Martin Grotjahn
Bodo Heimann

Hannover Center of Mechatronics
Appelstr. 11, D-30167
Hannover, Germany
grotjahn@mzh.uni-hannover.de
heimann@mzh.uni-hannover.de

Model-based Feedforward Control in Industrial Robotics

Abstract

Simple linear joint controllers are still used in typical industrial robotic systems. The use of these controllers leads to non-negligible dynamic path deviations for applications that require high path accuracy. These deviations result from the strong influence of nonlinearities, such as multi-body dynamics and gear friction. Sophisticated nonlinear control algorithms, known from the literature, are still not used because they usually require an expensive change of the control architecture. Therefore, different compensation methods are compared in this paper which reduce the path deviations by correction of the desired trajectory. This means that no torque interface is required, only an interface for path corrections is necessary. Such an interface normally exists so that the methods can simply be implemented within standard industrial controls. Using the industrial robot Siemens manutec-r15 *the methods are experimentally compared with respect to their efficiency and practical applicability. Starting from this, one method is chosen for application to the state-of-the-art industrial robot* KUKA KR15. *The algorithm is based on a complete nonlinear dynamic model of the robot. It is integrated into the standard control KRC1. The experimental results prove the efficiency and the industrial applicability of the method.*

KEY WORDS—Industrial robots, modelling, dynamics, feedforward control

1. Introduction

Nowadays there is a large gap between the sophisticated control algorithms developed in robotics research and the simple control techniques used in industrial robotic applications. Typical industrial controllers disregard the nonlinearities of robot dynamics, such as friction and couplings between different links. Although these linear controllers are sufficient for applications that require only high positioning accuracy, they lead to non-acceptable deviations for applications requiring high path accuracy, e.g., laser-cutting.

The International Journal of Robotics Research
Vol. 21, No. 1, January 2002, pp. 45-60,

The nonlinear effects can be compensated for by powerful control algorithms, like computed-torque and adaptive methods (Sciavicco and Siciliano 1996), which have been developed in robotics research some years ago. The problem with these control algorithms is the necessity of a torque interface. Such an interface is usually not present in standard robot controllers. However, common industrial robots have an interface which is designed for handing over path corrections from external sensor signals. An algorithm is proposed by Lange and Hirzinger (1994, 1996, 1999a,b) which uses this interface for the reduction of path deviations. Corrections of the desired trajectory are calculated with respect to a linear model of each link's closed loop. Due to robot dynamics the resulting trajectory is closer to the original desired path. In spite of using linear models, nonlinear influences can be compensated by taking into account measured path deviations.

The authors of this paper proposed another approach, called nonlinear precorrection (Grotjahn et al. 1999). It combines the main ideas and advantages of the computed-torque method and the mentioned precorrection algorithm. As in the computed-torque method, the inverse dynamics of the robot are used to calculate the expected torques from the desired joint motions. Since no torque interface is available for feedforward control, the resulting values are transferred into trajectory corrections by an inverted controller model.

In this paper, the different approaches for model-based feedforward control are discussed. The methods are experimentally compared using the industrial robot *Siemens manutec-r15* in order to evaluate the improvement of path accuracy and practical applicability. These experimental investigations reveal that the nonlinear precorrection method combines excellent results with robustness. Therefore, it is integrated into the standard KRC1 control of the present-day robot *KUKA KR15* and experimentally tested.

The paper consists of four main sections. In the next section the causes of path deviations in robot dynamics are discussed. Section 3 introduces two different modeling approaches and identifies the associated parameters. The different control strategies are developed and applied to the *Siemens*

manutec-r15 in Section 4. In Section 5 the application to the commercial robot *KUKA KR15* is presented. Finally, a short summary is given.

2. Path Deviations in Robot Dynamics

Path deviations in industrial robotics arise from different reasons. These reasons can be grouped into path-planning errors, kinematic errors and dynamic errors. Path-planning and kinematic errors can be taken into account by high performance offline-programming and simulation tools, like the so-called "Realistic Robot Simulation Interface" (RRS) (Bernhardt et al. 1995). Dynamic errors, however, are still a major problem for applications that require high path accuracy.

The main deviating influences in robot dynamics result from gear elasticity, backlash and tracking errors. Elasticity and backlash are usually small, as in industrial robotics very stiff gears are used. Furthermore, compensation for these influences is difficult because normally only motor encoder measurements and no drive side measurements can be obtained.

Therefore this paper concentrates on the reduction of tracking errors. These result from the fact that in industrial robotics linear independent joint controllers are still used (see Section 5.2.1). These controllers cannot compensate for the nonlinear effects which determine industrial robot dynamics.

Besides the multi-body dynamics, friction is the main deviating nonlinear influence in industrial robotics. Figure 1 depicts the tracking errors of the second joint during a vertical circle of the end-effector of the robots investigated by the authors. The joint performs a motion with sinus-like velocity. Tracking errors dramatically increase at the zero-crossing of velocity. This reveals the strong impact of velocity dependent friction. Due to a change of sign of the joint velocity a large jump of friction torque occurs (see Section 3.1.1). This leads to significant tracking errors which are only slowly reduced by the integral part of the linear controller.

3. Modeling of Robot Dynamics

For the modeling of robots' dynamic behavior two different approaches are investigated. In the first one a nonlinear model is used, which contains the main deviating influences. The second approach comprises independent linear models of each joint's closed control loop. Although this approach is not capable of reflecting the nonlinear influences it can be used for their compensation.

3.1. Nonlinear Modeling

The nonlinear model is divided into a gear model and a rigid-body model. The gear model comprises the friction losses in motors, gears and bearings.

3.1.1. Modeling and Identification of Gear Friction

Normally, very stiff gears with small backlash are used in industrial robotics. Therefore, gear elasticity and backlash do not have to be considered for identification and feedforward control of industrial robots' dynamics. In addition to rigid body dynamics, only the losses in gears and bearings are taken into account by gear friction models. For their identification friction has to be separated from other dynamic effects. By using trajectories where only one axis is moved and selecting parts of the measurement with constant velocity, the effects of acceleration, centrifugal and Coriolis forces are avoided. If gravitation has an influence on the torque of the regarded axis, it has to be compensated by a model identified in advance. Thus, the following measurements can be assumed to solely reflect the influence of friction in gears and bearings. The identification of the gravitation model can be done without an explicit friction model by the method proposed in Section 3.1.2.

Friction losses of a single robotic joint are usually modeled as a torque $\tau_{i,f1}$ which is a function of its own rotational joint speed $\dot{q}_i$. This nonlinear function is mostly described by the sum of terms for viscous damping and dry friction (Armstrong-Hélouvry 1991, de Wit et al. 1991, Swevers et al. 2000, Pfeiffer and Hölzl 1995, Seeger 1992)

$$\tau_{i,f1} = a_{i,1}\dot{q}_i + a_{i,2}\text{sign}(\dot{q}_i)\,. \tag{1}$$

This simple model needs to be refined for more precise modeling (Daemi and Heimann 1996, Prüfer and Wahl 1994). Figure 2(a) shows measured friction torques of some axes of the *Siemens manutec-r15* over their full speed range, normalized to their maximum speed and maximum torque. It can be seen that all axes show significant degressive characteristics, not covered by the simple model given in eq. (1). A better description of the measured friction characteristics can be found by using one of the following equations:

$$\tau_{i,f2} = a_{i,1}\dot{q}_i + a_{i,2}\text{sign}(\dot{q}_i) + a_{i,3}\dot{q}_i^{\frac{1}{3}}, \tag{2}$$

$$\tau_{i,f3} = a_{i,1}\dot{q}_i + a_{i,2}\text{sign}(\dot{q}_i) + a_{i,3}\arctan(a_{i,4}\dot{q}_i). \tag{3}$$

For a given measured friction characteristic, an analytical description with a least square model error can easily be calculated for eqs. (1) and (2), since they depend only linearly on their parameters. For model (3) a nonlinear optimization procedure is applied. As depicted in Figure 2(b), both models lead to much better results than the classic one (eq. (1)). The mean quadratic error between the measured and modeled torques is used to decide whether model (2) or (3) gives a better description of the measured characteristics. It has to be kept in mind that modeled parameters $a_{i,j}$ found in this way no longer represent physical models (such as dry friction or viscous damping) but merely force the sum of all effects of the multistage gears into a mathematical description.

The major problem in friction modeling is the time variance of friction (Daemi and Heimann 1996, 1998, Prüfer and

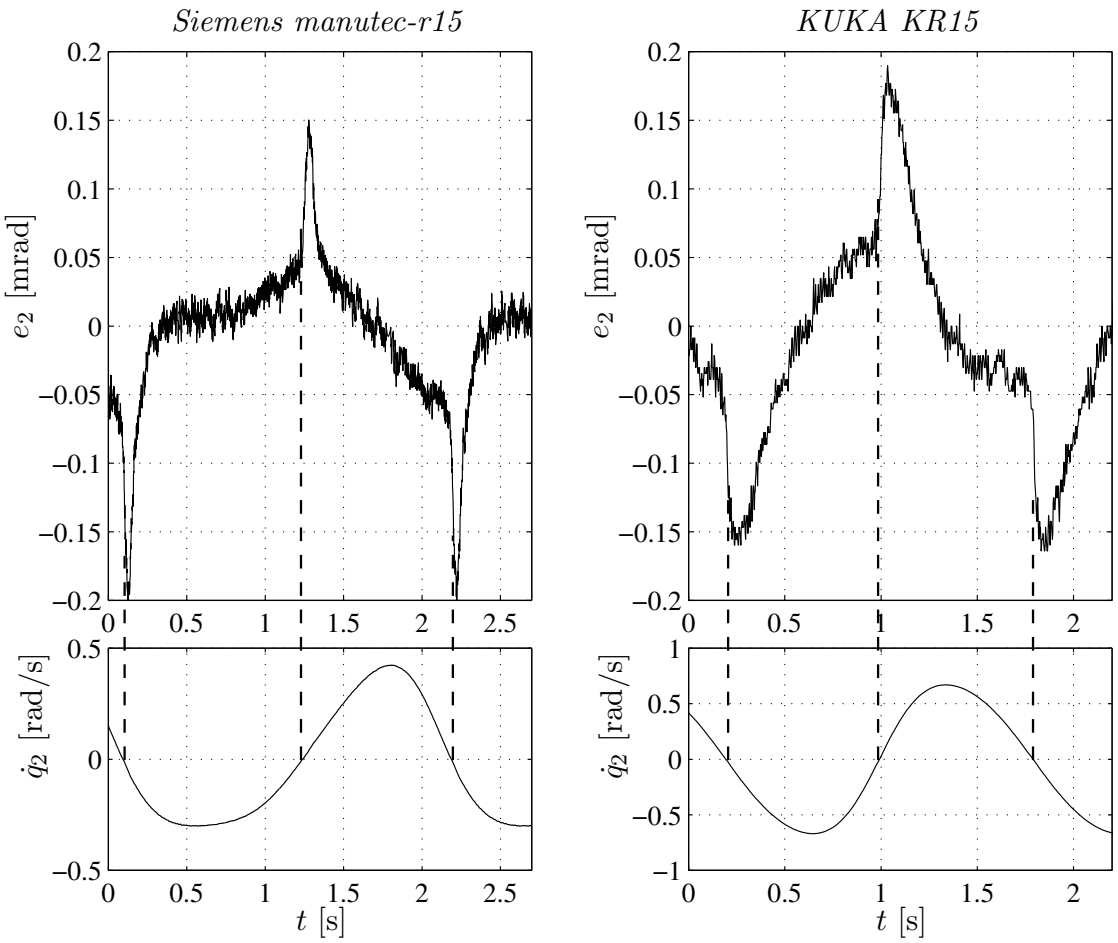

Fig. 1. Influence of friction on tracking errors of joint 2 for a vertical circle.

Wahl 1994). Commonly, relatively large time constants are assumed that arise from temperature variation in gears and bearings. Thus, the typical approach for friction modeling in robotics is to use some "warm-up" time where the robot is supposed to reach stationary conditions (Armstrong-Hélouvry 1991, Seeger 1992, Daemi and Heimann 1996). The measurements in Figure 3 reveal that this does not generally hold for geared robots.

Figure 3 shows friction with respect to time for a multistage robot gear which is continuously moved back and forth. The torque is measured every 12 s in periods with constant velocity. The motion is interrupted at different times for short periods (5 min, 1 min, and 2 min, respectively). It turns out that after short breaks the friction becomes significantly larger. This effect implies that not only temperature but also the distribution of lubricants determines friction conditions. That makes it difficult to guarantee equal conditions for the friction characteristics at two separate measurements. Actually, even during a precise friction measurement (which takes a few minutes) operating conditions might change. So, friction behavior cannot be exactly predicted. This can lead to systematic errors especially for rigid body identification but also for control.

An effect often described in the literature (Armstrong-Hélouvry 1991, de Wit et al. 1991, Swevers et al. 2000) is the so-called "Stribeck"-effect—a falling friction characteristic with increasing velocity. This effect occurs for very low velocities. For example, it can be modeled by the following equation:

$$\tau_{i,f4} = a_{i,1}\dot{q}_i + a_{i,2}\mathrm{sign}(\dot{q}_i) + a_{i,3}e^{-a_{i,4}\dot{q}_i}. \qquad (4)$$

This effect has indeed been observed for the *Siemens manutec-r15* (see Figure 4). Nevertheless, it is not included in our model due its strong dependency on operating conditions. The repeatability of friction is especially bad for low velocities. Comparisons between experiments and simulations have shown that the "Stribeck"-effect has only a minor influence on the dynamic behavior of the controlled system of the *Siemens manutec-r15* (Grotjahn and Heimann 1999).

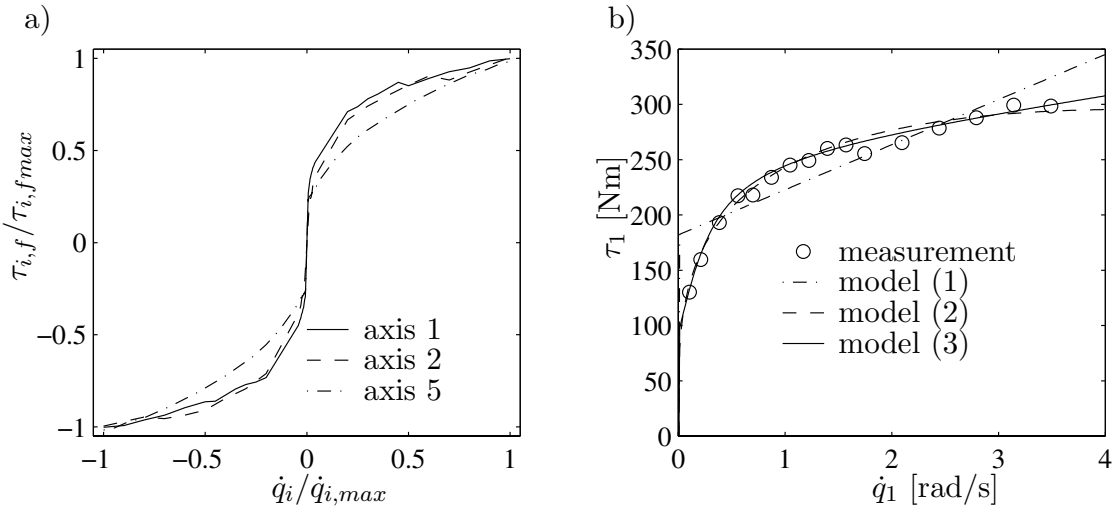

Fig. 2. Friction characteristics of the *Siemens manutec-r15*: (a) normalized characteristics of different axes; (b) adaption of friction models to measured characteristics of axis 1.

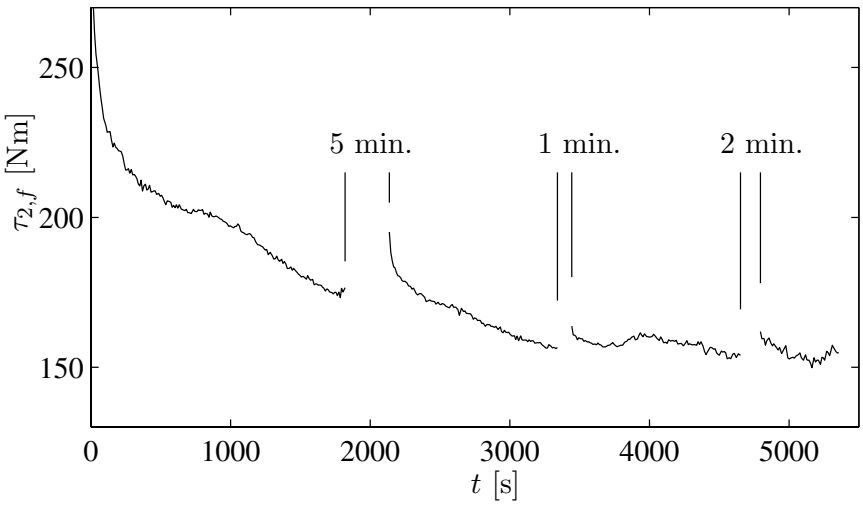

Fig. 3. Variations of friction during steadily back and forth motion with short breaks.

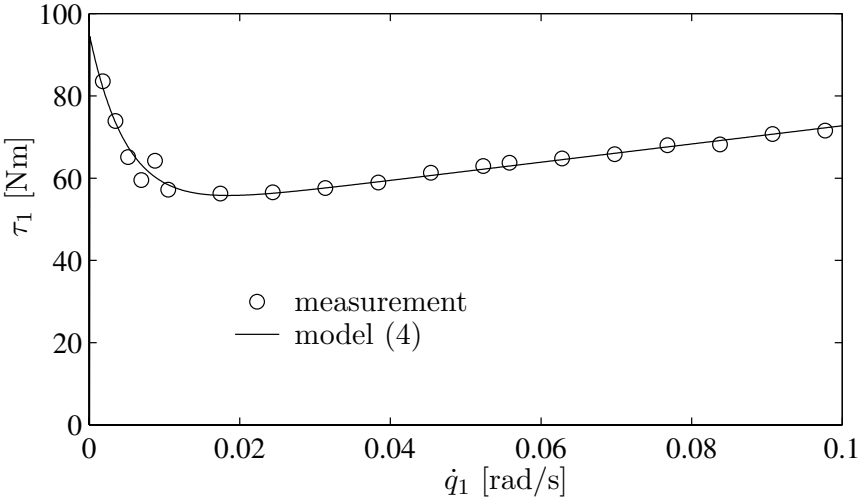

Fig. 4. Friction characteristic of axis 1 of the *Siemans manutec-r15* for low velocities.

3.1.2. Modeling and Identification of Rigid Body Dynamics

The dynamic equation of the robot's rigid body model can be written as

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) \quad \Longleftrightarrow \quad \boldsymbol{\tau} = \boldsymbol{A}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\,\boldsymbol{p}\,. \tag{5}$$

The left equation represents the usual form of the dynamic equation with the mass matrix $\boldsymbol{M}(\boldsymbol{q})$ as well as the vectors of centrifugal and Coriolis forces $\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}})$, gravitational $\boldsymbol{g}(\boldsymbol{q})$ and joint torques $\boldsymbol{\tau}$. The right equation is the corresponding parameter linear form. The base parameter vector $\boldsymbol{p}$ consists of the inertial and gravitational parameters of the links, e.g., masses and moments of inertia, and linear combinations of them. It has minimal order to guarantee identifiability of all elements p_i and can be derived automatically for any serial robot (Gautier and Khalil 1990). For typical industrial robots $\dim(\boldsymbol{p})$ is relatively small because of their symmetric link structure.

The base parameters can be divided into two groups: the gravitational parameter vector $\boldsymbol{p}_g$ comprises the parameters that occur in $\boldsymbol{g}(\boldsymbol{q})$, whereas the inertial parameter vector $\boldsymbol{p}_m$ consists of those parameters that only influence $\boldsymbol{M}(\boldsymbol{q})$ but not $\boldsymbol{g}(\boldsymbol{q})$. This division leads to the following formulation of the equations of motion:

$$\boldsymbol{\tau} = \underbrace{\boldsymbol{A}_{M,m}(\boldsymbol{q}, \ddot{\boldsymbol{q}})\,\boldsymbol{p}_m + \boldsymbol{A}_{M,g}(\boldsymbol{q}, \ddot{\boldsymbol{q}})\,\boldsymbol{p}_g}_{\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}} + \underbrace{\boldsymbol{A}_c(\boldsymbol{q}, \dot{\boldsymbol{q}})\,\boldsymbol{p}}_{\boldsymbol{c}(\boldsymbol{q},\dot{\boldsymbol{q}})} + \underbrace{\boldsymbol{A}_g(\boldsymbol{q})\,\boldsymbol{p}_g}_{\boldsymbol{g}(\boldsymbol{q})}\,. \tag{6}$$

There exists a vast amount of literature on the identification of $\boldsymbol{p}$. However, most of the methods are variations of the same identification scheme. The robot is moved along a trajectory which is optimized to guarantee maximum "excitation" of the parameters. Joint motion and torque are measured and friction is compensated by a model determined in advance.

This compensation leads to systematic errors due to the time variance of friction. Another problem, which is more important, is the fact, that optimized trajectories are needed. Standard industrial controls can only generate very simple trajectories. Even if the optimized trajectories are based on simple joint motions, like polynomials (Daemi and Heimann 1996) or Fourier series (Swevers et al. 1997), the practical realization is complicated. Therefore, optimized trajectories can be used in the field of robotics research but usually not in industrial robotics.

Therefore, another approach is suggested here. The method consists of two steps. The method requires only very simple trajectories and is, therefore, simply implementable in industrial robotics. It is based on the grouping in eq. (6).

In the first step, gravitational torques and moments of inertia are "measured" for a number of different joint configurations, so-called operating points. Each measurement is carried out by moving one single axis "back-and-forth" along some trapezoidal velocity profile in the neighborhood of the operating point. Such trapezoidal trajectories are very simple to generate by standard industrial controls. No specialized trajectories are needed, as only one property has to be identified and the trajectories are chosen to excite just this property.

For measurements of gravitational torque, long periods with constant velocity have to be included. The mean of the averaged torque at forward and backward motion gives the desired gravitational torque. For the measurement of the moments of inertia, motions with a higher share of acceleration are used. Gravitation is compensated and the moment of inertia is identified in connection with a simple friction model. The operating points are chosen by analyzing the structure of $\boldsymbol{M}(\boldsymbol{q})$ and $\boldsymbol{g}(\boldsymbol{q})$ to "excite" all dominant dependencies on the parameters $\boldsymbol{p}$ in order to include them in the estimation.

In the second step, the measurements of the gravitational torques and the moments of inertia are combined. Then, the parameters are estimated by using a weighted least squares criterion. For more details on the identification method see Daemi and Heimann (1998) and Grotjahn et al. (2001).

A comparison of the results for the *Siemens manutec-r15* in Daemi and Heimann (1996) and in Grotjahn et al. (2001) reveals that the two-step method leads to identification results which are similar to those of the conventional approach. The disadvantage of the two-step approach is the higher measurement effort. In contrast to the conventional approach, which needs only one trajectory, many experiments have to be performed. For the *Siemens manutec-r15* the procedure takes about 45 minutes. But, on the other hand, no optimization is necessary as the choice of operating points could be performed "by hand." The single measurements are very simple. This makes the approach applicable to standard industrial robot systems, which is the decisive advantage of the method.

3.2. Linear Modeling

Lange and Hirzinger (1994, 1996, 1999a,b) proposed the use of linear decoupled impulse response models of each joint's closed loop for reduction of path deviations. The model predicts the actual joint positions $q_i(k)$ based on the desired positions $q_{d,i}(k-l)$ for $l = m_{d,i}, .., m_i$:

$$q_i(k) = \sum_{l=m_{d,i}}^{m_i} g_i(l)\,q_{d,i}(k-l). \tag{7}$$

Here $g_i(l)$ denotes the estimate of one impulse response coefficient and $m_{d,i}$ the relative degree of the system. Theoretically, m_i must be infinite but for stable systems the first few $g_i(l)$ are sufficient. The joint index i is dropped for simplicity in the following.

The impulse response can be estimated using the extended LS-method or the Inverse-Covariance Kalman Filter (Maybeck 1979) by measuring one simple trajectory. Here, we used simple sinusoidal joint trajectories. No additional knowledge

about the structure of the system or the controller is used during the identification process. This is an advantage on the one hand because the procedure can be implemented on unknown systems. On the other hand, it can result in inferior model quality since available information about the controller and the physics of the system is not taken into consideration. Due to its linearity, the model of the robot is incapable of incorporating the nonlinear effects, especially friction. Nevertheless, it is possible to compensate even nonlinear effects by these simple models (see Section 4.2). This is achieved by taking into account measured tracking errors. Thus, the measured data and not the model contain the information on the nonlinear influences.

4. Feedforward Control

The compensation of nonlinear dynamics can only be performed by using feedforward controllers, because there is usually no possibility of changing the feedback controllers' structure of standard industrial controls. Conventionally, only the desired path can be changed by the feedforward controller, as no other interface exists (see Section 5.2.1). In the following section, a nonlinear feedforward control strategy, called nonlinear precorrection, is introduced, which is based on the friction and rigid-body model presented in Section 3.1. Subsequently, two compensation methods are presented which use linear models for improvement of path accuracy. Finally, the different methods are experimentally compared by application to the *Siemens manutec-r15*.

4.1. Nonlinear Precorrection

For nonlinear precorrection an inverse model of the closed loop system is used which consists of friction, rigid body dynamics and the known controller parameters. The necessary torques $\boldsymbol{\tau}_d$ for a desired trajectory $\boldsymbol{q}_d$ can be estimated using eqs. (2) or (3) and (5):

$$\begin{aligned}\boldsymbol{\tau}_d &= \boldsymbol{M}(\boldsymbol{q}_d)\ddot{\boldsymbol{q}}_d + \boldsymbol{c}(\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d) + \boldsymbol{g}(\boldsymbol{q}_d) + \boldsymbol{\tau}_f(\dot{\boldsymbol{q}}_d) \\ &= \boldsymbol{A}(\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d, \ddot{\boldsymbol{q}}_d)\boldsymbol{p} + \boldsymbol{\tau}_f(\dot{\boldsymbol{q}}_d). \end{aligned} \tag{8}$$

The vector $\boldsymbol{\tau}_f = \left[\tau_{1,f}, \ldots, \tau_{n,f}\right]^T$ comprises the friction losses of the different joints. The inverted controller model $\boldsymbol{F}_{contr}^{-1}$ converts the desired torques into corrections $\boldsymbol{\Delta q}_{cor} = \boldsymbol{F}_{contr}^{-1}\boldsymbol{\tau}_d$ that have to be added to the desired trajectory to achieve improved path accuracy. The structure of the complete system can be seen in Figure 5.

The overall structure of the trajectory precorrection resembles the well known computed-torque feedforward control (Sciavicco and Siciliano 1996):

$$\begin{aligned}\boldsymbol{\tau}_{act} &= \boldsymbol{F}_{contr}(\boldsymbol{\Delta q}_{d,cor} + \boldsymbol{q}_d - \boldsymbol{q}) \\ &= \boldsymbol{\tau}_d + \boldsymbol{F}_{contr}(\boldsymbol{q}_d - \boldsymbol{q}) \\ &= \boldsymbol{A}(\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d, \ddot{\boldsymbol{q}}_d)\boldsymbol{p} + \boldsymbol{\tau}_f(\dot{\boldsymbol{q}}_d) + \boldsymbol{F}_{contr}(\boldsymbol{q}_d - \boldsymbol{q}). \end{aligned} \tag{9}$$

The advantage of nonlinear trajectory precorrection compared to the computed-torque method is that one only needs to convey path information to the robotic system and no additional torque information. Only a path interface is necessary. So, the approach is applicable to standard industrial robot systems since such an interface is usually provided for integration of external sensor information into the control circuit. A torque interface, however, is normally not available without a costly change of the controller.

4.2. Feedforward Control by Linear Models

Although the linear models disregard nonlinearities, they can be used to their compensation by taking into account actual path deviations. The advantage is that arbitrary effects can be compensated, even effects which are not included in the complex nonlinear model. This is done by a "learning" algorithm that is presented in the following. Afterwards, the "training" of a feedforward joint controller is explained.

4.2.1. Linear Learning

"Learning" is based on the idea that measured path deviations can be used to adjust the trajectory that is transmitted to the controller so that the actual trajectory comes closer to the originally desired one. Thereby, path accuracy is improved by iterative optimization (Arimoto et al. 1984, Gorinevsky 1992, Gorinevsky et al. 1997) (see block diagram, Figure 6). With the desired trajectory $\boldsymbol{q}_d = [q_d(1), .., q_d(N)]^T$, the last commanded trajectory $\boldsymbol{q}_{d,cor}^p = [q_{d,cor}^p(1), .., q_{d,cor}^p(N)]^T$, which results from the last iteration p, and the actual trajectory $\boldsymbol{q}^p = [q^p(1), .., q^p(N)]^T$, the new corrected trajectory, are calculated in the general case by

$$\boldsymbol{q}_{d,cor}^{p+1} = \boldsymbol{q}_{d,cor}^p + \boldsymbol{f}(\boldsymbol{q}_d, \boldsymbol{q}^p) = \boldsymbol{q}_{d,cor}^p + \boldsymbol{\Delta q}_{d,cor}^{p+1}. \tag{10}$$

If the impulse response models (eq. (7)) are used for "learning," the following system of linear equations arises:

$$\begin{pmatrix} g(d) & 0 & \cdots & \\ g(d+1) & g(d) & 0 & \\ \vdots & & \ddots & \\ & g(m) & \cdots & g(d) \end{pmatrix} \begin{pmatrix} \Delta q_{d,cor}^{p+1}(1-d) \\ \Delta q_{d,cor}^{p+1}(2-d) \\ \vdots \\ \Delta q_{d,cor}^{p+1}(N-d) \end{pmatrix} = \begin{pmatrix} q_d(1) - q^p(1) \\ q_d(2) - q^p(2) \\ \vdots \\ q_d(N) - q^p(N) \end{pmatrix}$$

$$\Longleftrightarrow \boldsymbol{G}\boldsymbol{\Delta q}_{d,cor}^{p+1} = \boldsymbol{e}^p. \tag{11}$$

In order to solve this system of equations, the matrix $\boldsymbol{G}$ has to be inverted, which is equivalent to an inversion of the joint's closed-loop. As it can not be guaranteed that the identified

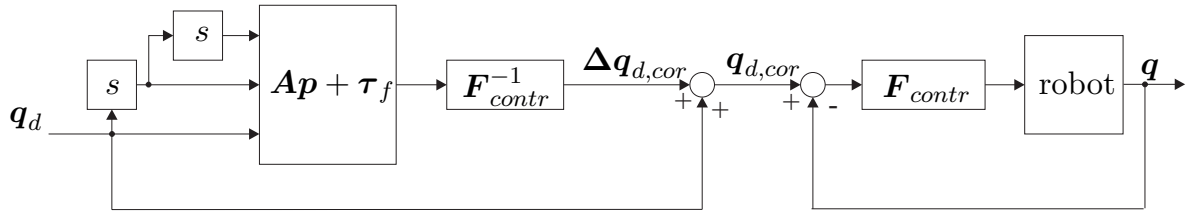

Fig. 5. Structure of the complete robot system with nonlinear trajectory precorrection.

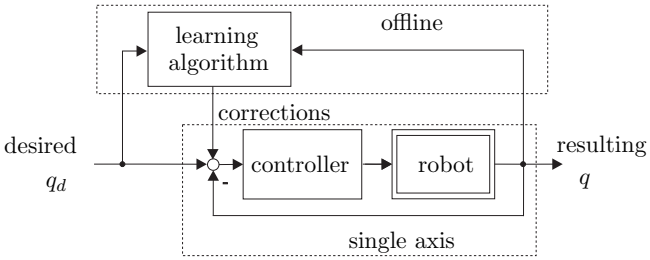

Fig. 6. Block diagram of "learning."

system is phase-minimum, an exact inversion is not always possible. In that case, the resulting corrected trajectory would include unacceptable oscillations. To avoid this, the extended LS-method or the Inverse-Covariance Kalman-Filter is again used. Norrlöf and Gunnarson (2000) propose a simple Butterworth filtering of the resulting desired trajectory. Other approaches for the inversion of non phase-minimum systems can be found in Wagner (1999).

The main advantage of "learning" is the fact that, by using the actual path errors, all systematic deviations influence the corrections. That means that influences can be compensated which are not reflected by the model. This holds not only for effects not modeled by the linear models used for learning, but also for effects not even reflected by the complex nonlinear model presented here. Another advantage is that the actual deviations of end-effector position and orientation can be taken into account if they are measurable. This is done by calculating motor deviations from path errors by using the Jacobian (Lange and Hirzinger 1999a,b).

The main disadvantage, however, is that every small change of the desired trajectory necessitates a new learning process. Furthermore, each learning process is relatively costly since the new trajectory has to be driven several times and convergence of the presented learning algorithm cannot be guaranteed. Therefore, an application by industrial users does not seem to be possible.

4.2.2. Training of a Feedforward Controller

In order to avoid the disadvantages of "learning" described in the previous section, Lange and Hirzinger proposed to "train" a feedforward controller from the learned behavior in Lange and Hirzinger (1994). For this, a feedforward controller model is identified by using again the extended LS-method or the Inverse-Covariance Kalman-Filter. For a block diagram see Figure 7.

As a matter of principle, a feedforward controller has the advantage that it can be calculated in real time. Therefore, it can be implemented in commercial controls. Furthermore, a powerful feedforward controller offers the possibility of transfering the "learned" behavior to similar trajectories. With that, small trajectory changes would not require new "learning."

The choice of the feedforward controller structure is a crucial point. A fundamental condition is that the model can satisfactorily reproduce the "learned" behavior. Lange and Hirzinger (1994) again used linear models:

$$q_{d,cor}(k) = q_d(k) + \sum_{l=m_t}^{m} r(l)(q_d(k+l) - q_d(k)). \qquad (12)$$

This approach, however, cannot compensate nonlinear effects, like friction. Figure 8(a) shows that the identification of the linear feedforward controller leads to a mixture of two effects. In addition to inertial influences, friction has a large impact at the beginning of the motion. Therefore, the corrections of the linear controller are too small in the beginning and too large at later zero-crossing of velocity. To improve this, the approach is extended by another term:

$$q_{d,cor}(k) = q_d(k) + \sum_{l=m_t}^{m} r(l)(q_d(k+l) - q_d(k)) + \sum_{l=p_t}^{p} s(l)\,(\mathrm{sign}(\dot{q}_d(k+l)) - \mathrm{sign}\,(\dot{q}_d(k)))\,. \qquad (13)$$

This auxiliary summand is suited to separate friction from inertial influences. With it, "learned" corrections are reproduced much better than by the linear model (see Figure 8(a)). As expected, this results in a much better tracking behavior (see Figure 8(b)). Although the "training" does not yield as good results as "learning," it seems to be an alternative which combines good results with simplicity.

4.3. Results for Siemens manutec-r15

The presented compensation methods are investigated by experimental application to the *Siemens manutec-r15*, in order

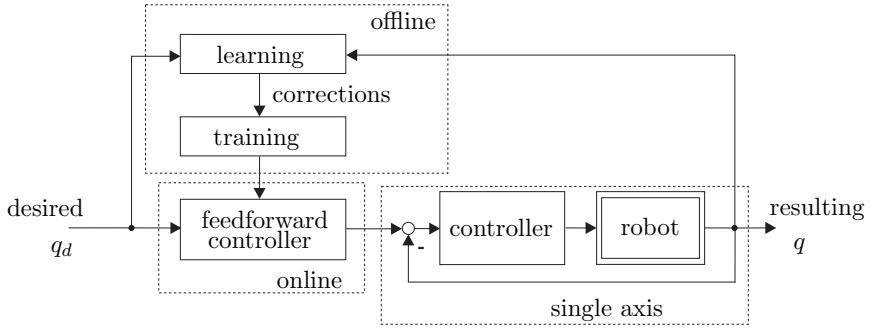

Fig. 7. Block diagram of "training."

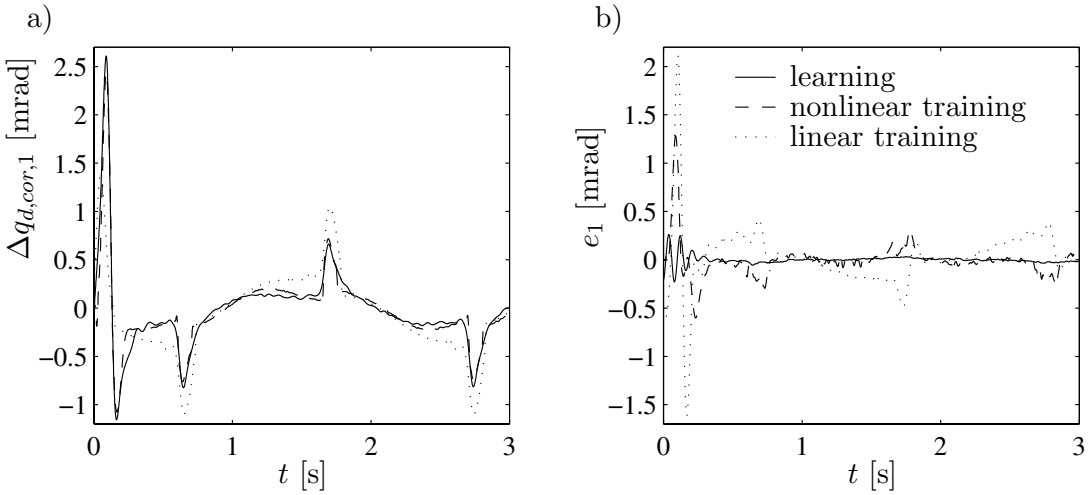

Fig. 8. Behaviour of the first axis of the *Siemans manutec-r15* for a vertical circle: (a) "learned" and "trained" corrections; (b) comparison of resulting tracking errors.

to evaluate their performance and their practical applicability. For that, several different test trajectories are used. Here, the results are explained by regarding two trajectories: a vertical circle and an edge in the x-y plane with a change of orientation by 90°.

The vertical circle has a diameter of 40 cm and a path velocity of 0.6 m/s. This means that the circle is approximately completed after 2.1 s. The cartesian path deviations $e_{car}(t) = (e_x^2(t) + e_y^2(t) + e_z^2(t))^{1/2}$ are depicted in Figure 9. It shows some general results which can be reproduced for all tested trajectories. The "training" yields the worst results of all methods. The nonlinear precorrection is much better. The "learning" even yields slight improvements.

In order to numerically evaluate the results the following error criterion is defined:

$$E_{car} = \left(\frac{1}{N} \sum_{j=1}^{N} \left[(x_d(j) - x(j))^2 + \left(y_d(j) - y(j) \right)^2 + (z_d(j) - z(j))^2 \right] \right)^{1/2}. \tag{14}$$

For all investigated trajectories, "learning" leads to a decrease of at least 80% after four iterations. The nonlinear precorrection reduces the criterion by at least 60%, whereas the "training" leads to a minimum reduction of only 35%.

Figure 10 reveals the reasons for these bad results for the "training." It shows the behavior for a motion where the end-effector is driven along an edge in the x-y plane and the orientation is changed by 90°. Although the path velocity is only 0.02 m/s, joint velocities and accelerations are quite high. Therefore, the couplings between different links have a strong impact. These effects can not be compensated by the decoupled "trained" feedforward joint controllers. Lange and Hirzinger proposed the use of neural networks to take these effects into account (1996, 1999b). Neural nets, however, are difficult to apply in industrial robotics because of their complexity and lack of robustness with respect to the use of arbitrary trajectories.

So, nonlinear precorrection is the only approach which combines efficient compensation of nonlinear deviations with practical applicability. "Learning" yields slightly better results. But robustness and stability properties must be

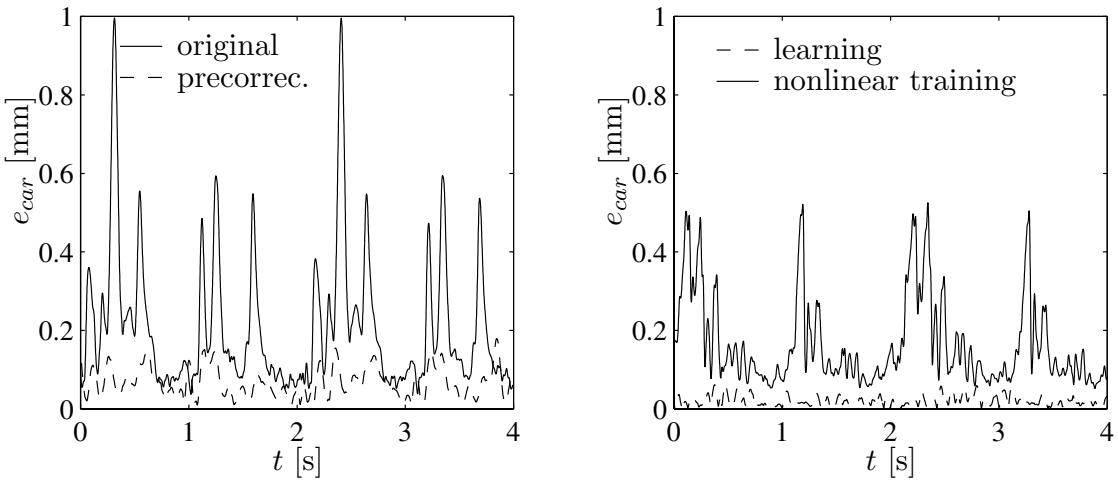

Fig. 9. Comparison of cartesian path errors for a vertical circle of the *Siemans manutec-r15*.

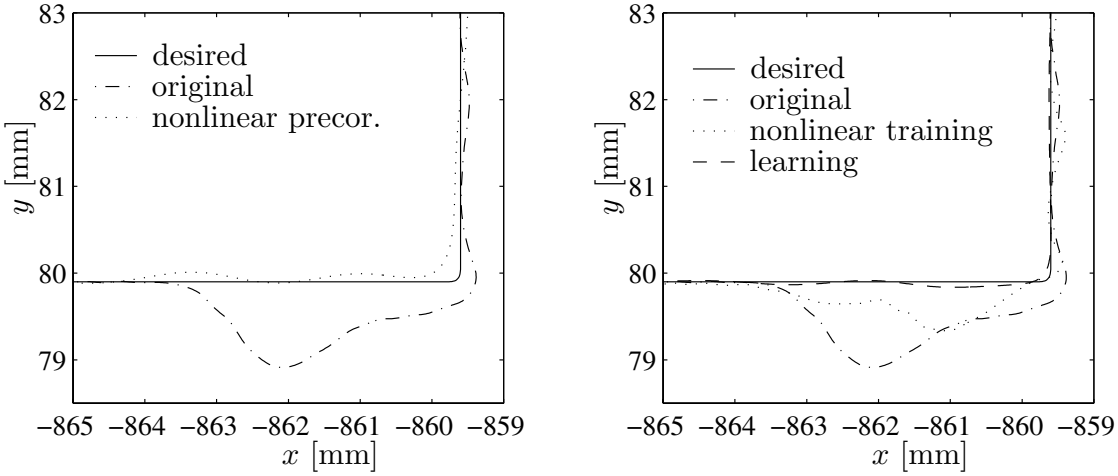

Fig. 10. Path behaviour of the *Siemans manutec-r15* for an edge in the x-y-plane.

improved for the integration in standard industrial controls. The proposed nonlinear "training" combines simplicity with applicability but is only suitable for slow trajectories for which couplings between the different links have only low impact. Therefore, only the nonlinear precorrection is applied to the *KUKA KR15* and integrated in the standard KRC1 control.

5. Application to KUKA-KR15

In order to prove the applicability of nonlinear precorrection in the area of industrial robotics it is implemented in a "state-of-the-art" commercial robotic system. The aim was to use no additional hardware, just to use the possibilities that the standard control KRC1 offers.

5.1. Identification of Robot Dynamics

As described in Section 3.1, the friction and rigid body models are separately identified. The results are summarized below.

Figure 11(a) depicts the normalized friction behavior of different joints of the *KUKA KR15*. Similar to the *Siemens manutec-r15*, the *KUKA KR15* shows degressive friction characteristics. Especially for the main axes, however, the degressive characteristic is not as significant as for the *Siemens manutec-r15* but still obvious. Thus, Figure 11(b) reveals that the friction models, eq. (2) and (3), can reflect the behavior much better than the classic model (eq. (1)).

The time variance of friction significantly complicates the modeling of the *Siemens manutec-r15*. The *KUKA KR15* also showed an obvious dependence on operating conditions and gear temperature respectively. In Figure 12 the friction

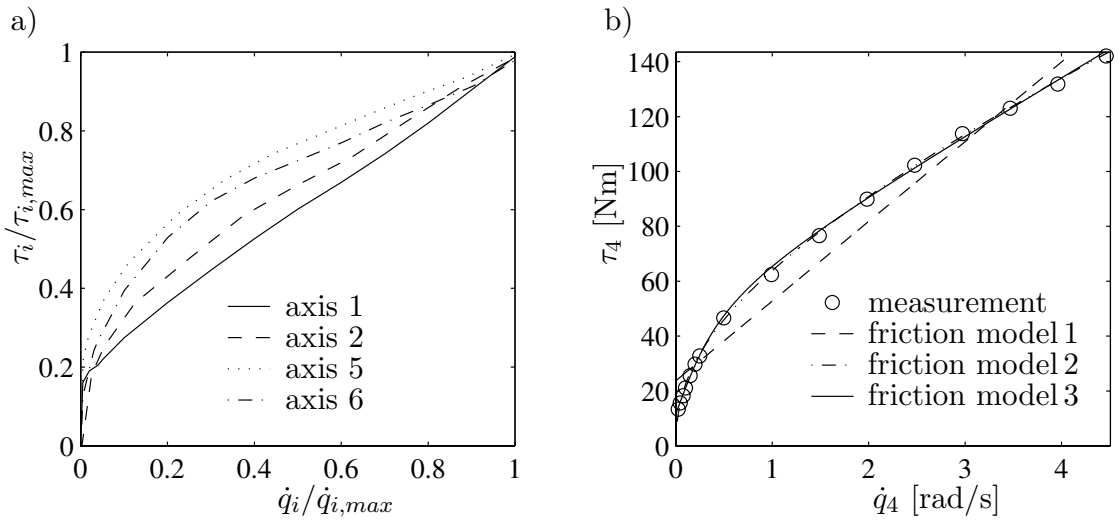

Fig. 11. Friction characteristics of the *KUKA KR15*: (a) normalized characteristics of different joints; (b) friction modeling for the fourth joint.

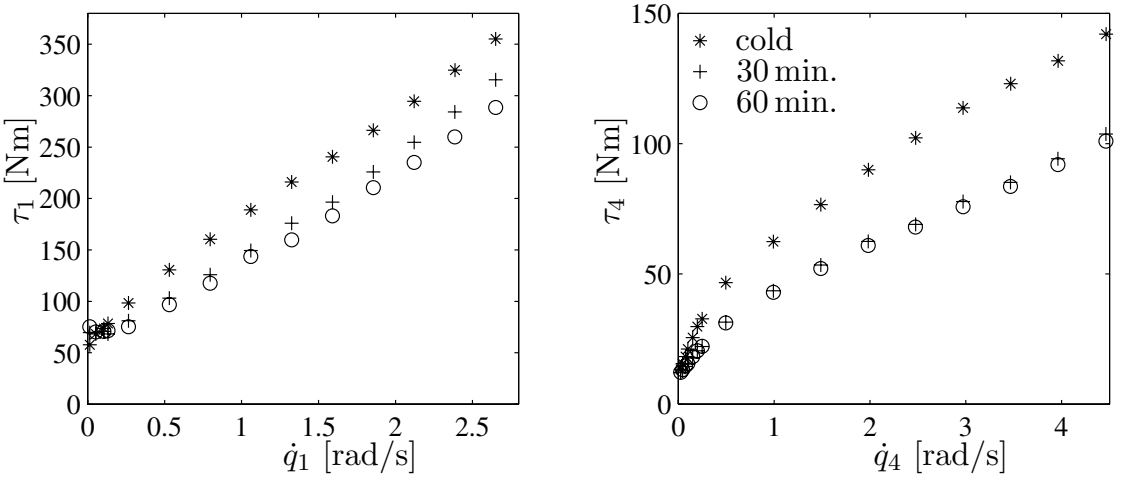

Fig. 12. Time variance of friction with respect to operating conditions for the first and the fourth joint of the *KUKA KR15*.

characteristics of the first and fourth joint are depicted. The characteristics are measured directly after switch on, after thirty minutes and after sixty minutes of continuous operation. In contrast to the *Siemens manutec-r15*, there are only small differences between the measurements after 30 min. and the measurements after 60 min. This implies stationary conditions. Therefore, the usual approach of "warming up the robot" seems to be acceptable.

Nevertheless, the rigid body model is identified by the two-step method described in Section 3.1. Simple trajectories with trapezoidal velocity profile are used. During the motion motor angles and currents are recorded using the so-called "oscilloscope"-function of the KRC1 control. In the first step, the measurements are used to identify gravitational and inertial torques for several different joint configurations. In the second step, the overall parameter vector p is identified from these single measurements.

For model validation, test trajectories are used which strongly differ from the trajectories used for identification. In Figure 13 a comparison is presented between the torques calculated by the model and the measured torques. The presented trajectory is defined with respect to ISO standard 9283 (ISO 1996). The respective path is presented in Figure 14. The cartesian path velocity is 500 mm/s.

The experiments show very good agreement between model and reality. Only for the hand axes do some small differences occur. These differences result from the stronger impact of friction which cannot be predicted as precisely as

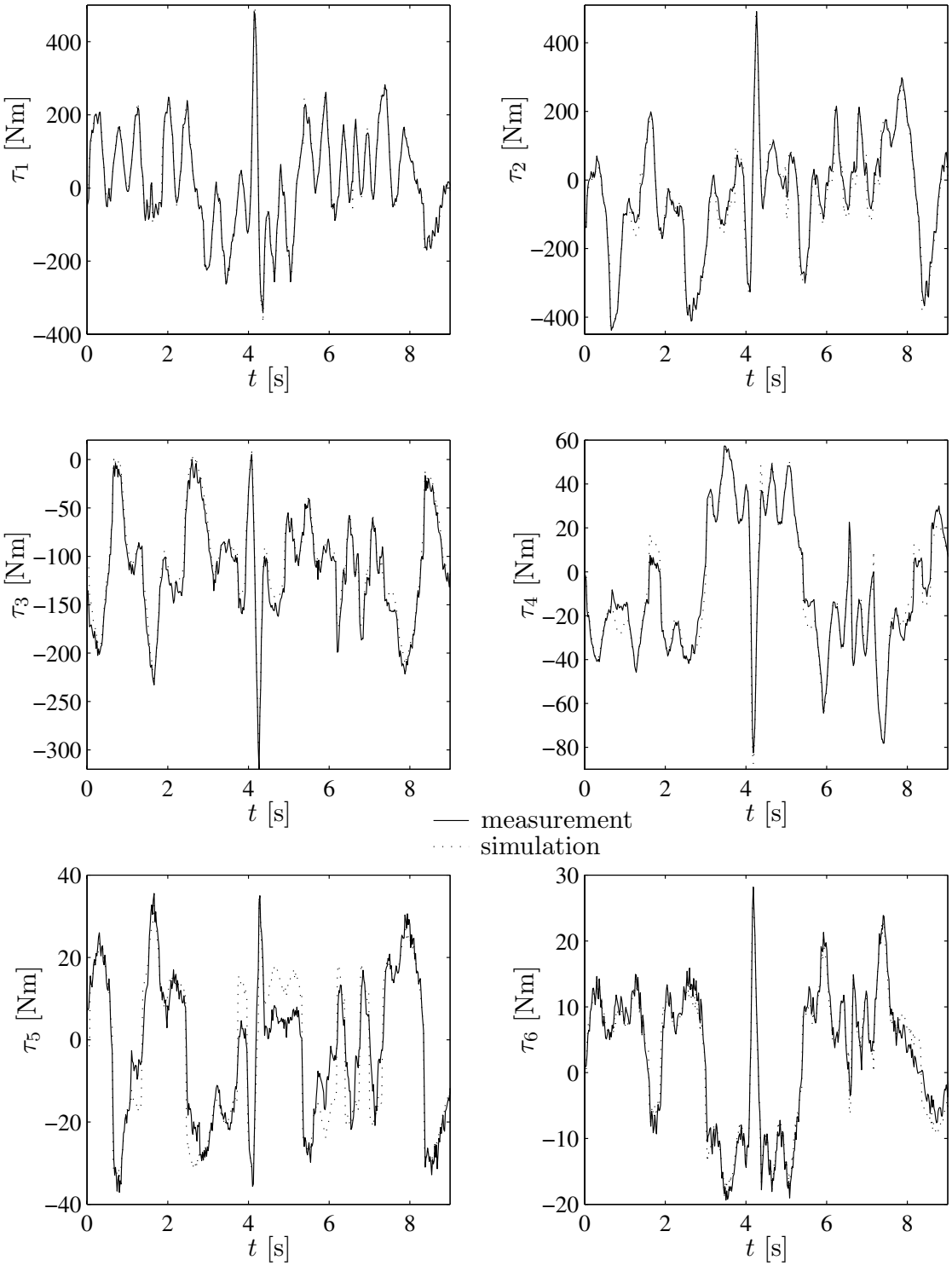

Fig. 13. Comparison between measured and calculated torques for ISO 9283-trajectory.

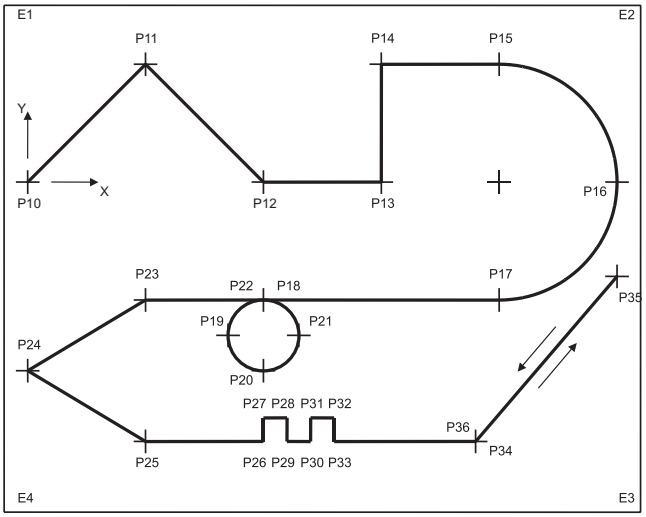

Fig. 14. ISO 9283-trajectory (edge length: 400 mm; inclined by 45°; see appendix (A.5) of ISO (1996) for details).

rigid body dynamics. Nevertheless, these results are an excellent base for the compensation of nonlinearities of the *KUKA KR15*.

5.2. Implementation within KRC1 Control

The KUKA KRC1 control offers the so-called "sensor task" for implementation of arbitrary C-code. The task is originally designed for integrating external sensor signals, like force measurements, into the control loop. It allows a change in the desired trajectory but no direct access to motor currents. In the next section, the control structure is presented. Afterwards, the identification of joint controllers and problems of the implementation within the "sensor task" are discussed.

5.2.1. Control Structure

The control architecture is split into a central part and six independent linear joint controllers. The central part runs on a Pentium II-400 MHz processor. It consists of safety and user-interface functions, as well as the calculation of desired values for the joint controllers and the execution of the "sensor task."

These values are calculated from the desired cartesian path by solution of the inverse kinematics. The inverse kinematics are solved every 12 ms, the so-called "interpolation (IPO) cycle." Afterwards the values are interpolated in order to fit to the joint controller cycle time of 2 ms. The "sensor task" is executed directly after the interpolation. Path changes made by the "sensor task" are added to the desired trajectory in the next step of the IPO cycle. This causes an inevitable time delay of 12 ms. Finally, the interpolated joint values are low-pass filtered in order to eliminate shocks and jerks. As the filter is not phase-minimum, it is not inverted for precorrection. Instead, its time constant is reduced as much as possible. The remaining influence is neglected.

The resulting desired trajectory is transferred to the joint controllers. According to the control manufacturer, the joint controllers are standard linear, cascaded P/PI controllers. The identification, however, yields slightly different results (see Section 5.2.2). The outer positional loop has a cycle time of 2 ms whereas the inner velocity loop runs with a 0.5 ms cycle. The actual velocity is reconstructed by a low-pass differentiator. For block diagram of the over-all control structure see Figure 15.

5.2.2. Identification of Joint Controllers

As structure and parameters of the joint controllers are usually not known by the user in advance it is necessary to identify them. The actual controller cycle time (2 ms) and the IPO cycle time (12 ms), which must be used for inversion and for identification, are different by a factor of six. This is a problem for identification because input signals are not constant during one time step. Nevertheless, it is possible to obtain good model quality for low-band dynamics.

In combination with the different cycle times, the integrator makes controller identification difficult because the integrator input is not constant during one IPO cycle. This leads to accumulating errors if the integrator is approximated with a higher cycle time. Therefore, the proportional and derivative feedback gains are identified separately from the integral gain. To achieve this, the integral gain is set to zero and a PD-controller of the form

$$u_{PD}(k) = k_P e(k) + k_D \dot{e}(k) \quad \text{with} \quad e(k) = q_d(k) - q(k) \tag{15}$$

is identified. The derivative is approximated by numerical differentiation of the form $\dot{x}(k) = (x(k) - x(k-1)/T_{IPO}$. The time constant of the low-pass differentiator is neglected because it is much smaller than the IPO cycle time.

Experimental investigations reveal that the real behavior is different from the structure given by the manufacturer. The controller output contains additional influences strongly correlated with velocity and acceleration. Thus, the model is extended by two terms:

$$u(k) = k_P e(k) + k_D \dot{e}(k) + k_1 \dot{q}(k) + k_2 \ddot{q}(k). \tag{16}$$

The model is identified by a simple LS-approach using different trajectories. All experiments yield similar results. Figure 16 shows a comparison between measurement and simulation. A good model quality is achieved in spite of the large sampling time.

For the identification of the integrator gain, the PD-controller is connected in series with a PI-controller. The proportional gain is one, so that only one parameter has to be determined. The motor current $i(k)$ can be calculated from

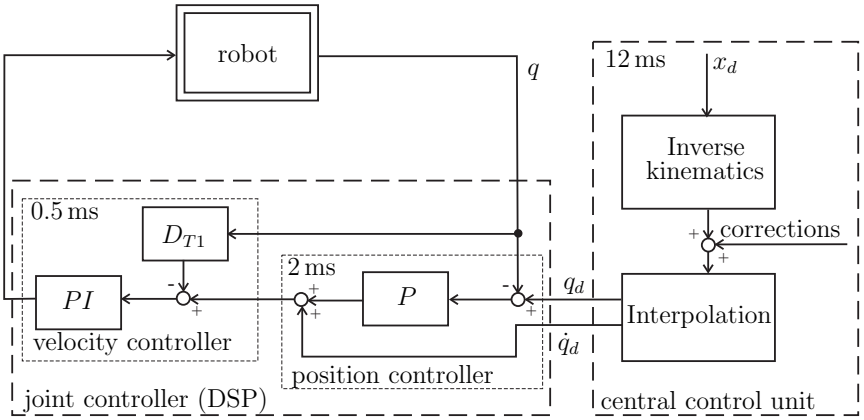

Fig. 15. Control architecture of the *KUKA KR15*.

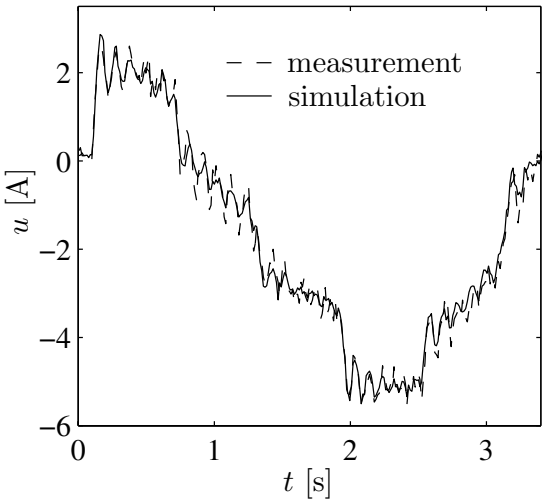

Fig. 16. Comparison of measurement and simulation for one joint PD-controller of the *KUKA KR15*.

the output of the PD-controller $u(k)$ by

$$i(k) = i(k-1) + (1+k_I)u(k) - u(k-1). \tag{17}$$

If k_I is directly estimated by a LS-approach only the one step prediction of the model is considered. Figure 17 shows that this leads to drift as modeling errors accumulate. Therefore, k_I is estimated by simulation of the controller behavior. The gain k_I is optimized by numerical evaluation of the overall quadratic error. Sophisticated optimization approaches could be used for this approach. But, as the range of reasonable values of k_I is small, a simple evaluation of the LS-criterion for a variety of value of k_I is sufficient. Figure 17 reveals that this approach yields much better results which, of course, holds also for the inversion of the model.

5.2.3. Implementation within Sensor Task

The precorrection scheme is implemented using the C programming language. Although execution time is restricted it is possible to calculate the complete inverse dynamics consisting of eqs. (3) and (5). Assuming proportionality, the desired motor currents i_d can be calculated from the desired torques τ_d and the motor constants given by the manufacturer.

In the next step, the controller eqs. (16) and (17) are inverted. For that, the following substitutions have to be made:

$$q_d := q_{d,cor}, \quad q := q_d, \quad e := \Delta q_{d,cor},$$
$$u := u_d \quad \text{and} \quad i := i_d.$$

This leads to the following equations:

$$u_d(k) = \frac{i_d(k) - i_d(k-1) + u_d(k-1)}{1+k_I} \tag{18}$$

$$\Delta q_{d,cor}(k) = \tag{19}$$
$$\frac{u_d(k) + k_D \Delta q_{d,cor}(k-1)/T_{IPO} - k_1 \dot{q}_d(k) - k_2 \ddot{q}_d(k)}{k_P + k_D/T_{IPO}}$$

$$q_{d,cor}(k) = q_d(k) + \Delta q_{d,cor}(k). \tag{20}$$

Equation (18) shows that u_d converges to zero in the static case for a reasonable choice of $k_I > 0$. This means that the static accuracy is not affected by the approach.

As mentioned in Section 5.2.1, the corrections are transferred to the joint controllers with a time delay of 12 ms. The only possibility to avoid this time-delay is the deactivation of the standard path-planner. For this, the original desired trajectory is recorded once. Then, not only the corrections $\Delta q_{d,cor}(k)$ but also $q_{d,cor}(k)$ are calculated in the sensor task by adding the stored $q_d(k)$ and the calculated $\Delta q_{d,cor}(k)$. Thus, the influence of the time delay can be experimentally investigated by comparing the results with and without standard path-planner respectively.

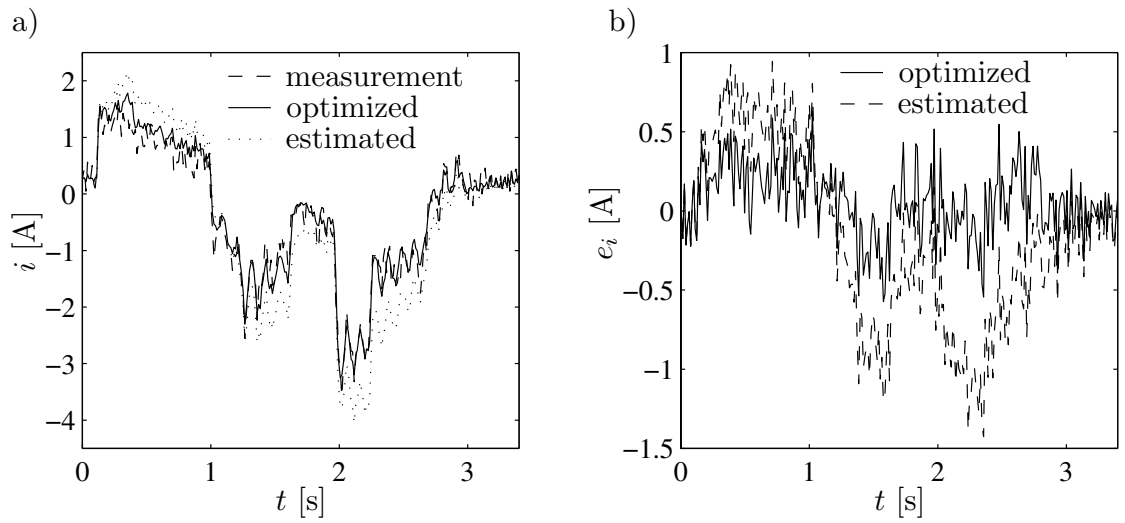

Fig. 17. Simulation results for PID-controller with "directly estimated" and "optimized" integrator gain: (a) absolute values; (b) deviations.

5.3. Experimental Results

The effectiveness of the presented approach is investigated by using trajectories designed with respect to the ISO 9283 standard (ISO 1996). For all trajectories, the end-effector is moved in a plane in the middle of the workspace, which is inclined by an angle of 45°. Three test trajectories are used. Trajectory 1 is given in Figure 14, and trajectories 2 and 3 are located in a cube with an edge length of 400 mm. Trajectory 2 is a triangle with an edge length 320 mm and trajectory 3 is a circle with a diameter of 320 mm. The cartesian speed of trajectory 1 is 500 mm/s, the speed of the other trajectories is 630 mm/s.

Figure 18(a) shows the remaining positional errors for trajectory 1. The improvements are significant. In particular, the reduction of the errors of the base axes leads to these excellent results. This is pointed out by Figure 18(b), in which the robot's behavior is depicted for a single axis motion of the second joint. The results for the hand axes are not as good. But their influence on positional accuracy is much smaller so that this worsens the path accuracy only slightly. For quantitative evaluation the previously defined cartesian error criterion (eq. (14)) and maximum position error

$$E_{max} = \max \sqrt{(x_d(j) - x(j))^2 + (y_d(j) - y(j))^2 + (z_d(j) - z(j))^2} \tag{21}$$

are used. Table 1 shows the results for the different trajectories. The maximal error is reduced by at least 30%, criterion (eq. (14)) by more than 60%. The remaining deviations result particularly from improper friction modeling (see Section 3.1.1) and the remaining influence of low-pass filtering of the desired trajectory (see Section 5.2.1).

A closer look at Table 1 reveals the influence of the time delay resulting from the shift by one IPO-cycle in the case that the original path-planner is used. There is no explicit difference for the maximal errors, whereas the error criterion (eq. (14)) shows that the average path deviations increase because of the time delay. Nevertheless, even with the conventional path-planner the results are excellent in comparison to the original results. This proves the applicability of the presented method in industrial robotics.

6. Conclusions

In this paper different algorithms for model-based feedforward control in industrial robotics are presented. The different methods are experimentally compared by application to the industrial robot *Siemens manutec-r15*. The so-called nonlinear precorrection is an adaptation of the well known computed-torque method. It combines excellent results with practical applicability. Therefore, it is implemented within the "state-of-the-art" industrial robotic system *KUKA KR15*. No additional hardware is used because the algorithm is integrated into the standard control. The experimental investigation yields excellent results and proves the applicability in the industrial field.

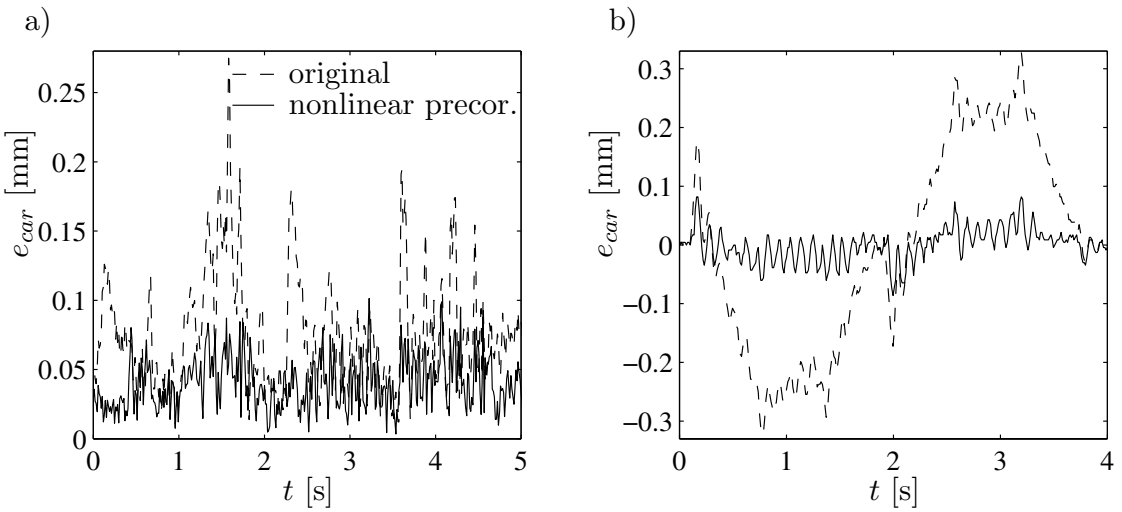

Fig. 18. Path deviations of the *KUKA KR15*: (a) ISO 9283 trajectory; (b) single motion of the second joint

Table 1. Results of Nonlinear Precorrection for *KUKA KR15* in mm

	Original		Corrected with Time Delay		Corrected without Time Delay	
	E_{max}	E_{car} (10^{-6})	E_{max}	E_{car} (10^{-6})	E_{max}	E_{car} (10^{-6})
Trajectory 1	0.27	6.7716	0.13	2.4065	0.19	1.9284
Trajectory 2	0.23	6.2481	0.14	2.9473	0.14	2.3570
Trajectory 3	0.17	6.8524	0.13	1.6579	0.10	1.4984

Acknowledgments

The authors work was funded by the German "Bundesministerium für Bildung und Forschung."

References

Arimoto, S., Kawamaru, S., and Miyazaki, F. 1984. Bettering operations of robots by learning. *Journal of Robotic Systems* 1(2):133–140.

Armstrong-Hélouvry, B. 1991. *Control of Machines with Friction*. Boston: Kluwer.

Bernhardt, R., Schreck, G., and Willnow, C. 1995. Realistic robot simulation. *Computing and Control Engineering Journal*, pp. 174–176.

Daemi, M., and Heimann, B. 1996. Identification and compensation of gear friction for modeling of robots. *Proc. 11th CISM-IFToMM Symp. on the Theory and Practice of Robots and Manipulators,* pp. 89–96.

Daemi, M., and Heimann, B. 1998. Separation of friction and rigid body identification for industrial robots. *Proc. 12th CISM-IFToMM Symp. on the Theory and Practice of Robots and Manipulators,* pp. 35–42.

de Wit, C. C., Noel, P., Aubin, A., and Brogliato, B. 1991. Adaptive friction compensation in robot manipulators: Low velocities. *The International Journal Robotics Research* 10(3):189–199.

Gautier, M., and Khalil, W. 1990. Direct calculation of minimum set of inertial parameters of serial robots. *IEEE Transactions on Robotics and Automation* 6(3):368–373.

Gorinevsky, D. 1992. Experiments in direct learning of feedforward control for manipulator path tracking. *Robotersysteme* 8:139–147.

Gorinevsky, D., Torfs, D., and Goldenberg, A. 1997. Learning approximation of feedforward control dependence on the task parameters with application to direct-drive manipulator tracking. *IEEE Transactions on Robotics and Automation* 13(4):567–580.

Grotjahn, M., and Heimann, B. 1999. Prediction of dynamic path errors for industrial robots. *Proc. 8th Int. Workshop on Robotics in Alpe-Adria-Danube Region,* Munich, pp. 77–82.

Grotjahn, M., Daemi, M., and Heimann, B. 2001. Friction and rigid body identification of robot dynamics. *Int. Journal of Solids and Structures* 38:1889–1902.

Grotjahn, M., Heimann, B., and Thiemann, A. 1999. Reduction of path errors for industrial robots by trajectory precorrection. *Proc. 2nd Int. Conf. on Recent Advances in Mechatronics,* Istanbul, pp. 46–51.

ISO. 1996. *Manipulating industrial robots—performance criteria and related test methods*, DIN EN ISO 9283.

Lange, F., and Hirzinger, G. 1994. Learning to improve the path accuracy of position controlled robots. *Proc. Conf. on Intelligent Robots and Systems*, Munich, pp. 494–501.

Lange, F., and Hirzinger, G. 1996. Learning of a controller for non-recurring fast movements. *Advanced Robotics* 10(2):229–244.

Lange, F., and Hirzinger, G. 1999a. Learning accurate path control of industrial robots with joint elasticity. *Proc. IEEE Int. Conf. on Robotics and Automation ICRA'99,* Detroit, pp. 2084–2089.

Lange, F., and Hirzinger, G. 1999b. Adaptive minimization of the maximal path deviations of industrial robots. *Proc. European Control Conf. ECC'99,* Karlsruhe.

Maybeck, P. 1979. Stochastic Models, Estimation and Control, Vol. 1, *Mathematics in Science and Engineering,* New York: Academic.

Norrlöf, M., Gunnarson, S. 2000. A model based iterative learning control method applied to 3 axes of a commercial industrial robot. *Proc. 6th IFAC Symp. on Robot Control (SYROCO),* Vol. 1, pp. 1–6.

Pfeiffer, F., and Hölzl, J. 1995. Parameter identification for industrial robots. *Proc. IEEE Int. Conf. on Robotics and Automation,* Nagoya, pp. 1468–1476.

Prüfer, M., and Wahl, F. M. 1994. Friction analysis and modelling for geared robots. *Proc. 4th IFAC Symp. on Robot Control,* Vol. 2, Capri, Italy, pp. 551–556.

Sciavicco, L., and Siciliano, B. 1996. *Modeling and Control of Robot Manipulators*, New York: McGraw Hill.

Seeger, G. 1992. Selbsteinstellende, modellgestütze Regelung eines Industrieroboters. *Fortschritte der Robotik*, no. 13, Vieweg-Verlag, Wiesbaden.

Swevers, J., Al-Bender, F., Gansemann, C., and Prajogo, T. 2000. An integrated friction model structure with improved presliding behavior for accurate friction compensation. *IEEE Transactions on Automatic Control* 45(4):675–686.

Swevers, J., Gansemann, C., Tükel, D., Schutter, J. D., and Brussel, H. V. 1997. Optimal robot excitation and identification. *IEEE Transactions on Robotics and Automation* 13(5):730–740.

Wagner, B. 1999. Analytische und iterative Verfahren zur Inversion linearer und nichtlinearer Abtastsystems. *Fortschritt-Berichte VDI, Reihe 8*, Nr. 763, VDI Verlag, Düsseldorf.