

## Research Article

# Optimizing Industrial Robots for Accurate High-Speed Applications

**Hubert Gattringer, Roland Riepl, and Matthias Neubauer**

*Institute for Robotics, Johannes Kepler University, Altenbergerstraße 69, 4040 Linz, Austria*

Correspondence should be addressed to Hubert Gattringer; [hubert.gattringer@jku.at](mailto:hubert.gattringer@jku.at)

Received 26 September 2012; Revised 14 December 2012; Accepted 13 January 2013

Academic Editor: Aydin Nassehi

Copyright © 2013 Hubert Gattringer et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Today's standard robotic systems often do not meet the industry's demands for accurate high-speed robotic applications. Any machine, be it an existing or a new one, should be pushed to its limits to provide "optimal" efficiency. However, due to the high complexity of modern applications, a one-step overall optimization is not possible. Therefore, this contribution introduces a step-by-step sequence of multiple nonlinear optimizations. Included are optimal configurations for geometric calibration, best-exciting trajectories for parameter identification, model-based control, and time/energy optimal trajectory planning for continuous path and point-to-point trajectories. Each of these optimizations contributes to the improvement of the overall system. Existing optimization techniques are adapted and extended for use with a standard industrial robot scenario and combined with a comprehensive toolkit with discussions on the interplay between the separate components. Most importantly, all procedures are evaluated in practical experiments on a standard robot with industrial control hardware and the recorded measurements are presented, a step often missing in publications in this area.

## 1. Introduction

State-of-the-art robotic systems are equipped with highly sophisticated industrial hardware, capable of short sample times and offering high computational power. This increasing arithmetic performance may be used to improve positioning accuracy and dynamic accuracy and compute time/energy-optimal trajectories. The proper fundamentals are found in the underlying mathematical models.

This contribution focuses on giving an overview of various nonlinear optimizations. With each optimization, a certain aspect of an arbitrary robotic system is improved. Applying all of them in sequence will provide better results regarding positioning accuracy and dynamic accuracy and will reduce the cycle times. The basis for this optimization lies in the kinematic and dynamical modeling of the system. Therefore, Section 2 starts with the deviation of these models enhanced by a model-based control strategy. Special emphasis is laid on the implementation on an industrial system including time delays. In Section 3, the static position accuracy of the robot is improved by considering unavoidable

tolerances in the robot kinematics. The main topic is obtaining optimal configurations for the identification process. The identification of the dynamic parameters, that are used for the model-based control, is described in Section 4. Also, the problem of optimal exciting trajectories is solved by nonlinear optimization techniques. Finally, Section 5 reviews time/energy optimal trajectory planning for continuous path trajectories and point-to-point trajectories.

Each section provides information on the used solvers for the various optimizations and also presents detailed results of the conducted experiments. Experiments are performed using an off-the-shelf robot, a Stäubli RX130L shown in Figure 3, with industrial control hardware and commercially available measurement hardware, to guarantee that all proposed methods are applicable for industry. For specifications of the articulated robot, please consult Table 1.

Previous works of the authors, which are also focused on industrial robotics, see, for example, [1], cover the basics like control and geometric path planning. Based on these works, a specific combination of algorithms is presented, each of which optimizes a specific aspect of the robotic system.

TABLE I: Technical specifications.

Stäubli RX130L industrial robot	
Degrees of freedom	6
Maximum load capacity	10 kg
Reach at wrist	1660 mm
Weight	≈200 kg

## 2. System Modeling and Control

**2.1. Kinematic Modeling.** The kinematic modeling can be divided in the direct kinematics problem and the inverse kinematics problem. The direct kinematics calculates the end-effector coordinates  $\mathbf{z}_E^T = (\mathbf{r}_E^T \boldsymbol{\varphi}_E^T)$  as a function of the joint coordinates  $\mathbf{q}$  and nominal geometric parameters  $\mathbf{p}_n$  as

$$\mathbf{z}_E = \mathbf{f}_{DK}(\mathbf{q}, \mathbf{p}_n), \quad (1)$$

see Figure 3. Vector  $\mathbf{r}_E$  indicates the end-effector position, which can be calculated by a sequential summation of the relative connecting vectors. For the orientation, a description in Cardan angles is used. A sequential multiplication of the relative rotation matrices between two successive joints delivers the rotation matrix  $\mathbf{A}_{IE}$  between end-effector frame ( $E$ ) and inertial frame ( $I$ ) from which the Cardan angles can be extracted. In this paper, also the notation tool center point (TCP) is used for the end-effector. Inverse kinematics calculates the joint coordinates  $\mathbf{q}$  as a function of end-effector coordinates  $\mathbf{z}_E$  and  $\mathbf{p}_n$ :

$$\mathbf{q} = \mathbf{f}_{IK}(\mathbf{z}_E, \mathbf{p}_n). \quad (2)$$

Solutions for the inverse kinematics problem of such standard robotic systems can be found in many textbooks, for example, [2] or [3].

**2.2. Dynamic Modeling.** The equations of motion for multi-body systems, like a robot, can be calculated by several methods. The method with minimal effort is the Projection Equation, see [4]. Linear momenta  $\mathbf{p} = m\mathbf{v}_c$  and angular momenta  $\mathbf{L} = \mathbf{J}\boldsymbol{\omega}_c$  are projected into the minimal space (minimal velocities  $\dot{\mathbf{q}}$ ) via the appropriate Jacobian matrices:

$$\sum_{i=1}^N \left[ \left[ \frac{\partial_R \mathbf{v}_c}{\partial \dot{\mathbf{q}}} \right]^T \left[ \frac{\partial_R \boldsymbol{\omega}_c}{\partial \dot{\mathbf{q}}} \right]^T \right]_i \left( {}_R \dot{\mathbf{p}} + {}_R \widetilde{\boldsymbol{\omega}}_{IR} {}_R \mathbf{p} - {}_R \mathbf{f}^e \right) = 0. \quad (3)$$

All the values like the translational velocity  $\mathbf{v}_c$  or the rotational velocity of the center of gravity  $\boldsymbol{\omega}_c$  can be inserted in arbitrary coordinate systems  $R$ . In contrast to  $\boldsymbol{\omega}_c$ ,  $\boldsymbol{\omega}_{IR}$  is the velocity of the used reference system. The matrix  $\mathbf{J}$  is the inertia tensor, while  $\widetilde{\boldsymbol{\omega}}\mathbf{p}$  characterizes the vector product  $\boldsymbol{\omega} \times \mathbf{p}$ .  $\mathbf{f}^e$  and  $\mathbf{M}^e$  are impressed forces and moments acting on the  $i$ th body. An evaluation of (3) yields the highly nonlinear equations of motion for the robot

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{Q}_M. \quad (4)$$

They are composed of the minimal coordinates  $\mathbf{q}$ , the configuration dependent, symmetric and positive definite mass matrix  $\mathbf{M}(\mathbf{q})$ , the vector  $\mathbf{g}(\mathbf{q}, \dot{\mathbf{q}})$  containing all nonlinear effects (like Coriolis, centrifugal, gravity, and friction forces), and the vector of the generalized actuating forces  $\mathbf{Q}_M$ .

Detailed dynamical modeling is an essential task because it is the basis for model-based control and optimal trajectory generation.

**2.3. Control.** The hardware setup is composed of an industrial PC, communicating via Powerlink with six ACOPOS servo drives, which power the synchronous motors of the Stäubli robot. The servo drives use cascaded controllers for precise position control of the robot's joints. However, the articulated robot is a highly nonlinear system due to the serially connected arm/joint units. Typically, using linear joint controllers on a highly nonlinear mechanical system will not lead to sufficiently accurate and dynamic end-effector motion.

**2.3.1. Model-Based Control.** This contribution suggests to effectively linearize the nonlinear mechanical system with a flatness-based feed-forward approach and use the servo drives' linear cascaded controllers to compensate model deviations and external disturbances. Thus, they are summarized for the sake of completeness. Introducing the state  $\mathbf{x}^T = (\mathbf{q}^T \dot{\mathbf{q}}^T)$ , the equations of motion, (4), in state space read

$$\frac{d}{dt} \begin{pmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{pmatrix} = \begin{pmatrix} \dot{\mathbf{q}} \\ \mathbf{M}(\mathbf{q})^{-1} (-\mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{Q}_M) \end{pmatrix} \quad (5)$$

with the input vector of motor torques  $\mathbf{Q}_M$ . With the flat output  $\mathbf{y} = \mathbf{q}$  and its derivatives with respect to time,

$$\begin{aligned} \mathbf{y} &= \mathbf{q}, \\ \dot{\mathbf{y}} &= \dot{\mathbf{q}}, \end{aligned} \quad (6)$$

$$\ddot{\mathbf{y}} = \ddot{\mathbf{q}} = \mathbf{M}(\mathbf{q})^{-1} (-\mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{Q}_M),$$

all state and input variables can be computed as a function of the flat output, see, for example, [5]. Consequently, the evolution of all system variables  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\mathbf{Q}_M$  is determined by a sufficiently smooth trajectory  $\mathbf{y}(t)$ .

Thus, combining the superimposed feed-forward branch with the servo drives' internal controllers yields the control law for the motor torques  $\mathbf{Q}_M$ :

$$\mathbf{Q}_M = \underbrace{\mathbf{M}(\mathbf{q}_d) \ddot{\mathbf{q}}_d + \mathbf{g}(\mathbf{q}_d, \dot{\mathbf{q}}_d)}_{\text{feed-forward } \mathbf{Q}_{M,ff}} + \underbrace{\mathbf{K}_p(\mathbf{q}_d - \mathbf{q}) + \mathbf{K}_d(\dot{\mathbf{q}}_d - \dot{\mathbf{q}})}_{\text{feed-back}}. \quad (7)$$

Please note that the feed-forward term in (7) guides the system states along the desired trajectory, while the feed-back control law ensures stability against disturbances and modeling errors with the positive diagonal matrices  $\mathbf{K}_p$  and  $\mathbf{K}_d$ . The stability of the overall system, applying (7), is proven in [6]. Alternatively, the feed forward control law may also be computed by using the structural properties of the robot's subsystem representation, as described in [7].

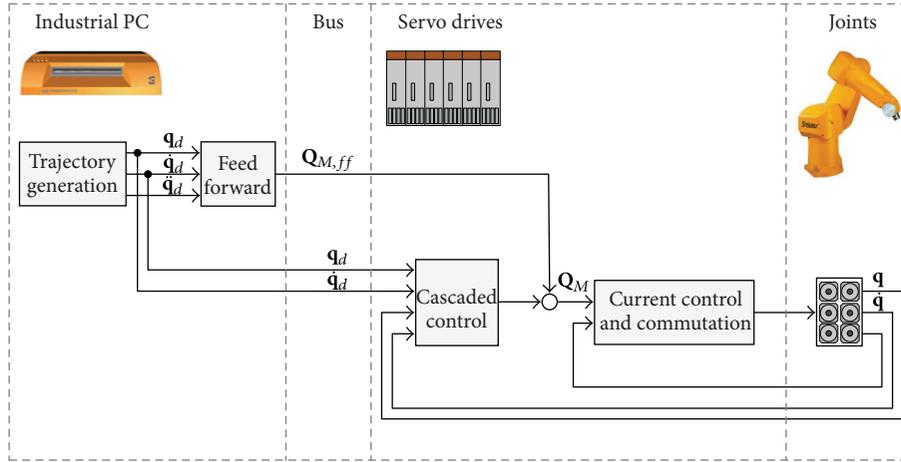


FIGURE 1: Block diagram of the control structure.

**2.3.2. Implementation.** In order to evaluate the quality of the suggested controls, the Stäubli RX130L robot is interfaced with the motion hardware. During this process, the robot's mechanics, AC motors, and resolvers are not changed or manipulated in any way.

The industrial PC is using  $400 \mu\text{s}$  as base sample time. Every sample step, the inverse kinematic problem is solved numerically to transform the desired end-effector movement into corresponding joint positions, velocities, and accelerations. With these reference joint variables ( $\mathbf{q}_d$ ,  $\dot{\mathbf{q}}_d$ , and  $\ddot{\mathbf{q}}_d$ ), the mathematical model is evaluated to compute the feed-forward motor torques as stated in (7). Then, for each joint, a separate Ethernet telegram is generated containing the reference positions and corresponding feed-forward motor torques.

The servo drives are parameterized to accept cyclic position inputs for the cascaded controller. To do so, the cyclically arriving network telegrams are received and the desired reference positions are passed to the set value inputs of the cascaded controllers. Also, the computed feed-forward torques are transformed to corresponding motor currents and added to the servo drives' current controllers set values. Figure 1 presents the corresponding block diagram.

**2.3.3. Experimental Results.** The quality of the implemented control strategy is validated by experiments with the Stäubli RX130L robot. Regarding industrial interests, the standardized trajectory found in the EN ISO NORM 9283, see [8], is used as reference trajectory. It consists of various elements: circles, straight lines, and squares in the typical workspace for an industrial application. The TCP velocity is chosen with  $1 \text{ m/s}$  and its acceleration with  $5 \text{ m/s}^2$ .

Figure 2(a) depicts the the lag errors of the first three joints. The other axes show similar results. The positive influence of the feed-forward loop is clearly observable. To present insights on the TCP-accuracy during high dynamics, the TCP errors for the standard and the improved control law along the trajectory are presented in Figure 2(b). They are

measured using a high-precision laser tracker LTD600 from Leica at a sampling rate of  $1 \text{ ms}$ .

Obviously, the positive effects of the model-based feed-forward approach drastically reduce the TCP and joint errors. Mainly, they compensate static effects, for example, resulting from gravity, and dynamic effects, for example, arising from friction and high accelerations while the standard cascaded controller provides stiffness, stability, and robustness.

### 3. Static Position Accuracy

Industrial robotic systems usually offer a satisfying repeatability. However, the lack of sufficient accuracy of the end-effector is responsible for time-consuming adjustments while commissioning new tasks. The sources of these errors are nonmodeled manufacturing tolerances, assembly inaccuracies, and joint encoder offsets.

**3.1. Kinematics with Error Parameters.** The direct kinematics, which compute the end-effector position  $\mathbf{r}_E$  and orientation  $\boldsymbol{\varphi}_E$  as function of the joint angles  $\mathbf{q}$  and nominal geometric parameters  $\mathbf{p}_n$ , can be extended by a set of error parameters  $\mathbf{p}_e$  regarding lengths, joint offsets, and assembly inaccuracies. For example, the connection vector between points 2 and 3 reads

$$\mathbf{r}_{23} = \begin{pmatrix} l_2 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} p_{e2x} \\ p_{e2y} \\ p_{e2z} \end{pmatrix}, \quad (8)$$

where  $l_2$  is the nominal length of the connection (part of  $\mathbf{p}_n$ ) and  $p_{e2x}$ ,  $p_{e2y}$ ,  $p_{e2z}$  are the position error values (part of  $\mathbf{p}_e$ ). Thus, the robot's TCP position and orientation is

$$\mathbf{z}_{Ee} = \begin{pmatrix} \mathbf{r}_{Ee} \\ \boldsymbol{\varphi}_{Ee} \end{pmatrix} = \mathbf{f}_{DKe}(\mathbf{q}, \mathbf{p}_n, \mathbf{p}_e). \quad (9)$$

For a standard six-axis industrial robot, as sketched in Figure 3,  $\mathbf{p}_e$  contains  $6 \cdot 3$  error parameters regarding length, 6 joint encoder offsets,  $6 \cdot 2$  angles for misaligned coordinate

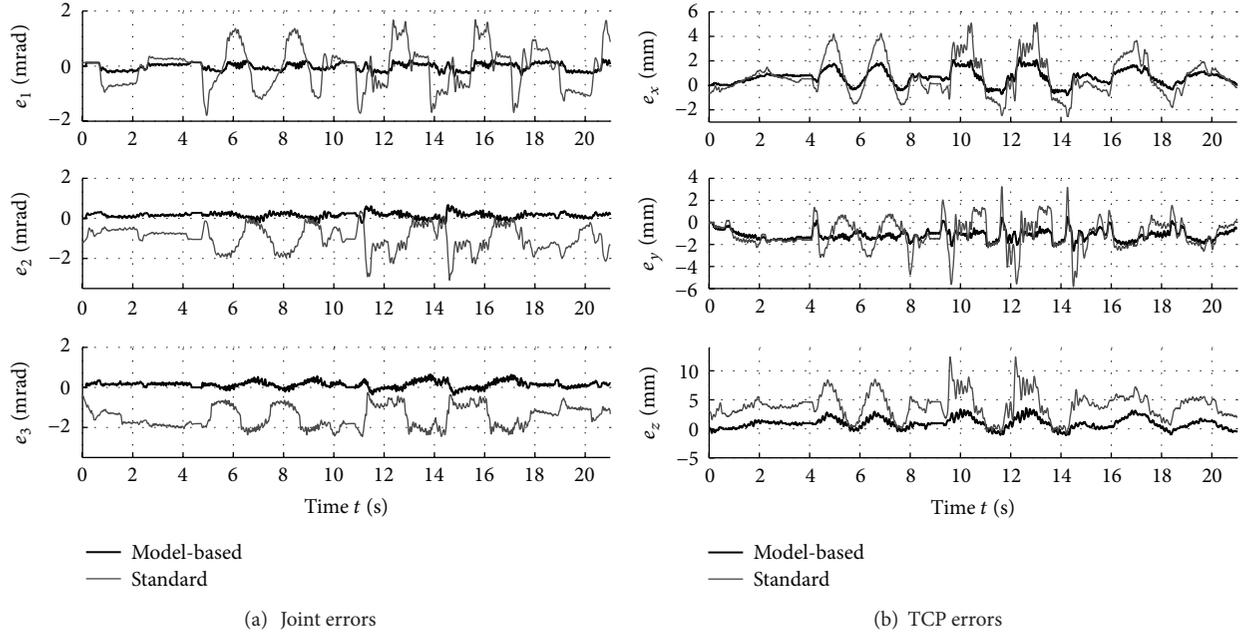


FIGURE 2: Measured errors with (model-based) and without feed-forward loop (standard).

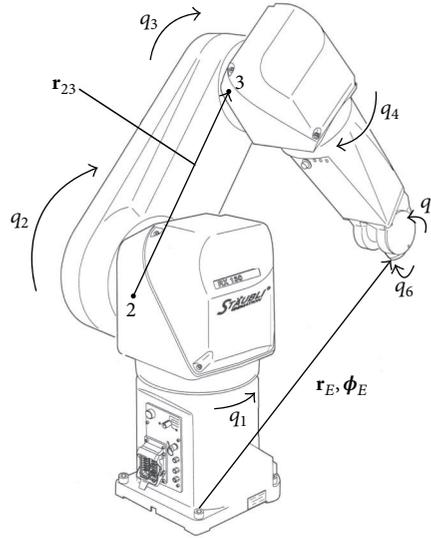


FIGURE 3: RX130L industrial robot.

frames between each link, and 3 tolerances for defining the inertial coordinate system. This results in a set of 39 parameters. To remove all linear dependencies, a QR-decomposition is applied, see Section 4.1 for details. Applying this method yields the final error parameter  $\mathbf{p}_e \in \mathbb{R}^{30}$ .

**3.2. Geometric Calibration.** The calibration process calculates valid values for the error parameter  $\mathbf{p}_e$  that minimize the error

$\mathbf{e}$  between measured TCP-positions  $\mathbf{z}_M$  and TCP-positions determined using the corresponding joint angles  $\mathbf{q}$ :

$$\mathbf{e} = \mathbf{z}_M - \mathbf{f}_{DKc}(\mathbf{q}, \mathbf{p}_n, \mathbf{p}_e) = 0. \quad (10)$$

This highly nonlinear equation can be solved by an iterative method. Representing (10) as a Taylor-series around  $\mathbf{p}_e$ ,

neglecting higher order terms and demanding that the error converges towards zero, yields

$$\mathbf{e} = \mathbf{e}(\mathbf{z}_M, \mathbf{q}, \mathbf{p}_n, \mathbf{p}_e^{(n)}) + \underbrace{\frac{\partial \mathbf{e}(\mathbf{z}_M, \mathbf{q}, \mathbf{p}_n, \mathbf{p}_e)}{\partial \mathbf{p}_e}}_{:= \widehat{\Theta}} \Big|_{\mathbf{p}_e^{(n)}} \Delta \mathbf{p}_e = 0. \quad (11)$$

For  $m$  measurements, (11) becomes

$$\underbrace{\begin{pmatrix} \mathbf{e}(\mathbf{z}_{M1}, \mathbf{q}_1, \mathbf{p}_n, \mathbf{p}_e^{(n)}) \\ \vdots \\ \mathbf{e}(\mathbf{z}_{Mm}, \mathbf{q}_m, \mathbf{p}_n, \mathbf{p}_e^{(n)}) \end{pmatrix}}_{\mathbf{Q}} + \underbrace{\begin{pmatrix} \widehat{\Theta}(\mathbf{z}_{M1}, \mathbf{q}_1, \mathbf{p}_n, \mathbf{p}_e^{(n)}) \\ \vdots \\ \widehat{\Theta}(\mathbf{z}_{Mm}, \mathbf{q}_m, \mathbf{p}_n, \mathbf{p}_e^{(n)}) \end{pmatrix}}_{\Theta} \Delta \mathbf{p}_e = 0, \quad (12)$$

which is an overdetermined equation system solvable by standard least-squares techniques. The minimum of the quadratic error is found with

$$\frac{\partial}{\partial \Delta \mathbf{p}_e} \left[ \frac{(\mathbf{Q} + \Theta \Delta \mathbf{p}_e)^T (\mathbf{Q} + \Theta \Delta \mathbf{p}_e)}{2} \right]^T = \Theta^T \Theta \Delta \mathbf{p}_e + \Theta^T \mathbf{Q} = 0, \quad (13)$$

yielding the update law for the error parameter vector

$$\Delta \mathbf{p}_e = -[\Theta^T \Theta]^{-1} \Theta^T \mathbf{Q} \text{ with } \mathbf{p}_e^{(n+1)} = \mathbf{p}_e^{(n)} + \Delta \mathbf{p}_e. \quad (14)$$

Initial values for  $\mathbf{p}_e^{(0)}$  can be assumed to be zero because the error parameters regarding length, and so forth, are considerably smaller than the nominal geometric parameters. To finally obtain good position accuracy, the identified parameters  $\mathbf{p}_e$  must be considered in the inverse kinematics algorithm. Any suitable numerical solution is viable and can handle the set of additional parameters.

**3.3. Optimal Configurations for Calibration.** It is obvious that the geometric calibration only identifies parameters well that are excited accordingly. To overcome this problem, the whole work- or joint-space could be discretized for taking the necessary measurements. While this approach is viable for small systems, it is heavily time and cost consuming for systems with multiple degrees of freedom. Such complex systems are common in today's industrial applications. As a consequence, an optimal set of configurations, which guarantees well-excited error parameters, needs to be found.

**3.3.1. Observation Index and Algorithm.** In order to ensure a well-conditioned problem formulation and also well-excited parameters  $\mathbf{p}_e$ , the minimum singular value of the covariance matrix  $\Lambda = \Theta^T \Theta$  in (14) is used as observation index and optimization criterion and further described by  $B = \sigma_m(\Lambda)$ . As, for example, [9–11] show, there exist various choices for

valid observation indices with slightly different properties. The proposed algorithm, which has been presented in [12], allows the use of arbitrary observation indices.

Before introducing the optimization algorithm, two basic operations are defined:

- (i) Add a configuration: adding a new configuration with the Jacobian  $\widehat{\Theta}_j$  to the current information matrix  $\Theta_i$  yields

$$\Theta_{i+1} = \begin{bmatrix} \Theta_i \\ \widehat{\Theta}_j \end{bmatrix}, \quad \text{and consequently } \Lambda_{i+1} = \Lambda_i + \widehat{\Theta}_j^T \widehat{\Theta}_j, \quad (15)$$

which directly influences the change of the chosen observation index

$$\Delta B = \sigma_m(\Lambda_{i+1}) - \sigma_m(\Lambda_i). \quad (16)$$

- (ii) Remove a configuration: similar to the previous case, removing a configuration  $\widehat{\Theta}_j$  from the current information matrix  $\Theta_i$  is described by

$$\Theta_i = \begin{bmatrix} \Theta_{i+1} \\ \widehat{\Theta}_j \end{bmatrix} \text{ with } \Lambda_{i+1} = \Lambda_i - \widehat{\Theta}_j^T \widehat{\Theta}_j, \quad (17)$$

$$\Delta B = \sigma_m(\Lambda_{i+1}) - \sigma_m(\Lambda_i).$$

Additionally, the complete set of  $N$  measurable configurations is called the configuration-set  $\Pi$ . An arbitrary subset of it is denoted by the set  $\xi$ . These operations and definitions are the basics for the following optimization method.

- (1) Initialization: a given number of  $n$  configurations is randomly selected from the initial configuration-set  $\Pi$ . This subset  $\xi_0$  must lead to a nonsingular covariance-matrix  $\Lambda_0$  to ensure a computation of the observation index  $B(\Lambda_0)$ . Should this not be the case for the chosen configurations, then the random selection process is simply repeated.
- (2) Remove and replace: the configuration that leads to the minimal decrease of the observation index,  $\Delta B(\Lambda_{i+1})$ , is removed from the current configuration set  $\xi_i$ . Then, the resulting subset  $\xi_{i+1}$  is complemented by exactly that configuration of  $\Pi$  that maximizes  $\Delta B(\Lambda_{i+2})$ . For both operations, the correct element is found by an exhaustive search over the respective set. Note that due to the simple way to calculate  $\Delta B$  (see ((15)–(17))), the computational effort for these searches is not as dramatic as it may sound. This remove-and-replace strategy effectively sorts out configurations that do not improve the overall conditioning of the covariance matrix and replaces them with ones that do. The step is repeated as long as one remove-and-replace operation still improves the observation index. Otherwise, the next step is executed.
- (3) Add additional configurations: from the configuration-set  $\Pi$  exactly the one configuration (again,

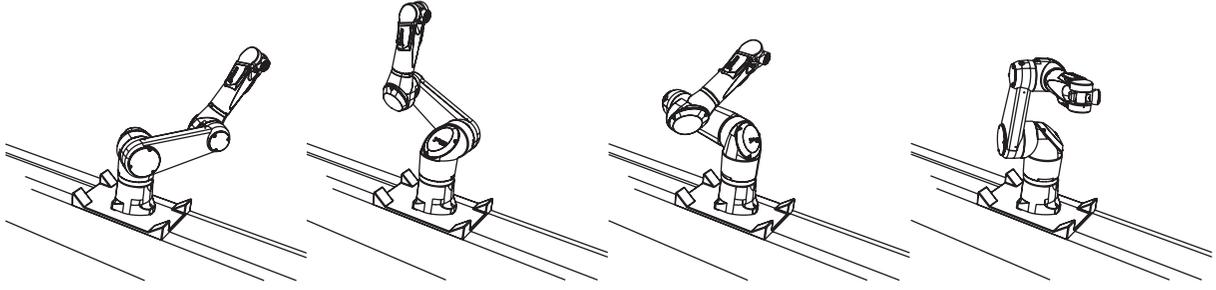


FIGURE 4: Some exemplary optimal configurations.

found by exhaustive search) that, by itself, maximizes  $\Delta B(\Lambda_{i+i})$ , is added to the current subset  $\xi_i$ . This step is iterated until the observation index cannot be further increased by such an operation or until a user-specified maximum number of desired configurations is reached. Otherwise, execute the final step.

- (4) Final remove and replace: analogously to step 2, each single configuration of the subset  $\xi_i$  is evaluated. If any remove and replace operation as described above still increases the observation index, this exchange is executed. Otherwise, the procedure is terminated.

The algorithm may be applied multiple times to prevent getting stuck at a local minimum. The solution with the maximum observation index is used for the experimental analysis. Nevertheless, finding these optimal configurations is time consuming and lasts about five minutes on a standard PC. Since these calculations are performed offline before any measurements are done, this is not a critical issue.

**3.4. Experimental Results.** To evaluate the performance of the proposed algorithm, detailed experimental results for the RX130L robot are presented. First, the robot's joint-space is discretized by 13 equally large intervals per degree of freedom. This results in a set of  $13^6 \approx 4.8 \cdot 10^6$  configurations. In reality the coordinates of the end-effector are measured by the visual infrared tracker system 3D-Creator, manufactured by the Boulder Innovation Group. A probe emitting infrared light by 4 diodes is mounted on the end-effector of the robot. This light is recorded by three CCD cameras and the position and orientation in world coordinates are calculated by stereovision. Robot configurations that are not visible by this system are removed. The remaining set consists of approximately 4000 configurations. Then, the algorithm is used to compute 32 optimal configurations. As Figure 4 emphasizes, the optimal configurations differ strongly from each other to keep the condition number of the covariance matrix low, also leading to well-excited parameters. Compared to a linear scan of the workspace, the observation index is reduced by the factor 18.5 while keeping the number of necessary measurements very small.

For information on the position and orientation accuracy before (uncal.,  $\mathbf{p}_e = 0$ ) and after calibration (cal), see Table 2. The orientation errors are represented in Cardan angles.

## 4. System Identification

In this section, the identification of the dynamical parameters describing the robot behavior, see (4), is covered. This is clearly an essential task because it is the basis for model-based control and optimal trajectory generation.

**4.1. Dynamic Parameter Identification.** An important property of the equations of motion is the linearity with respect to dynamic parameters  $\mathbf{p} \in \mathbb{R}^P$  characterizing the robot links and motors. By introducing the information matrix  $\Theta \in \mathbb{R}^{n,P}$  ( $n$  degrees of freedom), (4) can be restructured to

$$\Theta(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \mathbf{p} = \mathbf{Q}_M. \quad (18)$$

The parameter vector  $\mathbf{p}$  is the composition of the the joint parameters  $\mathbf{p}_{\text{joint},i}$

$$\mathbf{p}_{\text{joint},i}^T = (m \quad \mathbf{r}_c^T \quad \mathbf{J}^T \quad d_c \quad d_v)_i \quad (19)$$

containing the mass  $m_i$ , the center of mass  $\mathbf{r}_{c,i}$ , the elements of the inertia tensor  $\mathbf{J}_i^T = (A \ B \ C \ D \ E)_i$ , and Coulomb/viscous friction coefficients  $d_{c,i}$  and  $d_{v,i}$  for the  $i$ th joint-arm unit. Since the dynamical parameters are confidential manufacturer data, all these parameters have to be identified. Usually, the information matrix  $\Theta$  and the parameter vector  $\mathbf{p}$  contain linear dependencies, which have to be removed before the identification process. A QR decomposition and reformulation leads to

$$\widehat{\mathbf{Q}}^T \Theta = \widehat{\mathbf{R}}, \quad (20)$$

where  $\widehat{\mathbf{Q}} \in \mathbb{R}^{n,P}$  is an orthogonal matrix and  $\widehat{\mathbf{R}} \in \mathbb{R}^{P,P}$  is an upper triangular matrix. The nonidentifiable parameters are those whose corresponding elements on the diagonal of the matrix  $\widehat{\mathbf{R}}$  are zero. In contrast, if  $|\widehat{\mathbf{R}}_{ii}| > 0$ , then the corresponding column in  $\Theta$  is independent. Let the  $p - s$  independent columns be collected in the matrix  $\Theta_1$  and the corresponding parameters be collected in  $\mathbf{p}_1$ . The dependent columns and parameters are represented by  $\Theta_2$  and  $\mathbf{p}_2$ , respectively, such that

$$\Theta \mathbf{p} = [\Theta_1 \quad \Theta_2] \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{bmatrix} = [\Theta_1 \quad \Theta_1 \kappa] \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{bmatrix}, \quad (21)$$

is fulfilled, since  $\Theta_2$  is linearly dependent on  $\Theta_1$  by a matrix  $\kappa \in \mathbb{R}^{p-s,s}$ . In (21)  $\kappa$  can be shifted to the parameters leading to

$$\Theta \mathbf{p} = \Theta_1 \underbrace{(\mathbf{p}_1 + \kappa \mathbf{p}_2)}_{\mathbf{p}_b}, \quad (22)$$

with the base parameters  $\mathbf{p}_b \in \mathbb{R}^{p-s}$ . For the calculation of  $\kappa$ , (20) is again evaluated with separated matrices  $[\Theta_1 \Theta_2]$  and  $\widehat{\mathbf{Q}} = [\widehat{\mathbf{Q}}_1 \widehat{\mathbf{Q}}_2]$ , respectively, yielding

$$\widehat{\mathbf{Q}}^T [\Theta_1 \Theta_2] = \begin{bmatrix} \widehat{\mathbf{R}}_1 & \widehat{\mathbf{R}}_2 \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{R}}_{11} & \widehat{\mathbf{R}}_{12} \\ \widehat{\mathbf{R}}_{21} & \widehat{\mathbf{R}}_{22} \end{bmatrix} \quad (23)$$

and therefore

$$\widehat{\mathbf{Q}}^T [\Theta_1 \Theta_1 \kappa] = \begin{bmatrix} \widehat{\mathbf{R}}_1 & \widehat{\mathbf{R}}_1 \kappa \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{R}}_{11} \\ \widehat{\mathbf{R}}_{21} \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{R}}_{11} \\ \widehat{\mathbf{R}}_{21} \end{bmatrix} \kappa. \quad (24)$$

Combining (23) and (24),  $\kappa$  can be obtained as

$$\kappa = \widehat{\mathbf{R}}_{11}^{-1} \widehat{\mathbf{R}}_{12}. \quad (25)$$

It contains geometric parameters like lengths, lengths squared, gear ratios, and so on, see [13] for details. The identification process can be performed by sequentially arranging  $k \gg p - s$  measurements in (18) and (22) leading to

$$\mathbf{e} = \underbrace{\begin{bmatrix} \Theta_1(\mathbf{q}_1, \dot{\mathbf{q}}_1, \ddot{\mathbf{q}}_1) \\ \vdots \\ \Theta_1(\mathbf{q}_k, \dot{\mathbf{q}}_k, \ddot{\mathbf{q}}_k) \end{bmatrix}}_{\Theta_f} \mathbf{p}_b - \underbrace{\begin{pmatrix} \mathbf{Q}_{M,1} \\ \vdots \\ \mathbf{Q}_{M,k} \end{pmatrix}}_{\mathbf{Q}_f} = \mathbf{0}. \quad (26)$$

The unknown base parameters  $\mathbf{p}_b$  are calculated with the least-squares method leading to

$$\left\{ \frac{\partial}{\partial \mathbf{p}_b} \left[ \frac{\mathbf{e}^T \mathbf{e}}{2} \right] \right\}^T = \Theta_f^T \Theta_f \mathbf{p} - \Theta_f^T \mathbf{Q}_f = \mathbf{0}, \quad (27)$$

and therefore the solution

$$\mathbf{p}_b = [\Theta_f^T \Theta_f]^{-1} \Theta_f^T \mathbf{Q}_f \quad (28)$$

is directly available since  $\text{Rank}(\Theta_f) = p - s$ .

**4.2. Optimal Exciting Trajectories.** Similar to the problem of finding optimal configurations for the geometric calibration, the dynamic parameter identification requires exciting trajectories. In contrast to the steady-state configurations for the calibration, the identification trajectories include an additional dimension, the continuous time. Again, the aim is to minimize the condition number ( $\text{cond}$ ) of the covariance matrix  $\Lambda_f = \Theta_f^T \Theta_f$  by choosing suitable trajectories. A promising approach is introduced with [14]. This section summarizes these ideas, presents solutions to the optimization problem, and verifies the results on the RX130L robot.

TABLE 2: Errors before (uncal.,  $\mathbf{p}_e = \mathbf{0}$ ) and after geometric calibration (cal.).

	Position and orientation errors			
	Maximum		Peak-to-peak	
	uncal.	cal.	uncal.	cal.
$e_x$ in mm	7.779	0.272	$\pm 6.171$	$\pm 0.271$
$e_y$ in mm	19.026	0.310	$\pm 4.267$	$\pm 0.303$
$e_z$ in mm	12.614	0.414	$\pm 1.402$	$\pm 0.376$
$e_\alpha$ in $^\circ$	1.140	0.128	$\pm 0.272$	$\pm 0.109$
$e_\beta$ in $^\circ$	1.817	0.191	$\pm 0.335$	$\pm 0.182$
$e_\gamma$ in $^\circ$	1.160	0.131	$\pm 0.514$	$\pm 0.116$

**4.2.1. Trajectory Description and Optimization.** An adequate description for the identification trajectory for the  $i$ th joint is given with

$$q_i(t) = \sum_{l=1}^{N_i} \left\{ \frac{a_{i,l}}{\omega_f l} \sin(\omega_f l t) - \frac{b_{i,l}}{\omega_f l} \cos(\omega_f l t) \right\} + q_{i0}. \quad (29)$$

This Fourier-series is parameterized by the finite order  $N_i$ , the coefficients  $a_{i,l}$  and  $b_{i,l}$ , the base angular frequency  $\omega_f$ , and the joint offset  $q_{i0}$ . To guarantee a periodic excitation of the robot,  $\omega_f$  is a common parameter for all joints. To speed up the optimization time, a discretization of (29) and derivatives for one period of the base frequency ( $T_f = 2\pi/\omega_f$ ) is done. The signals are therefore evaluated at discrete time values  $i\Delta T$ ,  $i = 0 \dots k$ ,  $\Delta T = T_f/k$ , where  $k$  is the number of discretization points. This provides the following optimization problem:

$$\begin{aligned} & \underset{\mathbf{a}, \mathbf{b}}{\text{Minimize}} && \text{cond}(\Lambda_f(\mathbf{a}, \mathbf{b}, \omega_f, k)) \\ & \text{subject to} && \left| \mathbf{q}(\mathbf{a}, \mathbf{b}, \omega_f, k) \right| \leq \mathbf{q}_{\max} \\ & && \left| \dot{\mathbf{q}}(\mathbf{a}, \mathbf{b}, \omega_f, k) \right| \leq \dot{\mathbf{q}}_{\max} \\ & && \left| \ddot{\mathbf{q}}(\mathbf{a}, \mathbf{b}, \omega_f, k) \right| \leq \ddot{\mathbf{q}}_{\max} \\ & && \left| \mathbf{Q}_M(\mathbf{a}, \mathbf{b}, \omega_f, k) \right| \leq \mathbf{Q}_{M,\max} \\ & && \mathbf{r}_E(\mathbf{a}, \mathbf{b}, \omega_f, k) \subset \mathbb{R}_{\text{allowed}}^3. \end{aligned} \quad (30)$$

The constraints include the mechanical joint limits  $\mathbf{q}_{\max}$ , the permitted joint velocities  $\dot{\mathbf{q}}_{\max}$ , the expected maximum joint accelerations  $\ddot{\mathbf{q}}_{\max}$ , and a space of allowed end-effector positions  $\mathbb{R}_{\text{allowed}}^3$ , which is used to avoid collisions with the robot itself and its surroundings. Due to the flatness property of the system, see Section 2.3.1, also the motor torques can be calculated dependent on the joint coordinates and derivatives. In contrast to [15], this motor torques are also used in our formulation. The optimization problem is solved with the Matlab-Optimization-Toolbox ( $f_{\text{mincon}}$ ). The parameters order  $N = 5$ , angular frequency  $\omega_f = 0.8$  rad/s, and  $k = 80$  turned out to work well to get adequate dynamical parameters, see Figure 5(b). For numerical scaling purposes, the columns of  $\Theta_f$  should also be normalized. Solving the optimization problem yields the optimal joint trajectories.

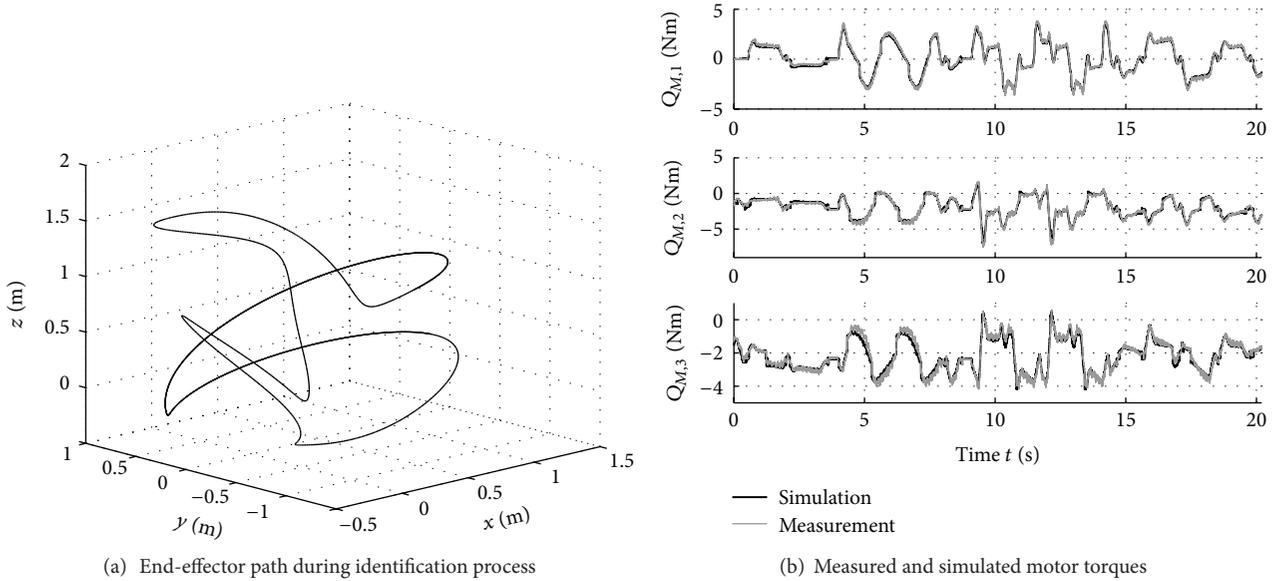


FIGURE 5: Identification trajectory and verification.

**4.3. Experimental Results.** The measurements for the overdetermined equations system (28) are taken to estimate the dynamical base parameters  $\mathbf{p}_b$ . Figure 5(a) presents a plot of the end-effector path. Along the path, velocities up to 5 m/s and accelerations up to 12 m/s<sup>2</sup> are achieved. To verify the quality of the identified parameters, the real robot's motor torques are recorded and compared to the simulated ones on a reference trajectory which is defined in the EN ISO NORM 9283, see [8]. Figure 5(b) emphasizes, by presenting the measured and simulated motor torques, the good parameter identification for the RX130L robot. Joints four, five, and six show an equivalent quality.

## 5. Optimal Trajectory Generation

There are two types of trajectory generation problems that have to be considered. The first one is the continuous path (CP) trajectory and the second one is the point-to-point (PtP) trajectory planning problem. For the first one, the geometric path in world coordinates is defined. Typical maneuvers are straight lines, circles, and splines. For the PtP planning, only the coordinates for the start-point  $\mathbf{z}_{E,0}$  and the end-point  $\mathbf{z}_{E,end}$  of the trajectory are given. These points can directly be transformed to joint coordinates  $\mathbf{q}_0, \mathbf{q}_{end}$ , and the motion planning can be performed in this coordinates. For both cases, physical constraints like motor speeds and torques have to be taken into account. Additionally, the computed trajectories have to result in continuous velocities, accelerations, and jerks. This helps that the gear elasticities are not overly excited, leading to a soft, vibration-less end-effector movement. Again, the equations of motion, derived in Section 2.2 and identified in Section 4, provide the basis for this approach.

**5.1. Continuous Path Trajectory.** CP applications appear in many robotic applications like arc welding, deburring, and laser cutting. Due to the nonlinear kinematic interrelation between Cartesian space and the robot's joint space, high motor speeds and accelerations may occur, especially near kinematic singularities. The CP motion planning can be divided into two subproblems, namely, a geometric path planner, describing a path  $\mathbf{z}_E(\sigma)$

$$\mathbf{z}_E = \begin{pmatrix} \mathbf{r}_E(\sigma) \\ \boldsymbol{\varphi}_E(\sigma) \end{pmatrix} = \mathbf{z}_E(\sigma) \in \mathbb{R}^6 \quad (31)$$

as a function of a (scalar) trajectory parameter  $\sigma \in [\sigma_0, \sigma_{end}]$  and a dynamic path planner that solves the problem of finding an optimized time behavior  $\sigma(t)$ , see also [16]. For example, a straight line in space is characterized by (39). The vector  $\mathbf{r}_E$  contains the end-effector position while  $\boldsymbol{\varphi}_E$  describes the evolution of the orientation in Cardan angles, both are functions of  $\sigma$ . The trajectory parameter  $\sigma$  itself is a function of time  $\sigma = \sigma(t)$  and the corresponding trajectory velocity and acceleration are

$$\dot{\sigma} = \frac{d\sigma}{dt}, \quad \ddot{\sigma} = \frac{d\dot{\sigma}}{dt}. \quad (32)$$

The trajectory parameter and its derivatives (trajectory speed and acceleration) represent the trajectory and its evolution in a demonstrative and easily interpretable manner. Figure 6 shows an example for a straight line, beginning at  $\sigma_0$  and ending at  $\sigma_{end}$ . With six degrees of freedom and six world coordinates, the analytical solution of the inverse kinematic problem, see for example, [3], for the RX130 robot is obtained

$$\mathbf{q} = \mathbf{f}(\mathbf{z}_E(\sigma)), \quad (33)$$

which can be differentiated with respect to time, leading to

$$\begin{aligned}\dot{\mathbf{q}} &= \frac{d}{dt} \mathbf{f}(\mathbf{z}_E(\sigma)) = \dot{\mathbf{f}}(\sigma, \dot{\sigma}) \\ \ddot{\mathbf{q}} &= \frac{d^2}{dt^2} \mathbf{f}(\mathbf{z}_E(\sigma)) = \ddot{\mathbf{f}}(\sigma, \dot{\sigma}, \ddot{\sigma}).\end{aligned}\quad (34)$$

Inserting the kinematic relations (33) and (34) into the equations of motion (4) yields the following description of the robot's dynamics:

$$\mathbf{Q}_M = \mathbf{M}(\sigma) \ddot{\mathbf{q}}(\sigma, \dot{\sigma}, \ddot{\sigma}) + \mathbf{g}(\sigma, \dot{\sigma}). \quad (35)$$

This representation allows to compute the motor torques as a function of the trajectory position, velocity, and acceleration. For additional details, see [17]. Please note that the inverse kinematics does not necessarily have to be computed analytically. Also numerical calculations of the inverse kinematics can be used.

**5.2. Optimization Strategy.** To find an optimal evolution along a desired trajectory, the objective reads

$$J = \min \int_0^{t_e} (k_1 + k_2 \mathbf{Q}_M^T \mathbf{Q}_M) dt, \quad (36)$$

where the motor torques  $\mathbf{Q}_M$  are calculated in (35). The design parameters  $k_1$  and  $k_2$  are manipulated to weight between time and/or energy optimality. However, the upper limit of the integral in (36) is unknown in advance, merely a result of optimization.

Thus, a transformation with  $\dot{\sigma} = d\sigma/dt \Rightarrow dt = (1/\dot{\sigma})d\sigma$  leads to an objective function with constant integration limits and  $\sigma$  as integration variable. Equation (38) summarizes the result. The objective function is also subject to a set of differential equations, describing the evolution of time along the trajectory parameter  $\sigma$ . The derivatives with respect to  $\sigma$  are used to compute all trajectory variables, which are found in the constraints. For instance, the trajectory acceleration is computed with

$$\ddot{\sigma} = \frac{d\dot{\sigma}}{dt} = \frac{d\dot{\sigma}}{d\sigma} \dot{\sigma} = \dot{\sigma}' \dot{\sigma}, \quad (37)$$

where a prime indicates the differential operator  $(\cdot)' = d(\cdot)/d\sigma$ . The input  $u$  in the set of differential equations needs to be computed by a numerical solver in order to minimize the objective function while satisfying all constraints.

Finalizing the following complete description:

$$\begin{aligned}\text{Minimize}_{\sigma} \quad & \int_{\sigma_0}^{\sigma_{\text{end}}} \frac{1}{\dot{\sigma}} (k_1 + k_2 \mathbf{Q}_M^T \mathbf{Q}_M) d\sigma \\ \text{subject to} \quad & \frac{d}{d\sigma} \begin{pmatrix} t \\ \dot{\sigma} \\ \dot{\sigma}' \end{pmatrix} = \begin{pmatrix} 1 \\ \dot{\sigma}' \\ u \end{pmatrix} \\ & |\mathbf{Q}_M| \leq \mathbf{Q}_{M,\text{max}} \\ & |\dot{\mathbf{q}}| \leq \dot{\mathbf{q}}_{\text{max}} \\ & |\ddot{\mathbf{q}}| \leq \ddot{\mathbf{q}}_{\text{max}}\end{aligned}\quad (38)$$

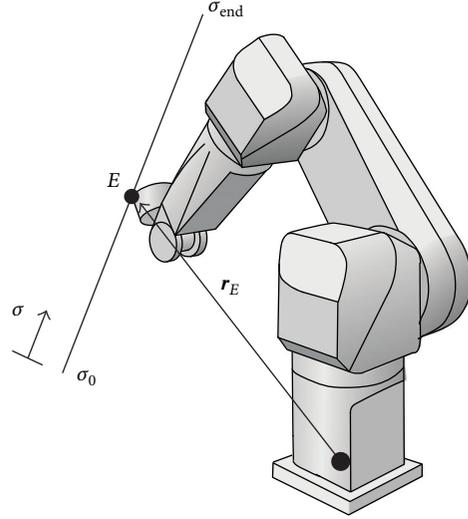


FIGURE 6: A sample trajectory.

the motor torques, motor velocities, and accelerations are constrained. To compute an optimal solution  $u$  for the optimization problem stated in (38), the MUSCOD-II software package is utilized. Based on [18], it is developed by the Interdisciplinary Center for Scientific Computing at the University of Heidelberg. Using the direct multiple-shooting method, see [19, 20], the original continuous optimal control problem is reformulated as a nonlinear programming (NLP) problem which is then solved by an iterative solution procedure, a specially tailored sequential quadratic programming (SQP) algorithm. For further details on the used software package, see [21].

**5.3. Experimental Results—CP Trajectory.** For a straight tool center movement with constant end-effector orientation, the trajectory is given by

$$\mathbf{z}_E = \begin{pmatrix} x_0 + (x_{\text{end}} - x_0) \sigma \\ y_0 + (y_{\text{end}} - y_0) \sigma \\ z_0 + (z_{\text{end}} - z_0) \sigma \\ \mathbf{0} \end{pmatrix}, \quad \sigma \in [0, 1], \quad (39)$$

where index 0 indicates the start-point and index end the end-point, respectively. For a grid of 200 shooting intervals, the time optimal case  $k_1 = 1, k_2 = 0$  is computed. Minimizing energy would lead to not only smoother, but also longer trajectories for  $\sigma$  and is therefore not considered in this example. The resulting trajectory parameters  $t(\sigma), \dot{\sigma}$ , and  $\dot{\sigma}'$  are retransformed into the time domain to compute the corresponding reference joint angles  $\mathbf{q}_d(t)$  and the constrained feed-forward torques  $\mathbf{Q}_{M,ff}(t)$ . The corresponding velocity profile is depicted in Figure 7. The velocity-drop occurs due to passing closely to a singular configuration.

The experiment with the Stäubli robot is conducted analogously to the previous section. The reference angles and torques are transferred to the servo drives where the occurring torques and motor velocities are measured for verification. Figure 8 shows the torques and velocities of the

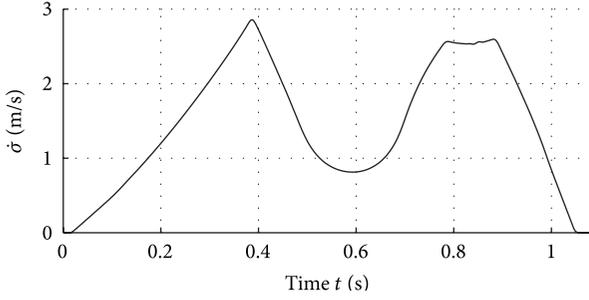


FIGURE 7: Velocity profile CP trajectory.

axes that actively constrain the trajectory. For readability reasons, the signals are normalized to their maximum values. The figure also shows that at least one constraint is always active, which is a necessary condition for the time-optimal solution.

**5.4. Point-to-Point Trajectory.** As an extension to the previous section, the PtP optimization is not limited to geometrically defined paths. Consequently, the one-dimensional variation of the trajectory parameter becomes a more complex two-point boundary value problem of the joint states with the additional task of minimizing time and/or energy, summarized in (40). Analogously to (38), the available joint velocities and torques are limited. Additionally, the joint limits  $\mathbf{q}_{\max}$  need to be considered and the desired initial and final configurations  $\mathbf{q}_0$ ,  $\dot{\mathbf{q}}_0$ ,  $\mathbf{q}_e$ , and  $\dot{\mathbf{q}}_e$  are introduced.

A compromise between time and energy optimality is adjusted with the parameters  $k_1$  and  $k_2$ . Also note that the highly nonlinear differential equations of motion are included in the set of constraints.

**5.5. Experimental Results—PtP Trajectory.** The optimization formulation for the point-to-point trajectory

$$\begin{aligned}
 & \text{Minimize}_{\mathbf{q}, t_e} \int_0^{t_e} (k_1 + k_2 \mathbf{Q}_M^T \mathbf{Q}_M) dt \\
 & \text{subject to} \quad |\mathbf{Q}_M| \leq \mathbf{Q}_{M,\max} \\
 & \quad \quad \quad |\dot{\mathbf{Q}}_M| \leq \dot{\mathbf{Q}}_{M,\max} \\
 & \quad \quad \quad |\mathbf{q}| \leq \mathbf{q}_{\max} \\
 & \quad \quad \quad |\dot{\mathbf{q}}| \leq \dot{\mathbf{q}}_{\max} \\
 & \quad \quad \quad \frac{d}{dt} \begin{pmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{pmatrix} = \begin{pmatrix} \dot{\mathbf{q}} \\ \mathbf{M}^{-1}(\mathbf{q}) (-\mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{Q}_M) \end{pmatrix} \\
 & \quad \quad \quad \mathbf{q}(0) = \mathbf{q}_0 \\
 & \quad \quad \quad \dot{\mathbf{q}}(0) = \dot{\mathbf{q}}_0 \\
 & \quad \quad \quad \mathbf{q}(t_e) = \mathbf{q}_e \\
 & \quad \quad \quad \dot{\mathbf{q}}(t_e) = \dot{\mathbf{q}}_e
 \end{aligned} \tag{40}$$

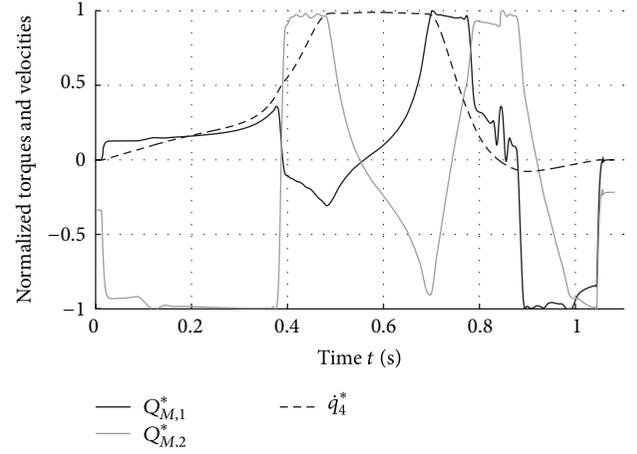


FIGURE 8: Active constraints recorded during experiment.

is solved with MUSCOD-II setting  $k_1 = 1$  and  $k_2 = 0$  (time optimality). While the one-dimensional variation problem of the previous section is computed in one second, the PtP trajectory takes about three minutes to be calculated with a standard workstation. After solving the optimization problem, the system states and torque inputs for the two-point boundary problem are known and may be utilized as reference values. The initial conditions  $\mathbf{q}_0$ ,  $\dot{\mathbf{q}}_0$ ,  $\mathbf{q}_e$ , and  $\dot{\mathbf{q}}_e$  are chosen equals to the start- and endpoint of the previous section's straight-line trajectory. This allows a comparison between both methods. The overall time of the trajectory was reduced from  $t_e = 1.04$  s to  $t_e = 0.67$  s. Figures 9(a) and 9(b) show the optimized torques and velocities. Again, the most interesting signals are selected for reasons of readability, all normalized to their maximum values. Experiments using these results on the RX130L robot verify validity of the optimized trajectory. The maximum end-effector velocity and acceleration that are reached along this trajectory are 5.2 m/s and 48.6 m/s<sup>2</sup>, respectively. To avoid high jerks in joint space, which would induce vibrations, the torque rates are limited to  $\mathbf{Q}_{M,\max}$ .

## 6. Conclusions

This work summarizes various optimization strategies for robotic systems to improve the overall performance. The individual results of each section not only verify the chosen approach but also offer values for expected accuracies of modern medium scale robotic systems. The suggested ideas are also applicable to smaller or larger scale robots with an arbitrary amount of degrees of freedom.

It is important to note that the quality of the control as well as the trajectory optimization is only as good as the underlying set of (kinematic or dynamical) parameters. As a consequence, the optimal configurations for calibration and the optimal-exciting trajectories for the identification process provide the basis for control and trajectory generations. All proposed methods can be automated by designing according to algorithms and used during the commissioning of robot applications.

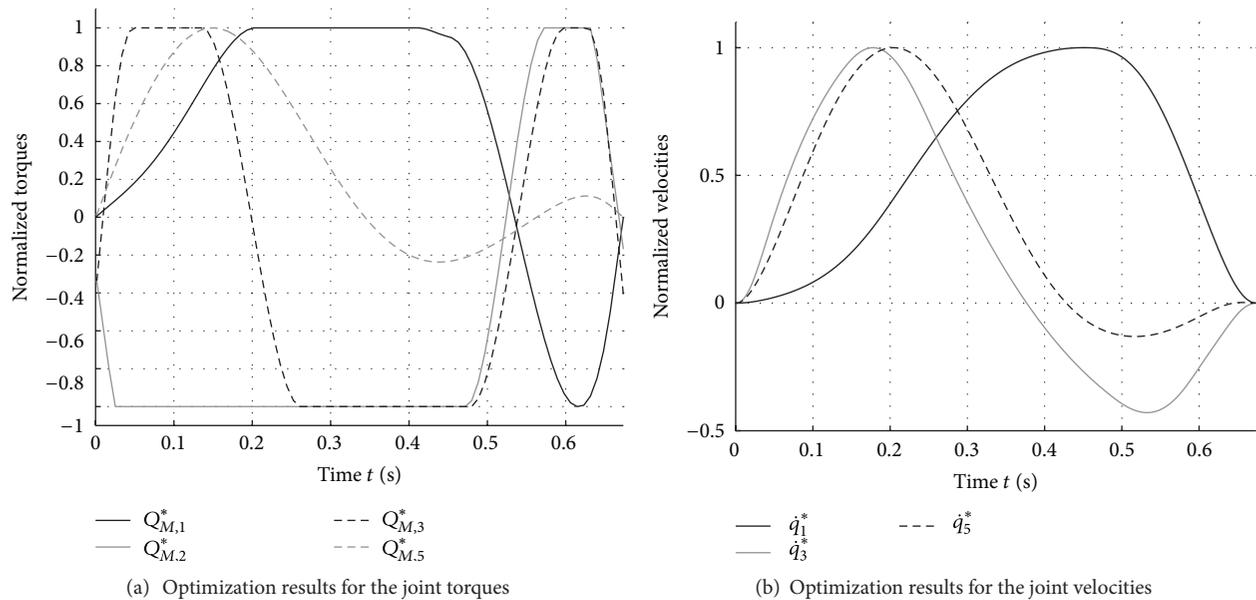


FIGURE 9: Results PtP trajectories.

## Acknowledgments

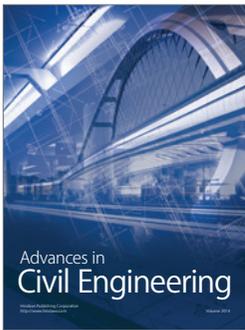
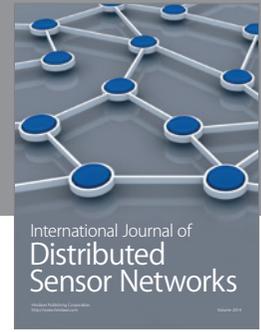
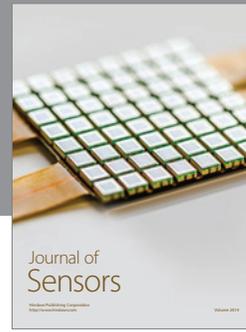
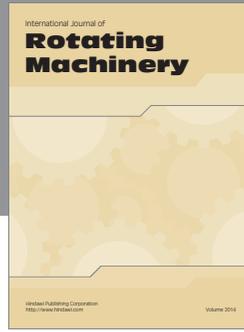
Support of the current work within the framework of the Austrian Center of Competence in Mechatronics (ACCM) is gratefully acknowledged. The support of the Interdisciplinary Center for Scientific Computing at the University of Heidelberg is acknowledged, especially to Moritz Diehl and Andreas Poschka who kindly gave support. One of the basic ideas of this contribution is to review and present details using industrial, not laboratory, equipment. However, the authors do not have a direct relation to any mentioned hardware and software products. This information is consciously included, enabling the reader to review technical specifications for further comparisons.

## References

- [1] R. Riepl and H. Gattringer, "Accurate high-speed robotics [Dissertation]," Johannes Kepler University, B&R Industrial Ethernet Award, 2009.
- [2] W. Khalil and E. Dombre, *Modeling, Identification and Control of Robots*, Kogan Page Science, London, UK, 2004.
- [3] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*, Advanced Textbooks in Control and Signal Processing Series, Springer, London, UK, 2009, Edited by A. J. J. Grimble.
- [4] H. Bremer, *Elastic Multibody Dynamics: A Direct Ritz Approach*, Springer, New York, NY, USA, 2008.
- [5] A. Isidori, *Nonlinear Control Systems*, Springer, New York, NY, USA, 1985.
- [6] A. Kugi, "Introduction to tracking control of finite- and infinite-dimensional systems," *Stability, Identification and Control in Nonlinear Structural Dynamics (SICON)*, 2008.
- [7] H. Gattringer and H. Bremer, "A penalty shooting walking machine," in *IUTAM Symposium on Vibration Control of NonLinear Mechanisms and Structures*, H. Ulbrich and W. G. Günthner, Eds., Springer, 2005.
- [8] ISO NORM 9283, "Manipulating industrial robots—performance and criteria," Norm, EN ISO, 9283, 1998.
- [9] C. H. Menq, J. H. Borm, and J. Z. Lai, "Identification and observability measure of a basis set of error parameters in robot calibration," *Journal of Mechanisms, Transmissions, and Automation in Design*, vol. 111, no. 4, pp. 513–518, 1989.
- [10] M. R. Driels and U. S. Pathre, "Significance of observation strategy on the design of robot calibration experiments," *Journal of Robotic Systems*, vol. 7, no. 2, pp. 197–223, 1990.
- [11] Y. Sun and J. M. Hollerbach, "Observability index selection for robot calibration," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '08)*, pp. 831–836, May 2008.
- [12] Y. Sun and J. M. Hollerbach, "Active robot calibration algorithm," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '08)*, pp. 1276–1281, May 2008.
- [13] H. Gattringer, *Starr-Elastische Robotersysteme: Theorie und Anwendungen*, Springer, Berlin, Germany, 2011.
- [14] 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.
- [15] D. Kostić, B. de Jager, M. Steinbuch, and R. Hensen, "Modeling and identification for high-performance robot control: an RRR-robotic arm case study," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 6, pp. 904–919, 2004.
- [16] R. Johanni, *Optimale Bahnplanung bei Industrierobotern*, Technische Universität München, München, Germany, 1988.
- [17] R. Riepl and H. Gattringer, "An approach to optimal motion planning for robotic applications," in *Proceedings of the 9th International Conference on Motion and Vibration Control*, 2008.
- [18] D. Leineweber, I. Bauer, A. Schäfer, H. Bock, and J. Schlöder, "An efficient multiple shooting based reduced SQP strategy

for large-scale dynamic process optimization (Parts I and II),” *Computers and Chemical Engineering*, vol. 27, pp. 157–174, 2003.

- [19] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Science+Business Media, LLC, New York, NY, USA, 2nd edition, 2006.
- [20] J. T. Betts, *Practical Methods for Optimal Control Using Nonlinear Programming*, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 2001.
- [21] M. Diehl, D. B. Leineweber, and A. Schäfer, *MUSCOD-II Users’ Manual*, Universität Heidelberg, Heidelberg, Germany, 2001.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

