e. a set of symbolic expressions involving multiple parameters. Roughly speaking, in this approach, one strives to construct proper manipulation trajectories that enable collections of data sets which are suitable for an accurate estimation of all involved parameters. It is important to emphasize that the results of the identification procedure are highly dependent on the synthesis of proper identifying manipulation trajectories which, typically, amount to proper formulations of the corresponding optimization problems. For the cost function the 2-norm condition number of the regressor matrix is commonly chosen. Due to the periodic and non-convex nature of the cost functions and constraints, the optimization problem generally produces a large number of local minima. Some approaches apply gradient based numerical local solvers [1], sometimes in conjunction with multiple starting points [2], [3]. Genetic algorithms (GA) have been also used for finding the global minima in certain problem classes, as they are able to cover the whole search space [4], [5], [6]. In the present paper, we combine the advantages of the latter approaches by suggesting a customized metaheuristic algorithm, called the *memetic algorithm*. Memetic algorithms (MAs - also known as "Hybrid Evolutionary Algorithms" (hybrid EAs)) combine different optimization ideas, such as population based methods used in genetic algorithms and local search methods, while trying to exploit any given knowledge about the problem [7]. We study the benefits of this computation technique in obtaining excitation trajectories that produce a highly accurate parameter estimation. As a testbed for verifying our procedure, we utilize the 6-axis industrial collaborative robot UR5.

The paper unfolds as follows: First, we construct a mathematical model for a serial manipulator in Section II in order to obtain an identification model, which is linear in terms of parameter aggregate expressions in Section III. In Section IV we state the optimization problem which is then solved by the proposed memetic algorithm. In Section V we test the procedure estimating the parameters of the UR5 and Section VI concludes the paper.

II. DYNAMIC MODEL OF SERIAL MANIPULATOR

A. Geometric Model

A serial manipulator can be described solely by the connection of its links, e.g. by the modified Denavit-Hartenberg (DH)-Convention in Khalil/Kleinfinger notation [8]. The geometric model of serial manipulator is defined by four DH-Parameters per link:

- α_j: the angle between z_{j-1} and z_j about x_{j-1};
- a_j: the distance between z_{j-1} and z_j along x_{j-1};
- θ_j: the angle between x_{j-1} and x_j about z_j;
- d_j: the distance between x_{j-1} and x_j along z_j.

B. Kinematic Model

To every link we attach a coordinate frame. The transformation matrix defining frame R_j relative to frame R_{j-1} is given as

$$
\begin{aligned}
{}^{j-1}\mathbf{T}_j &= \mathbf{Rot}(x,\alpha_j)\ \mathbf{Trans}(x,a_j)\ \mathbf{Rot}(z,d_j)\ \mathbf{Trans}(z,d_j) \\
&= \begin{bmatrix} C\theta_j & S\theta_j & 0 & a_j \\ C\alpha_j S\theta_j & C\alpha_j C\theta_j & -S\alpha_j & -d_j S\alpha_j \\ S\alpha_j S\theta_j & S\alpha_j C\theta_j & C\alpha_j & d_j C\alpha_j \\ 0 & 0 & 0 & 1 \end{bmatrix},
\end{aligned} \tag{1}
$$

with $S(\cdot) := \sin(\cdot)$ and $C(\cdot) := \cos(\cdot)$.

[1]Argtim Tika and [3]Naim Bajcinca are with the Department of Mechanical and Process Engineering, Technische Universität Kaiserslautern, Germany, argtim.tika@mv.uni-kl.de, naim.bajcinca@mv.uni-kl.de

[2] Jonas Ulmen is a graduate student of the TU Kaiserslautern, Germany, ulmen@rhrk.uni-kl.de

C. Dynamic Model

The inverse dynamics of a n-link serial manipulator is described by

$$\tau = \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}), \tag{2}$$

where $\mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ is the nonlinear function of motion of the manipulator, $\mathbf{q}$, $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$ are the generalized position, velocity and acceleration of the respective link joints and τ are the produced torques required for that motion. This equation can be derived by the Newton-Euler-Algorithm in the following form:

$$\tau = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{G}(\mathbf{q}, \dot{\mathbf{q}}). \tag{3}$$

$\mathbf{M}(\mathbf{q})$ is the so called mass matrix and $\mathbf{G}(\mathbf{q}, \dot{\mathbf{q}})$ is a vector containing the terms which depend on gravity, coriolis and centrifugal forces. The Newton-Euler equations, giving the forces and moments of link j at the origin of frame R_j, are given as:

$$\mathbf{F}_j = m_j \mathbf{V}_j + \dot{\omega}_j \times \mathbf{P}_j + (\omega_j \times (\omega_j \times \mathbf{P}_j)), \tag{4}$$

$$\mathbf{M}_j = \mathbf{J}_j \dot{\omega}_j + \omega_j \times (\mathbf{J}_j \omega_j) + \mathbf{P}_j \times \dot{\mathbf{V}}_j. \tag{5}$$

$\mathbf{F}_j$ are the forces and $\mathbf{M}_j$ are the moments acting on joint j. $\mathbf{V}_j$ is the linear and ω_j is the angular velocity. $\dot{\omega}_j$ and $\dot{\mathbf{V}}_j$ are the respective accelerations. The inertial parameters are:

- m_j: link mass of link j;
- $\mathbf{P}_j$: the first moments of link j, consisting of $(\mathbf{c}_j m_j)$, the vector $\mathbf{c}_j$, pointing from frame R_j to the center of mass of j multiplied with the link mass:

$$\mathbf{P}_j = \begin{bmatrix} P_{x_j} & P_{y_j} & P_{z_j} \end{bmatrix}^T = \begin{bmatrix} c_{x_j} m_j & c_{y_j} m_j & c_{z_j} m_j \end{bmatrix}^T \tag{6}$$

- $\mathbf{J}_j$: the inertial tensor from the origin of R_j, which is constructed as follows:

$$\mathbf{J}_j = \begin{bmatrix} J_{xx_j} & J_{xy_j} & J_{xz_j} \\ J_{xy_j} & J_{yy_j} & J_{yz_j} \\ J_{xz_j} & J_{yz_j} & J_{zz_j} \end{bmatrix}. \tag{7}$$

Note that these equations are formulated in the link's inertial frame, which results in linearity in the dynamic parameters [9], [10].

D. Extending the Dynamic Model: Friction and Rotor Inertia

To model the robot accurately, friction and rotor inertia cannot be neglected [11]. There are many friction models to choose from. We decide to use the classic model, describing the torques for every joint j exerted by friction as

$$\tau_{f,j} = F_{c,j} \operatorname{sgn}(\dot{q}_j) + F_{v,j} \dot{q}_j, \tag{8}$$

where $F_{c,j}$ are the coulomb and $F_{v,j}$ the viscous friction coefficients. The friction torques can be added linearly to the dynamic model.

Geared motors and transmission systems, which are build into every link of the robot, are able to produce measurable torques. For every link this portion is summarized by

$$\tau_{A,j} = A_j \ \ddot{q}_j, \tag{9}$$

with the inertial parameter A_j. These terms are taken into account by adding A_j to the mass matrix elements $M_{j,j}$.

III. Identification Model

Linearity in the dynamic parameters is a key feature for identifying robot parameters. By exploiting this feature the dynamic model can be written as [11]:

$$\tau = \mathbf{w}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\mathbf{X}, \tag{10}$$

where $\mathbf{X}$ consists of the n parameter vectors for every link j

$$\begin{aligned} \mathbf{X}_j &= \begin{bmatrix} \mathbf{X}_j^{(J)} & \mathbf{X}_j^{(P)} & \mathbf{X}_j^{(m)} & \mathbf{X}_j^{(f)} & \mathbf{X}_j^{(A)} \end{bmatrix}^T \\ \text{with} \quad \mathbf{X}_j^{(J)} &= \begin{bmatrix} J_{xx_j} & J_{xy_j} & J_{xz_j} & J_{yy_j} & J_{yz_j} & J_{zz_j} \end{bmatrix}, \\ \mathbf{X}_j^{(P)} &= \begin{bmatrix} P_{x_j} & P_{y_j} & P_{z_j} \end{bmatrix}, \\ \mathbf{X}_j^{(m)} &= m_j, \\ \mathbf{X}_j^{(f)} &= \begin{bmatrix} F_{c,j} & F_{v,j} \end{bmatrix}, \\ \mathbf{X}_j^{(A)} &= A_j. \end{aligned} \tag{11}$$

Applying the identification model at a sufficient number of points on some trajectories, we construct the following overdetermined linear system of equations in $\mathbf{X}$:

$$\mathbf{Y} = \mathbf{W}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\mathbf{X} + \rho, \tag{12}$$

where $\mathbf{W}$ is an $(r \times c)$ observation matrix or regressor, r is the total number of equations, c is the number of parameters such that $r \gg c$. $\mathbf{Y}$ is a vector of torque measurement data and ρ is the residual error vector. If $\mathbf{W}$ is full rank, the solution is:

$$\hat{\mathbf{X}} = (\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W} \ \mathbf{Y} = \mathbf{W}^+\mathbf{Y}, \tag{13}$$

where $\mathbf{W}^+$ is the pseudoinverse of $\mathbf{W}$. Since some columns of $\mathbf{w}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ are linearly dependent, not all parameters can be identified. Some cannot be identified at all, because they do not have any or a neglectable effect on the dynamic model. Others can only be identified in linear combination with other parameters. To achieve a full rank observation matrix, some columns and the according parameters are regrouped. This new observation matrix has to be filled with numeric values, such that the quality of the inverse $(\mathbf{W}^T\mathbf{W})^{-1}$ gives good results. This can be achieved by optimizing the data points of a trajectory with respect to an objective function, e.g. minimizing the condition number $\kappa = cond(\mathbf{W}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}))$.

The set of identifiable inertial parameters are called "base inertial parameters". These parameters are the actual physical parameters or a linear combination of those. There are numerous ways to reduce the model to this set of parameters, e.g. numerically, using the *SVD* or *QR-Decomposition* [12] or symbolically by finding a Groebner basis [13], [14] of the equations. Deduced from the terms of the energy model of serial robots there exists a closed form symbolic solution [15]. These rules give the complete set of base parameters and therefore the number of base parameters b. Note that, the parameters are regrouped in terms of the modified DH-Parameters (in Khalil/Kleinfinger notation) which suites the described geometric model. Using these relationships on (10) the dynamic model becomes

$$\tau = \mathbf{w}^{(\mathbf{B})}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \ \mathbf{X}^{(\mathbf{B})}, \tag{14}$$

where the vector of the regrouped parameters $\mathbf{X}^{(\mathbf{B})}$ is $b \times 1$ and $\mathbf{w}^{(\mathbf{B})}$ is $n \times b$ and has full rank. Finally, in a similar manner as in (12), we can now associate the latter equation with a measurement equation of the form

$$\mathbf{Y} = \mathbf{W}^{(\mathbf{B})}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\mathbf{X}^{(\mathbf{B})} + \rho, \tag{15}$$

with the matrix $\mathbf{W}^{(\mathbf{B})}$ that holds the according submatrices $\mathbf{w}^{(\mathbf{B})}$ to be evaluated in every time step.

IV. Excitation Trajectory

A. Trajectory Parameterization

The system (14) has to be evaluated at a sufficient number of data points, such that the overdetermined system (12) of this regrouped model gives a high quality least squares solution (13). The entries of $\mathbf{w}^{(\mathbf{B})}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ consist of functions with a variety of periodic terms ($\sin(\cdot)$ and $\cos(\cdot)$ terms of $\mathbf{q}$), multiplied by terms of $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$ and gravity. A good selection of data points that satisfies a sufficient excitation of all relevant terms inside $\mathbf{w}^{(\mathbf{B})}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ in one trajectory is called "persistently exciting trajectory". To find such a trajectory is difficult and is best solved as a nonlinear optimization problem

$$\min_{(\mathbf{q},\ \dot{\mathbf{q}},\ \ddot{\mathbf{q}})} \quad F(\mathbf{q},\ \dot{\mathbf{q}},\ \ddot{\mathbf{q}}) \tag{16}$$

$$\begin{aligned} \text{s.t.} \quad \mathbf{q}_{\min} \leq \mathbf{q} \ &\leq \mathbf{q}_{\max}, \\ |\dot{\mathbf{q}}| &\leq \dot{\mathbf{q}}_{\max}, \\ |\ddot{\mathbf{q}}| &\leq \ddot{\mathbf{q}}_{\max}. \end{aligned} \tag{17}$$

For the objective function $F(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ there exist many ideas [16], [11], [17], e.g. the D-optimality criterion [2], or the Hadamard inequality [18], but the most common is the condition number $cond(\mathbf{W}^{(\mathbf{B})}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}))$, which describes how much the error in $\mathbf{W}^{(\mathbf{B})}$ will cause errors in the solution $\hat{\mathbf{X}}^{(\mathbf{B})}$ by inverting $\mathbf{W}^{(\mathbf{B})}$. The optimization must respect constraints due to maximal joint range, velocities and acceleration and to prevent collisions with the robot's workspace which includes collisions with the robot itself.

The described optimization model only returns unconnected points of configurations, which have to be connected to one trajectory. One possible way is to apply interpolation (curve fitting) to the points [19], but the resulting trajectory is not guaranteed to have the same objective function value. The common way is to integrate trajectory models and their parameters into the optimization model. Trajectory models might be spline-variants [3] or variants of periodic excitation [20], [21], [22], [23]. A popular model is the Fourier series, which is given for the joint j as follows:

$$q_j(t) = \sum_{k=1}^{K} \left(\frac{a_k^{(j)}}{\omega_f^{(j)} k} \sin(\omega_f^{(j)} kt) - \frac{b_k^{(j)}}{\omega_f^{(j)} k} \cos(\omega_f^{(j)} kt) \right) + q_{j,0}\,. \tag{18}$$

The fundamental frequency $\omega_f^{(j)}$ of joint j can be chosen to be equal for all joints ($=: \omega_f$), to ensure periodicity. $a_k^{(j)}$ and $b_k^{(j)}$ are the amplitudes for the $1 \ldots K$ harmonic terms. $q_{j,0}$ is the offset and the trajectory evolves along time t. By differentiating (18), the terms for velocity and accelerations are obtained:

$$\begin{aligned} \dot{q}_j(t) &= \sum_{k=1}^{K} \left(a_k^{(j)} \cos(\omega_f kt) + b_k^{(j)} \sin(\omega_f kt) \right), \\ \ddot{q}_j(t) &= \omega_f \sum_{k=1}^{K} \left(b_k^{(j)} k\ \cos(\omega_f kt) - a_k^{(j)} k\ \sin(\omega_f kt) \right). \end{aligned} \tag{19}$$

Substituting the terms (18) and (19) in (16) for $\mathbf{q}$, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$, and writing the vectors $\mathbf{a}, \mathbf{b}$ of the Fourier coefficients and the starting joint configuration $\mathbf{q}_0$ as a vector $\mathbf{v}$ of optimization variables, i.e. $\mathbf{v} = \left[\mathbf{a}^T\ \mathbf{b}^T\ \mathbf{q}_0^T\right]^T$, the optimization model becomes

$$\begin{aligned} \min_{\mathbf{v}} \quad & F(\mathbf{v}) \\ \text{s.t.} \quad & \mathbf{q}_{\min} \leq \mathbf{D}_q \mathbf{v} \leq \mathbf{q}_{\max}, \\ & -\dot{\mathbf{q}}_{\max} \leq \mathbf{D}_v \mathbf{v} \leq \dot{\mathbf{q}}_{\max}, \\ & -\ddot{\mathbf{q}}_{\max} \leq \mathbf{D}_a \mathbf{v} \leq \ddot{\mathbf{q}}_{\max}, \\ & \mathbf{D}_f \mathbf{v} = \mathbf{0}. \end{aligned} \tag{20}$$

The terms of (18) and (19) can be written as a matrix $\mathbf{D}$. The matrices $\mathbf{D}_q, \mathbf{D}_v$ and $\mathbf{D}_a$ are the corresponding Fourier terms specific to the constraints (17). With the help of matrix $\mathbf{D}_f$ additional constraints for the first and final time (t_0 and t_f)

$$\begin{aligned} \mathbf{q}(t_0) &= \mathbf{q}(t_f), \\ \dot{\mathbf{q}}(t_0) &= \dot{\mathbf{q}}(t_f) = \mathbf{0}, \\ \ddot{\mathbf{q}}(t_0) &= \ddot{\mathbf{q}}(t_f) = \mathbf{0}, \end{aligned} \tag{21}$$

are expressed also in Fourier terms. The constraints (21) are included into the optimization model to use the trajectory multiple times and to ensure reliable behavior. The frequency ω_f is predetermined.

B. Trajectory Optimization

Due to the periodic and non convex nature of the functions in $\mathbf{w}^{(\mathbf{B})}$, any considerable objective function is expected to yield many local minima and has to be treated by a global optimization approach. To this end, we propose a memetic algorithm. Memetic algorithms extend the allegory of the genetic algorithms to cultural terms. "Memes" instead of only "genes" are transferred between individuals and generations. The difference is that features of the individuals are not only recombined or slightly mutated, but undergo substantial improvement procedures until the result is passed on to the next generation. The applied procedures use information about the problem structure and the already known solutions [7].

Our approach basically consists of two repeated steps: We initialize a new generation $\mathbf{g}_{new}$ consisting of the starting points $\mathbf{v}_0$ using different methods. The starting points are then locally optimized by a nonlinear programming solver (NLP-solver). The results are the obtained solutions $\mathbf{v}$ to the problem. The basic structure of the procedure is shown in Algorithm 1. We define a maximum number of generations $G_{\max}$. Each generation has a population P which is the number of individuals $\mathbf{v}$.

Algorithm 1 Memetic Algorithm: Top Level

while $G \leq G_{\max}$ **do**
  if $G \leq G_R$ **then**
    generate $\mathbf{g}_{new}$ with randomGenerator$(\cdot)$;
  else
    generate candidates $\mathbf{g}_{cand}$ with Algorithms 3, 4;
    select $\mathbf{g}_{new}$ out of best $\mathbf{g}_{cand}$;
  end if
  for all Individuals $\mathbf{v}_0$ in Generation $\mathbf{g}_{new}$ **do**
    $[\mathbf{v}, F(\mathbf{v})] =$ NLP-solver$(\mathbf{v}_0)$
  end for
  add $\mathbf{v}$ to solutions $\mathbf{V}$;
  add $F(\mathbf{v})$ to solution values $\mathbf{F}$;
  if new best solution obtained **then**
    decrease mutating factors;
  else
    increase mutating factors;
  end if
  G++;
end while

Algorithm 2 Random Generator

for numberOfIndividuals $= 1$ to P **do**
  generate random sequence of points $\bar{\mathbf{q}}_R$;
  solve the system $\mathbf{v}_0 = \bar{\mathbf{D}}^+ \bar{\mathbf{q}}_R$;
  add $\mathbf{v}_0$ to $\mathbf{g}_{new}$
end for
return $\mathbf{g}_{new}$;

Algorithm 3 Recombination

while *rounds* $\leq$ *maxRounds* **do**
  select randomly γ-many features of the Parents $\mathbf{g}_{best}$;
  $\mathbf{g}_{recomb}$ = permute every selected feature randomly between the individuals;
  for all $\mathbf{v}$ in $\mathbf{g}_{recomb}$ **do**
    evaluate the fitness function $f(\mathbf{v})$;
  end for
  insert every improved $\mathbf{v}$ in $\mathbf{g}_{recomb}$ into $\mathbf{g}_{cand}$;
end while
return candidates $\mathbf{g}_{cand}$;

Algorithm 4 Local Search

build δ by (27) with the points $\mathbf{v}$ and $\mathbf{v}'$;
while *loops* $\leq$ *maxLoops* **do**
  $\mathbf{v}_{cand} = \mathbf{v} - \delta \odot \mu(0, \sigma)$;
  evaluate the fitness $f(\mathbf{v}_{cand})$;
  loops++;
  if $f(\mathbf{v}_{\mathbf{cand}}) \leq f(\mathbf{v})$ **then**
    add $\mathbf{v}_{cand}$ into $\mathbf{g}_{cand}$;
    do localsearch with $\mathbf{v}_{cand}$ and $\mathbf{v}'$
  end if
end while, return $\mathbf{g}_{cand}$;

First we generate a number of G_R random generations with Algorithm 2. A sequence of points, choosing alternatively $\mathbf{q}$, $\dot{\mathbf{q}}$, or $\ddot{\mathbf{q}}$ (=: $\bar{\mathbf{q}}$), is generated within the boundaries (17) as well as in a band of some upper and lower boundaries as follows:

$$\begin{aligned} \mathbf{q}_{\min} \leq -\mathbf{q}_{ub} \leq \mathbf{q} \ &\leq -\mathbf{q}_{lb} < \mathbf{0}, \\ \mathbf{0} < \mathbf{q}_{lb} \leq \mathbf{q} \ &\leq \mathbf{q}_{ub} \leq \mathbf{q}_{\max}, \\ \mathbf{0} < \dot{\mathbf{q}}_{lb} \leq &|\dot{\mathbf{q}}| \leq \dot{\mathbf{q}}_{ub} \leq \dot{\mathbf{q}}_{\max}, \\ \mathbf{0} < \ddot{\mathbf{q}}_{lb} \leq &|\ddot{\mathbf{q}}| \leq \ddot{\mathbf{q}}_{ub} \leq \ddot{\mathbf{q}}_{\max}, \end{aligned} \tag{22}$$

while enforcing (21). The lower boundaries ensure a minimum of excitation. The random sequence $\bar{\mathbf{q}}_R$ is then approximated by a Fourier series, which is heavily inspired by [2]. Depending on the choice of $\bar{\mathbf{q}}$ with a matrix $\bar{\mathbf{D}}$ of the corresponding Fourier terms the resulting matrix-vector equation

$$\bar{\mathbf{D}}\mathbf{v} = \bar{\mathbf{q}}_R, \tag{23}$$

where

$$\bar{\mathbf{q}}_R = \begin{bmatrix} \bar{q}_1(t_0) & \bar{q}_1(t_0+1) & \dots & \bar{q}_1(t_f) \\ \bar{q}_2(t_0) & \bar{q}_2(t_0+1) & \dots & \bar{q}_2(t_f) \\ \vdots & \vdots & \ddots & \vdots \\ \bar{q}_n(t_0) & \bar{q}_n(t_0+1) & \dots & \bar{q}_n(t_f) \end{bmatrix}, \tag{24}$$

and $\bar{\mathbf{D}}$ consists of the corresponding Fourier terms. Solving (23)

$$\mathbf{v} = \bar{\mathbf{D}}^+ \bar{\mathbf{q}}_R \tag{25}$$

provides the individual $\mathbf{v}$. The vector $\mathbf{v}$ is not necessarily a feasible starting point, but often only a few solver steps away from feasibility as well as to local minima for the chosen objective function. The individuals of $\mathbf{g}_{new}$ are then solved by the NLP-solver and the objective function value and the optimized individuals are saved.

If the number of randomly generated generations G_R is reached, the remaining generations are generated by local search (Algorithm 4) and recombination (Algorithm 3) utilizing a fitness function

$$f(\mathbf{v}) = \xi_1 F(\mathbf{v}) + \xi_2 c(\mathbf{v}), \tag{26}$$

where $F(\mathbf{v})$ is the objective function and $c(\mathbf{v})$ is the sum of violated constraints. ξ_1 and ξ_2 are scaling terms, which can be tuned for better objective function value or feasibility of the obtained starting solutions $\mathbf{v}_0$. In Algorithm 3 we use the P best individuals $\mathbf{g}_{best}$ and create candidates for a new generation $\mathbf{g}_{cand}$ out of them. By using recombination the hope is that by exchanging various features, all the successful features will be combined in one individual. The crossover factor γ describes how many features will be transferred. We randomly permute every feature over the whole generation for multiple iterations and evaluate the fitness of the individuals using (26). Everytime an improvement is achieved, we add the corresponding candidate $\mathbf{v}_0$ to the next generation.

Algorithm 4 aims at finding candidates $\mathbf{g}_{cand}$ for new starting points in the neighbourhood of an already known local minimum. Because of the periodic nature of the functions

in $\mathbf{w}^{(\mathbf{B})}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})$, we assume that local minima exist in a set of periodic distances

$$\delta = \beta|\mathbf{v}-\mathbf{v}'|. \tag{27}$$

These distances are unknown, but knowing about multiple local minima we can use the distances between two local minima by choosing $\beta = 1$ and $\mathbf{v}'$ as another solution. Another approach is to push a given solution over a distance out of its own basin of attraction right into the next one. We have no a priori information about these distances either but knowing the distance between a solution $\mathbf{v}_i$ and its starting point $\mathbf{v}' = \mathbf{v}_{0,i}$, might be a good estimate. In this case we choose $\beta = 2$. The assumptions above are only intuitions and not reliable in any way. Furthermore no assumption can be made about the direction of the steps to take. Therefore we multiply the entries of δ elementwise with a vector of Gaussian distributed numbers with zero mean and a standard deviation of σ denoted by $\delta \odot \mu(0,\sigma)$. We use local search with both versions based on (27) consecutively on a random selection of individuals in $\mathbf{V}$.

The new generation $\mathbf{g}_{new}$ is selected by the best candidates – according to (26) – produced by recombination (Algorithm 3) and both versions of local search (Algorithm 4). The new generation is optimized by the NLP-solver. Depending on whether a new best solution is found, the optimization parameters $[\gamma,\sigma]$ are adapted. If we find a new best solution, we decrease them, otherwise we increase them, providing more randomness and therefore the procedure relies less on the made assumptions.

V. Application: UR5

The Universal Robots UR5 is a six rotary joint manipulator, designed to work in a collaborative environment, which means that any error in the regarding task, control policy or model design might result in damaging other production means or human collaborators. For our further research we need a very accurate model. Despite UR5's popularity, there is only one publication [24] about dynamic parameter estimation of the UR5, in which a different kinematic model is used in contrast to this paper. We show a modified DH-Parameter kinematic model and a symbolic regrouping of the base parameters.

We attach the coordinate frames R_j according to the mDH-Convention at the proximal end of the link as in Figure 1. The associated mDH-Parameters are listed in Table I. Using the recursive Newton-Euler formulation (4), (5) we derive the dynamic model (3) considering the friction model (8). Exploiting its property being linear in the dynamic parameters, we reformulate the model in the form of (10). Since we do not want to make any assumptions about the parameters, the full parameter vector (11) is used which is subsequently regrouped using the relations of [15]. The regrouped parameters are shown in Table II. Using the regrouped parameters and only the associated columns of the matrix $\mathbf{w}$ we build the system (14). The obtained matrix $\mathbf{w}^{(\mathbf{B})}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})$ has the dimension 6×52 and the associated parameter vector $\mathbf{X}^{(\mathbf{B})}$ is 52×1.

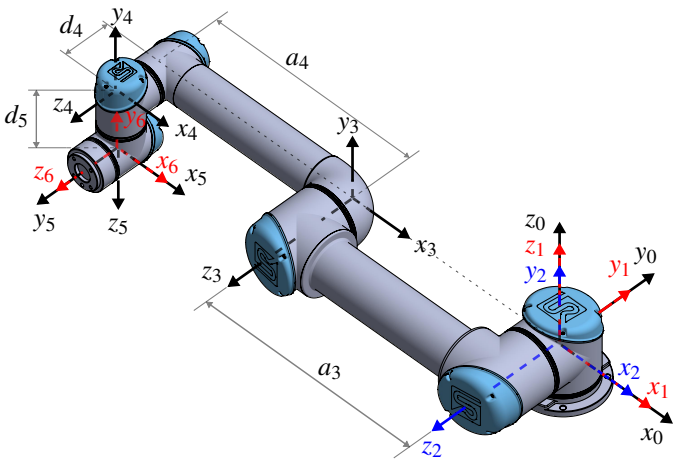

Fig. 1: UR5 with coordinate frames.

TABLE I: Modified Denavit-Hartenberg parameters for the UR5 (Khalil/Kleinfinger-Notation).

Link	α_j	a_j(m)	θ_j	d_j(m)
1	0	0	θ_1	0
2	$\frac{\pi}{2}$	0	θ_2	0
3	0	−0.425	θ_3	0
4	0	−0.392	θ_4	0.109
5	$\frac{\pi}{2}$	0	θ_5	0.094
6	$-\frac{\pi}{2}$	0	θ_6	0

The system is parameterized as described above. The constraints (17) are chosen to be

$$\begin{bmatrix} -2\pi \\ 0 \\ -0.7\pi \\ -2\pi \\ -2\pi \\ -2\pi \end{bmatrix} \leq \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{bmatrix} \leq \begin{bmatrix} 2\pi \\ \pi \\ 0.7\pi \\ 2\pi \\ 2\pi \\ 2\pi \end{bmatrix}, \tag{28}$$

$$|\dot{\mathbf{q}}| \leq 3.2\,\mathrm{rad/s}, \quad |\ddot{\mathbf{q}}| \leq 25\,\mathrm{rad/s^2}. \tag{29}$$

The constraints on q_2 and q_3 are essentially part of the non collision constraints. This way we prevent q_2 from hitting the table and link 3 from colliding the end effector with parts of the robot. Both joints are capable of rotating 2π in both directions. For further non collision constraints we chose simple rules instead of a full collision detection scheme to save computational effort. We restrict the joint such that

$$\begin{aligned} |q_2 + \frac{\pi}{2} + q_3| &\leq \frac{3\pi}{4} \\ \sin(q_4)\ |\cos(q_5)| &\leq 0.15. \end{aligned} \tag{30}$$

For the optimization 20 time samples over a period of $T = 10$ seconds are taken into account. We chose $\omega_f = \pi/T$ for all joints. The relations (22) are realized by setting the lower and upper boundaries as a percentage of their corresponding values of (28) and (29). The following results are created choosing $\bar{\mathbf{q}} := \dot{\mathbf{q}}$ and $\dot{\mathbf{q}}_{lb} = 0.2\,\mathrm{rad/s}$ and $\dot{\mathbf{q}}_{ub} = 0.9\,\mathrm{rad/s}$.

For our objective function we choose the 2-norm condition

TABLE II: Regrouped parameters and standard deviations.

$\mathbf{X}^{(\mathbf{B})}$	Symbolic terms	σ_i
1	$(m_3+m_4+m_5+m_6)a_3^2+(m_4+m_5+m_6)d_4^2+2d_4P_{z_4}+(m_4+m_5+m_6)a_4^2+J_{yy_3}+J_{yy_4}+J_{zz_1}+J_{yy_2}+A_1$	0.0094
2	$(m_4+m_5+m_6)d_4^2+(m_4+m_5+m_6)a_4^2+J_{xx_2}-J_{yy_2}-(a_4^2+d_4^2)(m_4+m_5+m_6)-(m_3+m_4+m_5+m_6)a_3^2$	0.0077
3	J_{xy_2}	0.0042
4	$J_{xz_2}+a_3(P_{z_3}+P_{z_4}+(m_4+m_5+m_6)d_4)$	0.0055
5	J_{yz_2}	0.0047
6	$J_{zz_2}+(m_3+m_4+m_5+m_6)a_3^2+A_2$	0.0070
7	$(-m_3-m_4-m_5-m_6)a_3+P_{x_2}$	0.0019
8	P_{y_2}	0.0014
9	$(m_4+m_5+m_6)d_4^2+J_{xx_3}-J_{yy_3}-(a_4^2+d_4^2)(m_4+m_5+m_6)$	0.0070
10	J_{xy_3}	0.0035
11	$((m_4+m_5+m_6)d_4+P_{z_4})a_4+J_{xz_3}$	0.0038
12	J_{yz_3}	0.0041
13	$J_{zz_3}+(m_4+m_5+m_6)a_4^2$	0.0045
14	$(-m_4-m_5-m_6)a_4+P_{x_3}$	0.0013
15	P_{y_3}	0.0012
16	$(m_5+m_6)d_5^2+2P_{z_5}d_5+J_{yy_5}+J_{xx_4}-J_{yy_4}$	0.0095
17	J_{xy_4}	0.0047
18	J_{xz_4}	0.0054
19	J_{yz_4}	0.0046
20	$J_{zz_4}+J_{yy_5}+2P_{z_5}d_5+(m_5+m_6)d_5^2$	0.0059
21	P_{x_4}	8.88e-4
22	$(-m_5-m_6)d_5+P_{y_4}-P_{z_5}$	0.0013
23	$J_{xx_5}+J_{yy_6}-J_{yy_5}$	0.0048
24	J_{xy_5}	0.0020
25	J_{xz_5}	0.0030
26	J_{yz_5}	0.0045
27	$J_{zz_5}+J_{yy_6}$	0.0057
28	P_{x_5}	0.0013
29	$P_{z_6}+P_{y_5}$	0.0010
30	$J_{xx_6}-J_{yy_6}$	0.0028
31	J_{xy_6}	0.0013
32	J_{xz_6}	0.0019
33	J_{yz_6}	0.0013
34	J_{zz_6}	0.0032
35	P_{x_6}	9.15e-4
36	P_{y_6}	9.24e-4
37	$F_{c,1}$	0.0131
38	$F_{c,2}$	0.0139
39	$F_{c,3}$	0.0139
40	$F_{c,4}$	0.0137
41	$F_{c,5}$	0.0127
42	$F_{c,6}$	0.0142
43	$F_{c,6}$	0.0124
44	$F_{v,2}$	0.0127
45	$F_{v,3}$	0.0104
46	$F_{v,4}$	0.0099
47	$F_{v,5}$	0.0113
48	$F_{v,6}$	0.0102
49	$F_{v,6}$	0.0035
50	A_4	0.0030
51	A_5	0.0059
52	A_6	0.0047

number of the matrix

$$F := \kappa = cond(\mathbf{W}^{\mathbf{B}}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})) = \frac{\sigma_{\max}}{\sigma_{\min}}, \tag{31}$$

where $\sigma_{\max}$ and $\sigma_{\min}$ are the largest and smallest singular value of $\mathbf{W}^{\mathbf{B}}$. We ran Algorithm 1 multiple times for a maximum number of generations $G_{\max} = 10$ of which are $G_R = 4$ random generations with a generation size of $P = 100$. As a local nonlinear solver we selected *Matlab's fmincon*. In most new generations a new best individual is

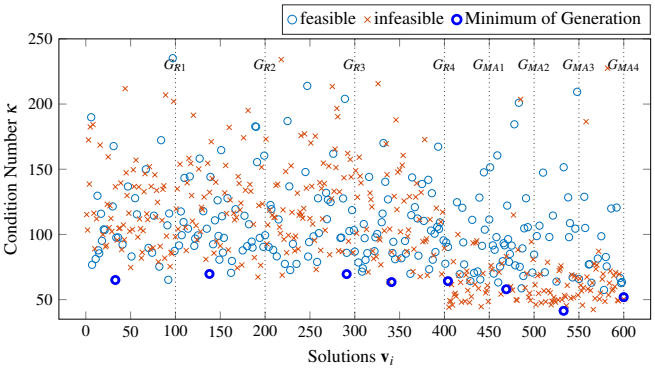

Fig. 2: Solutions for the optimized excitation trajectory obtained by Algorithm 1. The vertical lines indicate the generation which the solution $\mathbf{v}_i$ belongs to. Note that the obtained minima utilizing random initialization are all worse than the ones obtained by the MA. The best solution is obtained in the third MA-generation. The fourth MA-generation did not yield any improvements, which caused the algorithm to stop. The solutions are widely scattered, and their quality is heavily dependent on initial conditions. By finding better initial conditions for the NLP-solver, the MA is able to obtain better solutions. Numerical issues lead the NLP-solver to converge to infeasible solutions, which constitute a great proportion.

obtained. Algorithm 4 in both versions and Algorithm 3 are more or less equally contributing to the search for new, better candidates $\mathbf{v_0}$. Because of the implemented fitness function $f(\mathbf{v})$ the candidates are (often) close to feasible and grant new feasible solutions $\mathbf{v}$. In the end we achieved a condition number of $\kappa = 41$. For comparison: The best solution by far only using Algorithm 2 was $\kappa = 67$. The optimization progress of our method over generations of initial conditions is shown in Figure 2. The optimized trajectories were carried out with help of "Universal-Robots-ROS-Driver". The driver uses a position-controller, which also takes velocities and accelerations into account. Supplied with trajectory points $(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})$ in a rate of around 0.160 seconds, the robot follows the planned trajectory sufficiently, also resulting in only minor deviations from condition numbers of the planned trajectory. The identification trajectory in joint space is given in Figure 4. We drive the trajectory for 20 periods. Measurements are taken by the robots internal control unit in a rate of 125Hz, resulting in $s = 25000$ samples. Relevant measurements for the parameter estimation which can be obtained directly are actual joint positions, velocities and electric motor currents. Due to measurement noise, we filter the velocities and currents using a robust local regression (*Matlab's rloess*) to obtain smooth signals. Using the smoothed velocities, the accelerations are calculated with central difference derivation. With the smoothed actual currents $i_j^{(a)}$, it is possible to estimate the actual joint torques $\tau_j^{(a)}$. The control unit also gives access to data of the internal controller, such as targeted motor currents $i_j^{(t)}$ and targeted joint torques $\tau_j^{(t)}$. We assume a linear relationship between currents and torques. For static configurations actual and target currents are equal and it can be assumed that targeted and actual torques are also equal for such cases

$$i_j^{(a)}\frac{\tau_j^{(t)}}{i_j^{(t)}} = i_j^{(a)}\zeta_j = \tau_j^{(a)}. \tag{32}$$

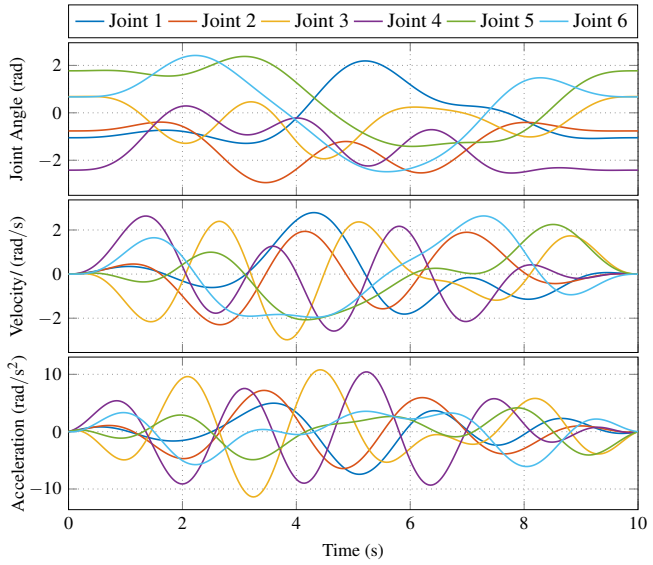

Fig. 3: Joint position, velocity and acceleration of the optimized persistently exciting trajectory over one period.

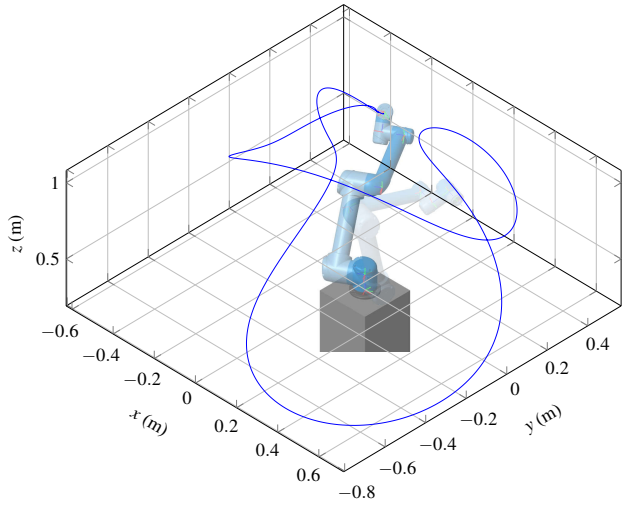

Fig. 4: End effector position over one period of the optimized persistently exciting trajectory.

We chose poses, in which the static moment acting on the observed joint is presumably close to its maximum. Gathering data for multiple configurations the current-torque constants ζ_j for every joint can be obtained using averages.

We substitute the measured $\mathbf{q}$, $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$ into $\mathbf{W}^{(\mathbf{B})}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})$ and obtain $\mathbf{W}^{(\mathbf{B})}$, which is of dimension 150000×52. To provide the opportunity to compare our results with the existing ones found by [24], we use the same statistical quality criteria. The standard deviation for the i-parameter is

$$\sigma_i = \sqrt{(\mathbf{W}^{(\mathbf{B})^T}\mathbf{W}^{(\mathbf{B})})^{-1}_{i,i}}, \tag{33}$$

and the normalized error of the estimated torques over the whole regression is

$$\rho_N = \frac{1}{s}\sqrt{\rho^T\rho}. \tag{34}$$

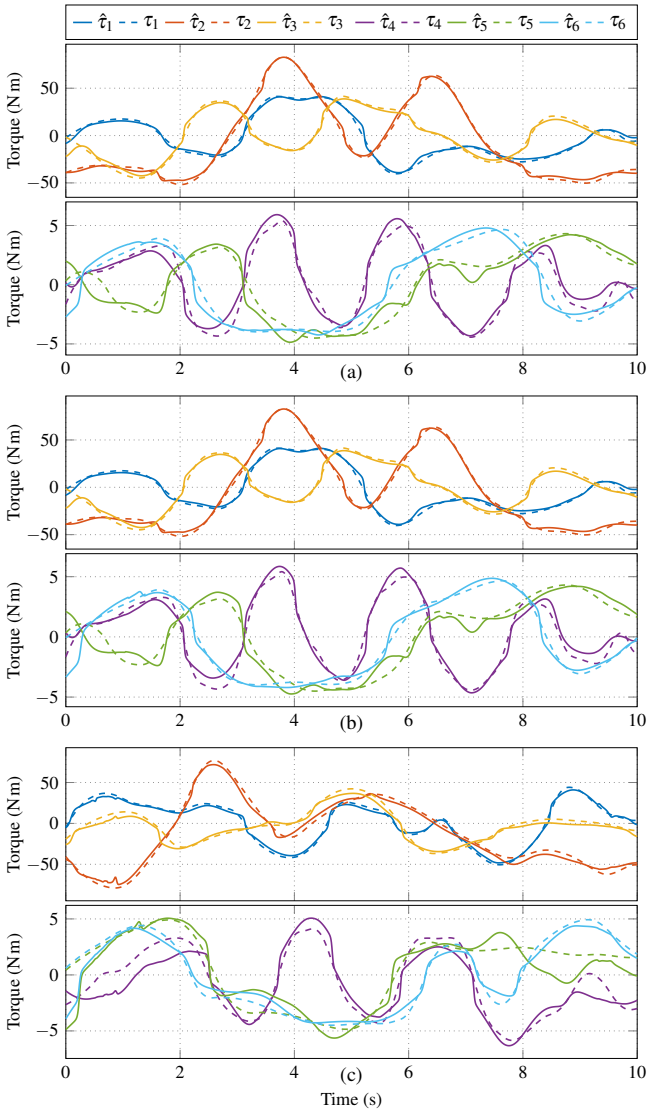

Fig. 5: Predicted torques $\hat{\tau}_j$ compared to measured torques τ_j. **(a)** CLS regression method on the optimized trajectory, **(b)** uses the OLS regression method on the same trajectory. **(c)** Torque prediction using estimated parameters obtained by OLS regression on a validation trajectory. Note that **(b)** OLS and **(a)** CLS look very similar.

We also calculate normalized errors for each single joint

$$\rho_{Nj} = \frac{1}{s}\sqrt{\rho_j^T\rho_j}, \tag{35}$$

where ρ_j is the vector of residuals for the corresponding joint. We estimate the parameters using ordinary least square regression (OLS), which provides a good fit. Because of the chosen kinematic model some of the parameters $\mathbf{X}^{(\mathbf{B})}_{1..36}$ were expected to be negative, but for the rotor inertia negative values result in a non symmetric or not positive definite mass matrix, which is physically impossible. The same result was found in [24]. We also try a constraint least squares (CLS) approach, forcing the rotor inertia to be positive. Since the values for the rotor inertia are small anyway, the constraint fit deteriorates negligibly in quality. We validate the estimated parameters obtained from both methods by testing how the calculated models predict the torques for another excitation trajectory. We also calculate the errors

TABLE III: Normalized errors.

ρ_{Nj}	Est. OLS	Est. CLS	Val. OLS	Val. CLS
1	0.0122	0.0123	0.0102	0.0094
2	0.0143	0.0146	0.0172	0.0199
3	0.0134	0.0132	0.0151	0.0170
4	0.0034	0.0043	0.0050	0.0058
5	0.0037	0.0038	0.0068	0.0061
6	0.0024	0.0033	0.0025	0.0036
ρ_N	0.0040	0.0040	0.0044	0.0049

of the validation. An overview of the estimation ("Est.") and validation ("Val.") errors is shown in Table III. We see that for both regression methods, the normalized errors are small when using the identification trajectory as well as the validation trajectory. Referring once more to Table II, we see that the standard deviations σ_i for all parameters are sufficiently small. Figure 5 shows how close the predictions $\hat{\tau}_j$ are to the actual measurements of the joint torques τ_j. Note that the predictions for the OLS and CLS parameters are nearly indistinguishable. Because of that and since we are mostly interested in validating the sufficiency of the achieved condition number, we only show the OLS parameters for validation in Figure 5(c).

VI. Conclusion

We built a dynamic model for the UR5 and identified its dynamic parameters. For a good estimation, a persistently exciting trajectory has been found by condition number optimization. Optimization was done using a memetic algorithm. We incorporated problem-specific knowledge to generate suitable starting positions, resulting in a sufficient condition number. Despite the good optimization result, it can not be ensured whether this is a global minimum. In our trials we used a very large population size and many generations to study the programmatic behavior. But this is not required in order to achieve good results in regular use cases.

The exciting trajectory was well suited to excite the dynamics, granting small standard deviations for the parameters and the subsequent regressions yield small normalized errors for the predicted joint torques. While some negative parameter values were expected due to the choice of the kinematic model, we found negative rotor inertia parameters using OLS. Even though this was mitigated by using constraint least square regression, we provided no proof that the estimated parameters lead to a positive definite mass matrix for all possible generalized coordinates.

In further steps, this can be resolved by using a nonlinear optimization model for the estimation as seen in [23], where a positive definite mass matrix is integrated into the optimization constraints.

The condition number, which was chosen as a objective function for the trajectory optimization, is dependent on the selection of base parameters and is not invariant to reparameterization. To show the validity of the proposed methods in broader generality, invariant objective functions like D-optimality could be used.

References

[1] M. Neubauer, H. Gattringer, and H. Bremer, "A persistent method for parameter identification of a seven-axes manipulator," *Robotica*, vol. 33, no. 5, pp. 1099–1112, 2015.

[2] K. Radkhah, D. Kulic, and E. Croft, "Dynamic parameter identification for the crs a460 robot," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3842–3847, IEEE, 2007.

[3] W. Rackl, R. Lampariello, and G. Hirzinger, "Robot excitation trajectories for dynamic parameter estimation using optimized b-splines," in *2012 IEEE international conference on robotics and automation*, pp. 2042–2047, IEEE, 2012.

[4] W. Wu, S. Zhu, X. Wang, and H. Liu, "Closed-loop dynamic parameter identification of robot manipulators using modified fourier series," *International Journal of Advanced Robotic Systems*, vol. 9, no. 1, p. 29, 2012.

[5] G. Calafiore, M. Indri, and B. Bona, "Robot dynamic calibration: Optimal excitation trajectories and experimental parameter estimation," *Journal of robotic systems*, vol. 18, no. 2, pp. 55–68, 2001.

[6] A. Zakharov and S. Halász, "Parameter identification of a robot arm using genetic algorithms," *Periodica Polytechnica Electrical Engineering*, vol. 45, no. 3-4, pp. 195–209, 2003.

[7] M. Gendreau, J.-Y. Potvin, *et al.*, *Handbook of metaheuristics*, vol. 2. Springer, 2010.

[8] W. Khalil and J. Kleinfinger, "A new geometric notation for open and closed-loop robots," in *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, vol. 3, pp. 1174–1179, April 1986.

[9] W. Khalil and J.-F. Kleinfinger, "Minimum operations and minimum parameters of the dynamic models of tree structure robots," *IEEE Journal on Robotics and Automation*, vol. 3, no. 6, pp. 517–526, 1987.

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

[11] W. Khalil and E. Dombre, *Modeling, identification and control of robots*. Butterworth-Heinemann, 2004.

[12] M. Gautier, "Numerical calculation of the base inertial parameters of robots," *Journal of robotic systems*, vol. 8, no. 4, pp. 485–506, 1991.

[13] H. Kawasaki, T. Shimizu, and K. Kanzaki, "Symbolic analysis of the base parameters for closed-chain robots based on the completion procedure," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1781–1786, IEEE, 1996.

[14] C. Kalker-Kalkman, "An implementation of buchbergers' algorithm with applications to robotics," *Mechanism and machine theory*, vol. 28, no. 4, pp. 523–537, 1993.

[15] 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, 1990.

[16] J. Wu, J. Wang, and Z. You, "An overview of dynamic parameter identification of robots," *Robotics and computer-integrated manufacturing*, vol. 26, no. 5, pp. 414–419, 2010.

[17] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer, 2016.

[18] J. Jin and N. Gans, "Parameter identification for industrial robots with a fast and robust trajectory design approach," *Robotics and Computer-Integrated Manufacturing*, vol. 31, pp. 21–29, 2015.

[19] K. R. Kozlowski, *Modelling and identification in robotics*. Springer Science & Business Media, 2012.

[20] J. Swevers, C. Ganseman, D. B. Tukel, J. De Schutter, and H. Van Brussel, "Optimal robot excitation and identification," *IEEE transactions on robotics and automation*, vol. 13, no. 5, pp. 730–740, 1997.

[21] J. Swevers, W. Verdonck, and J. De Schutter, "Dynamic model identification for industrial robots," *IEEE control systems magazine*, vol. 27, no. 5, pp. 58–71, 2007.

[22] J. Swevers, C. Ganseman, J. De Schutter, and H. Van Brussel, "Experimental robot identification using optimised periodic trajectories," *Mechanical Systems and Signal Processing*, vol. 10, no. 5, pp. 561–577, 1996.

[23] A. Bahloul, S. Tliba, and Y. Chitour, "Dynamic parameters identification of an industrial robot with and without payload," *IFAC-PapersOnLine*, vol. 51, no. 15, pp. 443–448, 2018.

[24] N. Kovincic, A. Müller, H. Gattringer, M. Weyrer, A. Schlotzhauer, and M. Brandstötter, "Dynamic parameter identification of the universal robots ur5,"