

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Model-Based Control for Industrial Robots: Uniform Approaches for Serial and Parallel Structures

Houssem Abdellatif and Bodo Heimann

1. Introduction

Nowadays, there are still two major challenges for industrial robotics in automated production. These are enhancing manufacturing precision and reducing cycle-times. Beside the advances made in the classic robotics over the last decades, new technologies are emerging in the industrial field aiming more flexible high-speed and accurate manufacturing. Robots with parallel structures are attracting the attention of automation industry as an innovative product with high dynamic potentials. Such robots, like the tricept are integrated nowadays by BMW, Volvo or Airbus in their manufacturing lines. Compared to each other, the classic serial (or open) chain robots and the parallel (or closed) chain robots have their specific benefits and suffer from own drawbacks. The proposed chapter gives a comparison of the two types in the scope of their suitability for solving modern problems in industrial robotics. Additionally, appropriate approaches are proposed to remove drawbacks of classic industrial control solutions. Hereby, it is focussed on model-based strategies for ameliorating control accuracy at high dynamics and therefore to expose solutions towards high-speed automation.

One of the main purposes of the proposed chapter is to contribute to extending the state of the art in industrial robotics by the innovative class of parallel robots. Furthermore, classic and advanced model-based control approaches are discussed for both robot types. Uniform methodologies for both classes are given. It is focused on crucial issues for practical application in the industrial field.

The first aspect is surely the modelling of kinematics (see section 2) and dynamics (see section 3) for serial and parallel robots. Here, an opposite duality in formalism is shown. By appropriate choice of minimal coordinates and velocities, the inverse dynamics of the two robot classes can be derived by the principle of virtual power. This yields computational highly efficient models that are well appropriate for real-time applications. Since the success of such feedforward control depends on the estimation quality of the model parame-

ters, appropriate strategies for experimental identification are provided in section 4. Thereby, two main categories of procedures are discussed: direct and indirect identification. The direct procedure tries to estimate model parameters from measurements achieved by one optimized trajectory. Indirect identification uses standard Point-to-Point motions that are distributed within the workspace. The choice of the method in praxis depends on the used control hardware and sensors. Each approach has own advantages and drawbacks for the here discussed two classes of robotic manipulators.

In section 5, further enhancement of control accuracy are demonstrated by providing pre-correction techniques, like iterative learning control, training or nonlinear pre-correction. Such powerful tools are highly appropriate for manufacturing or automation tasks that are repeated over and over. Furthermore, it is advantageous not only due to the simple requirement of standard position-correction interface but because complex modeling of disturbances is not necessary. The methodology is exposed uniformly for serial and parallel robots. Practical issues and some differences are pointed out. Experimental results prove than the suitability and effectiveness of the proposed methods for the studied classes of robots. All proposed approaches are substantiated by experimental results achieved on three different robots: the *Siemens Manutec-r15*, the *KUKA KR15* and the prototype *PaLiDA* as a parallel robot. The chapter is closed with conclusions and an outlook on the possible future of industrial robotics.

2. Kinematic Analysis

To enable giving uniform approaches for serial and parallel robots, elementary assumptions and definitions at the formal level have to be revised. As mentioned in the introduction, we will concentrate on the case of industrial relevant robotic systems, i.e. $n = 6$ -DOF non redundant mechanisms. Both mechanisms are supposed to have n_a actuated joints grouped in the vector \mathbf{q}_a , that defines the actuation space A. Additionally, passive joints are denoted by \mathbf{q}_p . Both vectors can be grouped in the joint vector $\mathbf{q} = [\mathbf{q}_a^T \mathbf{q}_p^T]^T$ that correspond consequently to the joint space Q. The operational or work-space W of an industrial robot is defined by the 6-dimensional pose vector \mathbf{x} containing the position and orientation of the end-effector (EE) with respect to the inertial frame. Let the vector \mathbf{z} now denotes the generalized (or minimal) coordinates, which contains the independent coordinates that are necessary to uniquely describe the system. Its dimension coincides therefore with the number of DOF's (Meirovitch, 1970; Bremer, 1988) and it defines the configuration space C.

Already at this formal level, important differences between serial open-chain robots and parallel closed-chain robots are necessary to consider. For classic industrial robots, the case is quite simple and well known. Such systems do

not have passive joints, the actuated joints correspond to the minimal coordinates, which yields the coincidence of almost all coordinate spaces:

$$q = q_a \Rightarrow Q \equiv A \text{ and } z = q_a = q \Rightarrow C \equiv Q \equiv A.$$

The case of 6-DOF parallel robot is more complicated. The pose vector x defines uniquely the configuration of the system. Besides the robot contains passive joints (Merlet, 2000)

$$z = x \Rightarrow C \equiv W \text{ and } q \neq q_a, q_a \neq z \Rightarrow C \neq Q \neq A$$

Consequently, more transformations have to be considered while operating parallel robots. A more serious issue in industrial praxis is that the configuration of parallel robots can not be directly measured, since only the positions of actuated joints are available. It is then necessary to consider this limitation in control issues. To keep uniform handling of both robotic types, it is recommended to focus on the configuration space defined by z . From this point of view the most important notions of kinematics are revisited in the following. The interested reader may be referred to standard books for deeper insight (Tsai, 1999; Sciavicco & Siciliano, 2000; Angeles, 2003; Merlet, 2000; Khalil & Dombre, 2002)

2.1 Kinematic Transformations

In robotics, the motion of each link is described with respect to one or more frames. It is though necessary to define specifications to transform kinematic quantities (positions, velocities and accelerations). Homogenous transformations are state of the art in robotics. Besides the fixed inertial frame $(KS)_0$ and the end-effector frame $(KS)_E$, each link i is associated with body-fixed frame $(KS)_i$. For efficient formulation and calculation of the homogenous transformations (given by T_i^{i-1}) between two adjacent links $i-1$ and i , it is recommended to use the modified DENAVIT-HARTENBERG-notation (or MDH), that yields unified formalism for open and closed-chain systems (Khalil & Kleinfinger, 1986; Khalil & Dombre, 2002). We obtain:

$$T_i^{i-1} = \begin{bmatrix} R_i^{i-1} & {}_{(i-1)}r_i^{i-1} \\ 000 & 1 \end{bmatrix} = \begin{bmatrix} c_{\vartheta_i} & s_{\vartheta_i} & 0 & a_i \\ s_{\vartheta_i}c_{\alpha_i} & c_{\vartheta_i}c_{\alpha_i} & -s_{\alpha_i} & -d_i s_{\alpha_i} \\ s_{\vartheta_i}s_{\alpha_i} & c_{\vartheta_i}s_{\alpha_i} & c_{\alpha_i} & d_i c_{\alpha_i} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

which is a function of the MDH-parameters ϑ_i , d_i , α_i and a_i (Khalil & Kleinfinger, 1986). The abbreviations s_x and c_x denote $\sin(x)$ and $\cos(x)$ respectively. The matrix \mathbf{R}_i^{i-1} and the vector ${}_{(i-1)}\mathbf{r}_i^{i-1}$ define orientation and position of frame i with respect to frame $i-1$. The kinematics of any kinematic chain gives an analytic determination of the joint variables ϑ_i (for revolute joints) and d_i (for prismatic joints) as well as their time derivatives. The velocity ${}_{(i)}\mathbf{v}_i$ and angular velocity ${}_{(i)}\boldsymbol{\omega}_i$ of each link i and the corresponding accelerations can be calculated recursively by the following equations:

$${}_{(i)}\mathbf{v}_i = {}_{(i)}\mathbf{v}_{i-1} + {}_{(i)}\tilde{\boldsymbol{\omega}}_{i-1} {}_{(i)}\mathbf{r}_i^{i-1} + \mathbf{e}_z \dot{d}_i \quad (2)$$

$${}_{(i)}\dot{\mathbf{v}}_i = {}_{(i)}\dot{\mathbf{v}}_{i-1} + {}_{(i)}\tilde{\dot{\boldsymbol{\omega}}}_{i-1} {}_{(i)}\mathbf{r}_i^{i-1} + {}_{(i)}\tilde{\boldsymbol{\omega}}_{i-1} {}_{(i)}\tilde{\boldsymbol{\omega}}_{i-1} {}_{(i)}\mathbf{r}_i^{i-1} + \ddot{d}_i \mathbf{e}_z + 2\dot{d}_i {}_{(i)}\tilde{\boldsymbol{\omega}}_{i-1} \mathbf{e}_z \quad (3)$$

$${}_{(i)}\boldsymbol{\omega}_i = {}_{(i)}\boldsymbol{\omega}_{i-1} + \mathbf{e}_z \dot{\vartheta}_i \quad (4)$$

$${}_{(i)}\dot{\boldsymbol{\omega}}_i = {}_{(i)}\dot{\boldsymbol{\omega}}_{i-1} + \dot{\vartheta}_i {}_{(i)}\tilde{\boldsymbol{\omega}}_{i-1} \mathbf{e}_z + \ddot{\vartheta}_i \mathbf{e}_z \quad (5)$$

where $\mathbf{e}_z = [0 \ 0 \ 1]^T$. The Tilde-operator ($\tilde{}$) defines the cross product $\tilde{\mathbf{a}}\mathbf{b} = \mathbf{a} \times \mathbf{b}$.

2.2 Direct and Inverse Kinematics

Industrial applications are characterized by being defined in the operational space W , whereas the robot is controlled in the actuation space A . It is therefore necessary to define and to calculate transformations between the two spaces. Calculating the resulting robot poses from given actuator positions correspond to the direct (or forward) kinematic transformation:

$$\mathbf{f}: A \rightarrow W$$

$$\mathbf{q}_a \rightarrow \mathbf{x} = \mathbf{f}(\mathbf{q}_a)$$

Reciprocally, the inverse (or backward) kinematic transformation is used to obtain actuator positions from a given robot pose:

$$\mathbf{g}: W \rightarrow A$$

$$\mathbf{x} \rightarrow \mathbf{q}_a = \mathbf{g}(\mathbf{x})$$

We mentioned above, that only the minimal coordinates describe the system uniquely. Consequently, only the transformations having the argument set be-

ing the configuration space can be computed or given in a closed form. This fact explains, that the solution of the inverse problem is quite simple and available analytically for parallel robots ($C \equiv W$). Whereas the solution of the forward kinematics can be generally obtained only in a numerical way (Tsai, 1999; Merlet, 2000). In contrast, the forward kinematics can be easily obtained for serial-chain robots ($C \equiv A$), whereas the inverse problem is generally cumbersome to solve. As it will be discussed in following sections, such system-inherent properties have an important impact on the practical implementation of control. E.g. the well-known computed-torque feedback approach is not suitable for parallel robots, since the minimal coordinates $\mathbf{z} = \mathbf{x}$ can not be measured.

For both robotic types the pose vector is defined as:

$$\mathbf{x} = [x \ y \ z \ \alpha \ \beta \ \gamma]^T,$$

where the end-effector position being $\mathbf{r}_E = [x \ y \ z]^T$ and its orientation $\boldsymbol{\pi} = [\alpha \ \beta \ \gamma]^T$ being defined according to the Roll-Picth-Yaw (RPY) Euler-convention (Tsai, 1999; Sciavicco & Siciliano, 2000). The homogeneous transformation between $(KS)_0$ and $(KS)_E$ is given by

$$\mathbf{T}_E^0(\mathbf{x}) = \begin{bmatrix} c_\beta c_\gamma & -c_\beta s_\gamma & s_\beta & x \\ c_\alpha s_\gamma + s_\alpha s_\beta c_\gamma & c_\alpha c_\gamma - s_\alpha s_\beta s_\gamma & -s_\alpha c_\beta & y \\ s_\alpha s_\gamma - c_\alpha s_\beta c_\gamma & s_\alpha c_\gamma + c_\alpha s_\beta s_\gamma & c_\alpha c_\beta & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.3 Differential Kinematics

The differential kinematics maps the velocity of the end-effector into the velocity of the actuated joints $\dot{\mathbf{q}}_a$ and vice versa. It is necessary to relate a desired motion in the task-space to the necessary motion of the actuated joints. This is achieved by the jacobian matrix

$$\dot{\mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial q_{a,1}} & \dots & \frac{\partial f_1}{\partial q_{a,n_a}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial q_{a,1}} & \dots & \frac{\partial f_n}{\partial q_{a,n_a}} \end{bmatrix} \dot{\mathbf{q}}_a \quad (7)$$

or simply

$$\dot{\mathbf{x}} = \mathbf{J}_A \dot{\mathbf{q}}_a. \quad (8)$$

The analytical Jacobian \mathbf{J}_A relates the time derivative of the pose vector to the articulated velocities. Since the orientation vector $\boldsymbol{\pi}$ is composed of pseudo-coordinates, whose time derivative has no physical meanings (Bremer, 1988, Meirovitch, 1970) it is convenient to define the rotational velocities of the end-effector in respect to the fixed frame: $\boldsymbol{\omega}_E = [\omega_x \ \omega_y \ \omega_z]^T$, such that

$$\boldsymbol{\omega}_E = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & s_\beta \\ 0 & c_\alpha & -s_\alpha c_\beta \\ 0 & s_\alpha & c_\alpha c_\beta \end{bmatrix}}_{\mathbf{R}_K(\alpha, \beta)} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} \quad (9)$$

and therefore the definition of the geometric jacobian matrix \mathbf{J} :

$$\mathbf{v}_E = \begin{bmatrix} \dot{\mathbf{r}}_E \\ \boldsymbol{\omega}_E \end{bmatrix} = \mathbf{J}_A \dot{\mathbf{q}}_a, \text{ with } \mathbf{J} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_K \end{bmatrix} \mathbf{J}_A \quad (10)$$

By regarding eq. (7) it is obvious that the analytic derivation of the jacobian is only available, when the direct kinematic solution $\mathbf{f}(\mathbf{q}_a)$ is given in a closed form. This is the case for classic open-chain robots, whereas for parallel robots, the inverse jacobian \mathbf{J}^{-1} is available (Merlet, 2000). For such mechanisms, the jacobian is obtained by numerical inversion of its analytically available inverse (Merlet, 2000; Abdellatif et al., 2005a). The mobility of robots depends on the structure of the related jacobian that describes the velocity and also the force transmission between the operational space and the actuation space. It is well known, that singularities occur at configurations, when the jacobian loses its rank ($\det(\mathbf{J}) = 0$). The study of singularities is omitted in this paper. The interested reader may be referred to standard and excellent literature in this area (Gosselin & Angeles, 1990; Sciavicco & Siciliano, 2000; Merlet, 2000; Bonev, 2002).

It is now necessary to define further quantities to describe the motion of robotic manipulators. In analogy to the generalized coordinates, the generalized velocities are introduced (Meirovitch, 1970; Bremer, 1988) and are denoted by $\dot{\mathbf{s}}$. They always present a linear combination of the time-derivatives of the generalized coordinates $\dot{\mathbf{z}}$. The simplest case is when these combinations correspond to the identity:

$$\dot{\mathbf{s}} = \mathbf{I} \dot{\mathbf{z}} \Rightarrow \dot{\mathbf{s}} = \dot{\mathbf{z}}$$

This is the case of classic open-chain robots: $\dot{\mathbf{s}} = \dot{\mathbf{q}}_a$. For parallel manipulators, the end-effector's velocities are chosen to be the generalized coordinates:

$\dot{\mathbf{s}} = \mathbf{v}_E \neq \dot{\mathbf{x}} = \dot{\mathbf{z}}$. This formal property has also an important impact in the practice. The symbolic derivation of the Lagrangian equations of motions becomes very messy for parallel robots, such that its implementation in real-time control systems is very restrictive (Tsai, 1999).

The last fundamental step of our revised kinematic analysis is the definition of limb's jacobians \mathbf{J}_{T_i} and \mathbf{J}_{R_i} that relate its translational and its angular velocities to the generalized velocities of the robot, respectively:

$$\mathbf{J}_{T_i} = \frac{\partial {}^{(i)}\mathbf{v}_i}{\partial \dot{\mathbf{s}}} \quad \text{and} \quad \mathbf{J}_{R_i} = \frac{\partial {}^{(i)}\boldsymbol{\omega}_i}{\partial \dot{\mathbf{s}}}$$

The use of the modified DENAVIT-HARTENBERG-notation allows also a recursive calculation of the limb's jacobians:

$$\mathbf{J}_{T_i} = \mathbf{R}_{i-1}^i \left(\mathbf{J}_{T_{i-1}} - {}^{(i-1)}\tilde{\mathbf{r}}_i^{i-1} \mathbf{J}_{R_{i-1}} \right) + \mathbf{e}_z \frac{\partial \dot{d}_i}{\partial \dot{\mathbf{s}}} \quad (11)$$

$$\mathbf{J}_{R_i} = \mathbf{R}_{i-1}^i \mathbf{J}_{R_{i-1}} + \mathbf{e}_z \frac{\partial \dot{\vartheta}_i}{\partial \dot{\mathbf{s}}} \quad (12)$$

The next subsection demonstrates the efficiency and uniformity of the proposed method for deriving the kinematics of a serial and a parallel industrial robot.

2.4 Application of the Kinematic Analysis of Industrial Robots

2.4.1 Serial Manipulators: Case Study KUKA KR15

The direct kinematics of serial-chain robots is straight forward. The transformation matrix can be calculated by starting from the base and evaluating the single \mathbf{T}_i^{i-1} . By solving

$$\mathbf{T}_E^0(\mathbf{q}_a) = \prod_{i=0}^n \mathbf{T}_i^{i-1}(q_{a,i}) = \mathbf{T}_E^0(\mathbf{x}),$$

we obtain the pose vector \mathbf{x} . The jacobian is also joint-wise simple to obtain:

$$\mathcal{J}(\mathbf{q}) = [\mathbf{J}_1 \mid \mathbf{J}_2 \mid \dots \mid \mathbf{J}_n] \text{ with } \mathbf{J}_i = \begin{bmatrix} {}^{(0)}\mathbf{e}_z^i \times {}^{(0)}\mathbf{x}_{(0)}^i \mathbf{r}_E^i \\ {}^{(0)}\mathbf{e}_z^i \end{bmatrix} \quad (13)$$

This can be deduced by using the MDH-notation and the recursive formulae given above. Although the solution of the inverse kinematics is generally hard to obtain for open-chain mechanisms, industrial robots are characterized by simple geometry, such that a closed-form solution exists. This is the case here, where the three last revolute joint axes intersect at a common point (spherical wrist) (Sciavicco & Siciliano, 2000).

2.4.2 Parallel Manipulators: Case Study PaLiDA

The general method of calculating the inverse kinematics of parallel robots is given by splitting the system into a set of subchains. The structure is opened and separated into "legs" and an end-effector-platform. Hereby the enclosure constraints have to be calculated, which are the vectors connecting A_j with B_j

$$\mathbf{r}_{B_j}^{A_j} = \begin{bmatrix} x_j & y_j & z_j \end{bmatrix}^T = -\mathbf{r}_{A_j}^0 + \mathbf{r}_E^0 + \mathbf{R}_{E(E)}^0 \mathbf{r}_{B_j}^E. \quad (14)$$

Thus, every chain can now be regarded separately as a conventional open-chain robot with a corresponding end-effector position at $\mathbf{r}_{B_j}^{A_j}$. MDH-Parameters are defined for each subchain and the direct kinematics is solved as described above. Since we consider non-redundant mechanisms, the resulting serial chains are very simple, such that a closed form solution always exists. For the studied case *PaLiDA*, the definition of the MDH-parameters and frames are depicted in Figure 2. The solution of the full inverse kinematics is obtained by

$$q_{a_j} = l_j = \sqrt{x_j^2 + y_j^2 + z_j^2} \quad (15)$$

$$\alpha_j = \arctan\left(\frac{x_j}{-z_j}\right) \quad (16)$$

$$\beta_j = \arctan\left(\frac{y_j}{r_j}\right), \quad (17)$$

which are quite simple equations. The differential kinematics can be deduced analytically for the inverse problem by the inverse jacobian:

$$\mathbf{J}(\mathbf{x})^{-1} = \frac{\partial \dot{\mathbf{q}}_a}{\partial \dot{\mathbf{s}}} = \begin{bmatrix} \frac{\partial \dot{\mathbf{q}}_a}{\partial \dot{\mathbf{r}}_E} & \frac{\partial \dot{\mathbf{q}}_a}{\partial \dot{\boldsymbol{\omega}}_E} \end{bmatrix} \quad (18)$$

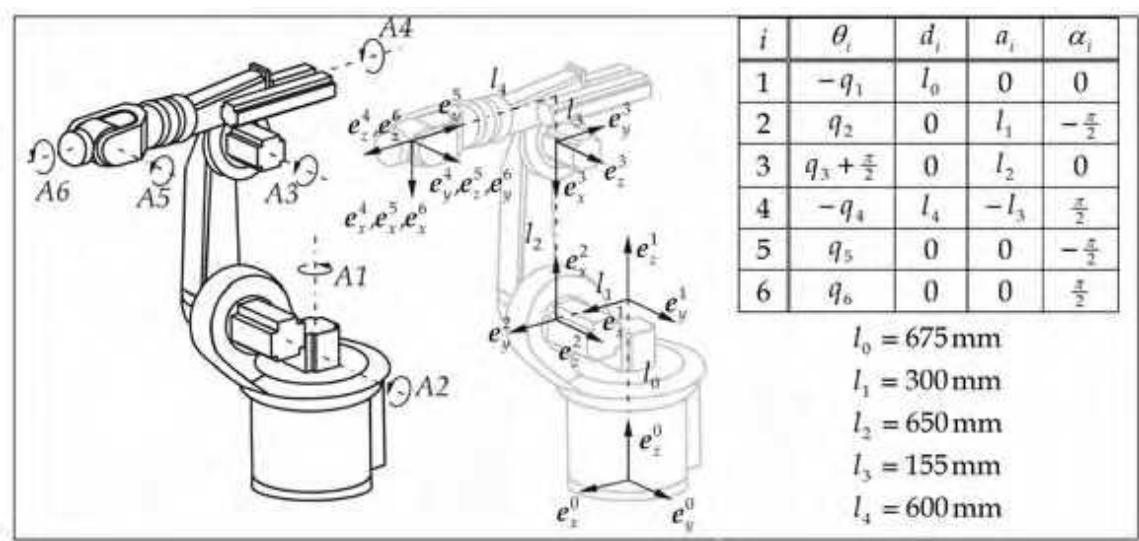


Figure 1. Definition of the MDH Coordinate systems and parameter for the KUKA KR 15

Many methods are proposed in the literature for calculating the inverse jacobian. We propose here the most straight-forward way in our case. Every single chain j corresponds to the j^{th} row of the inverse jacobian:

$$\mathbf{J}_j^1 = \frac{\partial \dot{q}_{a_j}}{\partial \mathbf{v}_{B_j}} \quad \frac{\partial \mathbf{v}_{B_j}}{\partial \dot{\mathbf{s}}} = \frac{\partial \dot{q}_{a_j}}{\partial \mathbf{v}_{B_j}} \begin{bmatrix} \frac{\partial \mathbf{v}_{B_j}}{\partial \dot{\mathbf{r}}_E} & \frac{\partial \mathbf{v}_{B_j}}{\partial \dot{\boldsymbol{\omega}}_E} \end{bmatrix} = \frac{\partial \dot{q}_{a_j}}{\partial \mathbf{v}_{B_j}} [\mathbf{I} - \tilde{\mathbf{r}}_{B_j}^E] \tag{19}$$

The velocities of the points B_j can be obtained by simply differentiating the constraint equation (14):

$$\mathbf{v}_{B_j} = \dot{\mathbf{r}}_E + \tilde{\boldsymbol{\omega}}_E \mathbf{r}_{B_j}^E = \dot{\mathbf{r}}_R + \tilde{\mathbf{r}}_{B_j}^E \boldsymbol{\omega}_E \tag{20}$$

By using the recursive laws given by eq. (3-5) the complete inverse kinematics of the subchains can be solved, yielding velocities and accelerations of each limb and moreover a functional relationship between q_{a_j} and \mathbf{v}_{B_j} that is needed for (19).

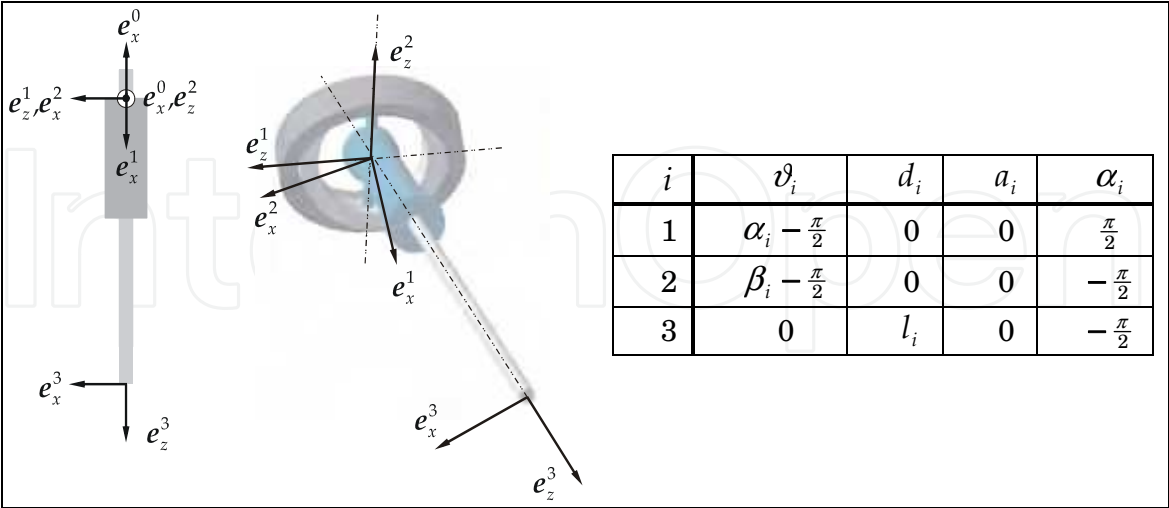


Figure 2. Definition of the MDH-parameters for a serial subchain of the hexapod *PaLiDA*

As conclusions, we can retain that the formal differences between parallel and serial robots have to be taken into account. A unified approach can be given if the notions of minimal coordinates and velocities are kept in mind. The MDH-notation provide the same procedure when solving kinematics for both robotic types. For parallel robots it is sufficient to formulate the constraint equations. Hereafter the mechanism is separated into serial subchains that can be treated exactly as any other open-chain manipulator.

3. Efficient Formulation of Inverse Dynamics

Model-based and feedforward control in industrial robotics requires computational efficient calculation of the inverse dynamics, to fulfill real-time requirements of standard control systems. The real-time calculation of the desired actuator forces \mathbf{Q}_a depends on the used approach for the derivation of the inverse Model. For the sake of clarity we concentrate first on rigid-body dynamics. The corresponding equations of motions for any manipulator type can be derived in the following four forms:

$$\mathbf{Q}_a = \mathbf{B}_a(\mathbf{z}, \dot{\mathbf{s}}, \ddot{\mathbf{s}})$$
 (21)

$$\mathbf{Q}_a = \mathbf{M}_a(\mathbf{z})\ddot{\mathbf{q}}_a + \mathbf{N}(\mathbf{z}, \dot{\mathbf{s}})$$
 (22)

$$\mathbf{Q}_a = \mathbf{M}_a(\mathbf{z})\ddot{\mathbf{q}}_a + \mathbf{c}_a(\mathbf{z}, \dot{\mathbf{s}}) + \mathbf{g}_a(\mathbf{z})$$
 (23)

$$\mathbf{Q}_a = \mathbf{A}_a(\mathbf{z}, \dot{\mathbf{s}}, \ddot{\mathbf{s}})\mathbf{p}_{\min}$$
 (24)

where \mathbf{Q}_a being the vector of actuation forces. The massmatrix is denoted by \mathbf{M} . The vectors \mathbf{c} and \mathbf{g} contain the centrifugal and Coriolis, and the gravitational terms, respectively. The vector \mathbf{N} includes implicitly \mathbf{c} and \mathbf{g} . Analogically, the vector $\mathbf{B}(\mathbf{z}, \dot{\mathbf{s}}, \ddot{\mathbf{s}})$ includes implicitly all terms of rigid-body dynamics. We notice here, that the index 'a' is used to distinguish the quantities that are related to the actuation space. A trivial but very important remark is that all model forms have in common, that the inputs are always given in the configuration space by \mathbf{z} , $\dot{\mathbf{s}}$ and $\ddot{\mathbf{s}}$, whereas the outputs are always given in the actuation space: \mathbf{Q}_a . Although, equations (21-24) yield exactly the same results, they are very different to derive and to calculate. Although eq. (23) is the most computational intensive form, it is very reputed in robotics because it is highly useful for control design and planning. The case of open-chain manipulators is easier. The coincidence of configuration space with the actuation space allows a straight-forward implementation of the Lagrangian formalism for its derivation. This is not the case for the parallel counterpart, where the same formalism leads to messy symbolic computation or in the worst case to non-closed form solution (Tsai, 1999). Therefore, we focus in the following on the most efficient¹ form (21) that can be derived uniformly for parallel and serial robots.

3.1 Derivation of the Rigid-Body Dynamics

The suggested approach is the Jourdainian principle of virtual power that postulates power equality balances with respect to the forces in different coordinate spaces (Bremer, 1988). For instance, a power balance equation is obtained as

$$\partial \dot{\mathbf{s}}^T \boldsymbol{\tau} = \partial \dot{\mathbf{q}}_a^T \mathbf{Q}_a \Leftrightarrow \boldsymbol{\tau} = \left(\frac{\partial \dot{\mathbf{q}}_a}{\partial \dot{\mathbf{s}}} \right)^T \mathbf{Q}_a \quad (25)$$

where $\boldsymbol{\tau}$ is the vector of the generalized forces. Equation (25) means that the virtual power resulting in the space of generalized velocities is equal to the actuation power. The power balance can be applied for rigid-body forces:

$$\mathbf{Q}_{a,rb} = \left(\frac{\partial \dot{\mathbf{q}}_a}{\partial \dot{\mathbf{s}}} \right)^{-T} \boldsymbol{\tau}_{rb} \quad (26)$$

¹ Parameterlinear equations of motions (24) are actually more computational efficient. Since they are derived from (21), they are discussed later on.

The generalized forces are computed as the summation of the power of all N_K limbs:

$$\boldsymbol{\tau}_{\text{rb}} = \sum_{i=1}^{N_K} \left[\left(\frac{\partial \mathbf{v}_{S_i}}{\partial \dot{\mathbf{s}}} \right)^T m_i \mathbf{a}_{S_i} + \left(\frac{\partial \boldsymbol{\omega}_i}{\partial \dot{\mathbf{s}}} \right)^T \left(\mathbf{I}_i^{(S_i)} \dot{\boldsymbol{\omega}}_i + \tilde{\boldsymbol{\omega}}_i \left(\mathbf{I}_i^{(S_i)} \boldsymbol{\omega}_i \right) \right) \right] \quad (27)$$

with $\mathbf{a}_{S_i} = \dot{\mathbf{v}}_{S_i} - \mathbf{g}$ being the absolute acceleration of the i^{th} link's center of gravity S_i . The velocity of the center of gravity, the mass and the inertia-tensor with respect to S_i are denoted by $\dot{\mathbf{v}}_{S_i}$, m_i and $\mathbf{I}_i^{(S_i)}$, respectively. To be able of using the recursion calculation of kinematic quantities (2-5, 11), eq. (27) is transformed to

$$\boldsymbol{\tau} = \sum_{i=1}^{N_K} \left[\underbrace{\left(\frac{\partial {}_{(i)}\mathbf{v}_i}{\partial \dot{\mathbf{s}}} \right)^T}_{\mathbf{J}_{R_i}^T} \left(m_{i(i)} \mathbf{a}_i + {}_{(i)}\dot{\tilde{\boldsymbol{\omega}}}_i \mathbf{s}_i + {}_{(i)}\tilde{\boldsymbol{\omega}}_i {}_{(i)}\tilde{\boldsymbol{\omega}}_i \mathbf{s}_i \right) + \underbrace{\left(\frac{\partial {}_{(i)}\boldsymbol{\omega}_i}{\partial \dot{\mathbf{s}}} \right)^T}_{\mathbf{J}_{R_i}^T} \left({}_{(i)}\mathbf{I}_i^{(i)} \dot{\boldsymbol{\omega}}_i + {}_{(i)}\tilde{\boldsymbol{\omega}}_i \left({}_{(i)}\mathbf{I}_i^{(i)} \boldsymbol{\omega}_i \right) + {}_{(i)}\tilde{\mathbf{s}}_i \mathbf{a}_i \right) \right] \quad (28)$$

with \mathbf{s}_i being the vector of the i^{th} body's first moment² $\mathbf{s}_i = \begin{bmatrix} s_{i_x} & s_{i_y} & s_{i_z} \end{bmatrix}^T = m_{i(i)} \mathbf{r}_{S_i}^i$ ($\mathbf{r}_{S_i}^i$: location of S_i with respect to the limb-fixed coordinate frame) and ${}_{(i)}\mathbf{I}_i^{(i)}$ being the inertia tensor about the same coordinate frame.

It is obvious, that the calculation of $\boldsymbol{\tau}_{\text{rb}}$ requires the quantities of motions of all bodies. The latter can be obtained by applying the kinematic analysis as already explained in the former section 2. The transformation of the generalized forces into the actuation space according to (2) is trivial for the case of serial robots ($\dot{\mathbf{s}} \equiv \dot{\mathbf{q}}_a$)

$$\left(\frac{\partial \dot{\mathbf{q}}_a}{\partial \dot{\mathbf{s}}} \right)^{-T} = \left(\frac{\partial \dot{\mathbf{q}}_a}{\partial \dot{\mathbf{q}}_a} \right)^{-T} = \mathbf{I}$$

² not to confuse with generalized velocities $\dot{\mathbf{s}}$

For parallel manipulators, the numerical calculation of the jacobian is necessary (see also section 2.3):

$$\mathbf{Q}_{a,rb} = \left(\frac{\partial \dot{\mathbf{q}}_a}{\partial \dot{\mathbf{s}}} \right)^{-T} \boldsymbol{\tau}_{rb} = \mathbf{J}^T \boldsymbol{\tau}_{rb}$$

The inverse dynamics presented by (28) is already highly computational efficient. It can be implemented in real-time within nowadays standard control systems for parallel as well as for serial ones. Such model can be further optimized and transformed into a linear form with respect to the minimal dynamic parameters \mathbf{p}_{\min} .

3.2 Minimalparameter Form of the Equations of Motion

By transforming the dynamics into form (24), two main advantages result. At one hand, regrouping the parameters will further reduce the calculation burden and at the other hand, one obtains the set of identifiable parameters of the robotic system. We focus now on the dynamic parameters presented by m_i , \mathbf{s}_i and \mathbf{I}_i . To regroup such parameters, the definition of two new operators $(\cdot)^*$ and $(\cdot)^\diamond$ are required first:

$$\boldsymbol{\omega}_i^* \mathbf{I}_i^\diamond := {}_{(i)}\mathbf{I}_i^{(i)} \boldsymbol{\omega}_i \quad (29)$$

$$\text{with } \boldsymbol{\omega}_i^* := \begin{bmatrix} \omega_{i_x} & \omega_{i_y} & \omega_{i_z} & 0 & 0 & 0 \\ 0 & \omega_{i_x} & 0 & \omega_{i_y} & \omega_{i_z} & 0 \\ 0 & 0 & \omega_{i_x} & 0 & \omega_{i_y} & \omega_{i_z} \end{bmatrix} \text{ and } \mathbf{I}_i^\diamond = \begin{bmatrix} I_{i_{xx}} & I_{i_{xy}} & I_{i_{xz}} & I_{i_{yy}} & I_{i_{yz}} & I_{i_{zz}} \end{bmatrix}^T$$

The inverse dynamics (28) can be written as

$$\begin{aligned} \boldsymbol{\tau}_{rb} &= \sum_{i=1}^{N_K} \left[\mathbf{J}_{T_i}^T \quad \mathbf{J}_{R_i}^T \right] \underbrace{\begin{bmatrix} \mathbf{0} & {}_{(i)}\dot{\tilde{\boldsymbol{\omega}}} + {}_{(i)}\tilde{\boldsymbol{\omega}} & {}_{(i)}\tilde{\boldsymbol{\omega}} & {}_{(i)}\mathbf{a}_i \\ {}_{(i)}\dot{\boldsymbol{\omega}}^* + {}_{(i)}\tilde{\boldsymbol{\omega}} & \boldsymbol{\omega}_i^* & -{}_{(i)}\tilde{\mathbf{a}}_i & \mathbf{0} \end{bmatrix}}_{\mathbf{H}_i} \underbrace{\begin{bmatrix} \mathbf{I}_i^\diamond \\ \mathbf{s}_i \\ m_i \end{bmatrix}}_{\mathbf{p}_{rb,i}} \\ &= \underbrace{\begin{bmatrix} \mathbf{H}_1 & \mathbf{H}_2 & \dots & \mathbf{H}_{N_K} \end{bmatrix}}_{\mathbf{H}(\mathbf{z}, \dot{\mathbf{s}}, \ddot{\mathbf{s}})} \underbrace{\begin{bmatrix} \mathbf{p}_1^T & \mathbf{p}_2^T & \dots & \mathbf{p}_{N_K}^T \end{bmatrix}^T}_{\mathbf{p}_{rb}} \end{aligned} \quad (30)$$

which is now linear in respect to the parameter set \mathbf{p}_{rb} , that groups all physical parameters of all limbs of the robot. Since each limb contributes with 1 mass parameter, 3 first-moment parameters and 6 inertia tensor elements, we obtain

for the robot the number of $(1 + 3 + 6) \times N_K$ physical parameters. The contribution of each single parameter to the dynamics is presented by the corresponding column of the matrix \mathbf{H}_i . The dimension of \mathbf{p}_{rb} has to be reduced for more computational efficiency and identifiability of the dynamics model. In the field of robotics, many researches have been achieved on this subject, especially for serial robots (Khalil & Kleinfinger, 1987; Gauthier & Khalil, 1990; Fiset et al., 1996). Recently the problem was also addressed for parallel mechanisms (Khalil & Guegan, 2004; Abdellatif et al., 2005a). Generally, the procedure consists in a first step of grouping the parameters for the open chains. Afterwards, one looks for further parameter reduction that is due to eventually existing closed kinematic loops. In Praxis, the first step is common for serial and parallel robots. For the latter, the structure is subdivided in single serial chains. The second step is achieved of course, only for parallel robots.

The matrices \mathbf{H}_i in (30) can be grouped for single serial kinematic chains, such that a recursive calculation:

$$\mathbf{H}_i = \mathbf{H}_{i-1} \mathbf{L}_i + \mathbf{K}_i \quad (31)$$

can be achieved. The matrices \mathbf{L}_i and \mathbf{K}_i are given in (Khalil & Dombre, 2002; Grotjahn & Heimann, 2000). The first step considers in eliminating all parameters $p_{rb,j}$ that correspond to a zero row \mathbf{h}_j of \mathbf{H} , since they do not contribute to the dynamics. The remaining parameters are then regrouped to eliminate all linear dependencies by investigating \mathbf{H} . If the contribution of a parameter $p_{rb,j}$ depends linearly on the contributions of some other parameters $p_{rb,1j}, \dots, p_{rb,kj}^*$, the following equation holds

$$\mathbf{h}_j = \sum_{l=1}^k a_{lj} \mathbf{h}_{lj} \quad (32)$$

Then $p_{rb,j}$ can be set to zero and the regrouped parameters $p_{rb,lj,new}$ can be obtained by

$$p_{rb,lj,new} = p_{rb,lj} + a_{lj} p_{rb,j}^* \quad (33)$$

The recursive relationship given in (31) can be used for parameter reduction. If one column or a linear combination of columns of \mathbf{L}_i is constant with respect to the joint variable and the corresponding columns of \mathbf{K}_i are zero columns, the parameters can be regrouped. This leads to the rules which are formulated in (Gauthier & Khalil, 1990; Khalil & Dombre, 2002) and in (Grotjahn &

Heimann, 2000). The use of MDH-notation is a benefit for applying the reduction rule in an analytical and a straight-forward manner. For revolute joints with variable ϑ_i , the other MDH-parameters are constant. This means that the 9th, the 10th and the sum of the 1st and 4th columns of \mathbf{L}_i and \mathbf{K}_i comply with the mentioned conditions. Thus, the corresponding parameters $I_{i_{yy}}$, s_{i_x} and m_i can be grouped with the parameters of the antecedent joint $i-1$. For prismatic joints however, the moments of inertia can be added to the carrying antecedent joint, because the orientation between both links remain constant. For a detailed insight, it is recommended to consider (Khalil & Dombre, 2002) and (Grotjahn & Heimann, 2000).

In the case of parallel robots, where the end-effector platform closes the kinematic loops, further parameter reduction is possible. The velocities of the platform joint points \mathbf{B}_j and those of the terminal MDH-frames of the respective leg are the same. The masses can be grouped to the inertial parameter of the EE-platform according to steiner's laws (Khalil & Guegan, 2004; Abdellatif et al., 2005a).

3.3 Integration of friction and motor inertia effects

For further accuracy enhancement of the inverse dynamics models, the effects of friction and motor inertia should be considered. Especially the first category is important for control applications (Grotjahn & Heimann, 2002; Armstrong-Hélouvry, 1991; Bona & Indri, 2005). The general case is considered, when friction is modeled in all active as well as in passive joints. The friction is given in the joint space \mathbf{Q} , usually as nonlinear characteristics $\mathbf{Q}_{f_i}(\dot{q}_i) = f(\dot{q}_i)$ with respect to the joint velocity, i.e.

$$\mathbf{Q}_{f_i}(\dot{q}_i) = r_{1_i} \text{sign}(\dot{q}_i) + r_{2_i} \dot{q}_i \quad (34)$$

The joint losses have to be mapped into the actuation (or control) space. Uniformly to the rigid-body dynamics, the Jourdainian principle of virtual power is recommended. The power balance for friction can be derived as

$$\mathbf{Q}_{a,f} = \left(\frac{\partial \dot{\mathbf{q}}}{\partial \dot{\mathbf{q}}_a} \right)^T \mathbf{Q}_f = \mathbf{J}^T \left(\frac{\partial \dot{\mathbf{q}}}{\partial \dot{\mathbf{s}}} \right)^T \mathbf{Q}_f \quad (35)$$

that means: the friction dissipation power in all joints (passive and active) has to be overcome by an equivalent counteracting actuation power. From the latter equation it is clear that the case of classic open-chain robots is restrictive, when the joint-jacobian $\partial \dot{\mathbf{q}} / \partial \dot{\mathbf{q}}_a$ is equal to the identity matrix. In the more general case of parallel mechanisms, friction in passive joints should not be

neglected like it is almost always assumed in control application for such robots (Ting et al., 2004; Cheng et al., 2003). The compensation of friction is simpler and more accurate for serial robots, since it can be achieved directly in all actuated joints. For the parallel counterpart, the compensation of the physical friction \mathbf{Q}_f is only possible indirectly via the projected forces $\mathbf{Q}_{a,f}$ to account for passive joints. Since the latter are usually not equipped with any sensors, friction compensation in parallel robots is less accurate, which is one of the typical drawbacks of such robotic systems.

By using friction models that are linear with respect to the friction coefficients, like (34) it is more or less easy to derive a linear form of (36). The following relationship results:

$$\mathbf{Q}_{a,f} = \mathbf{A}_f(\mathbf{z}, \dot{\mathbf{s}}) \mathbf{p}_f \quad (36)$$

where the friction coefficients are grouped in a corresponding parameter vector \mathbf{p}_f .

The inertial effects of drives and gears can be also considered and integrated in the dynamics with standard procedures like described in (Sciavicco & Siciliano, 2000; Khalil & Dombre, 2002). One of the advantages provided by parallel robots is the fact, that the motors are mainly installed on the fixed platform, such that they do not contribute to the dynamics. This issue remains - at least for industrial application – exclusive for conventional serial manipulators, where the motors are mounted on the respective limbs.

3.4 Example: Minimal rigid-body parameters

The illustrative example of minimal rigid-body parameters is considered to give an interesting comparison between serial and parallel manipulators in terms of dynamics modeling. The above described uniform approach is applied for the 6-DOF robots *KUKA KR15* and *PaLiDA*. According to the notations defined in the former section, the minimal parameters are derived for both systems. The results are illustrated in Table 1. Despite higher structural complexity, the minimal parameters of the parallel robot are less numerous and complex than those of the serial one. The single sub-chains of a parallel robot are usually identical and have simple structure, which yields identical and simple-structured parameters for the different chains. The kinematic coupling yields a further parameter reduction as the example demonstrates for $p_6 - p_{10}$. The inertial effects of the limbs directly connected to the moving platform are counted to the dynamics of the end-effector by taking ${}^{(E)}\mathbf{r}_{\mathbf{B}_j}^E = [r_{Bx_j} \quad r_{By_j} \quad r_{Bz_j}]^T$ into account (see also eq. (14)). The derivation of minimal parameters is of a major interest, since they constitute the set of identifi-

able ones (Gautier & Khalil, 1990). Following section discusses the experimental identification of parameters and the implementation of identified inverse models in control.

	<i>KUKA KR15</i>	<i>PaLiDA</i>
p_1	$I_{M_1} + I_{1_{zz}} + I_{2_{yy}} + I_{3_{yy}} + l_1^2(m_2 + m_3) + m_3 l_2^2 + (m_4 + m_5 + m_6)(l_1^2 + l_2^2 + l_3^2)$	$I_{1_{zz}} + I_{2_{yy}} + I_{3_{zz}}$
p_2	$I_{2_{xx}} - I_{2_{yy}} - l_2^2(m_3 + m_5 + m_6)$	$I_{2_{xx}} + I_{3_{xx}} - I_{2_{yy}} - I_{3_{zz}}$
p_3	$I_{2_{xz}}$	$I_{2_{zz}} + I_{3_{yy}}$
p_4	$I_{M_2} + I_{2_{zz}} + l_2^2(m_3 + m_4 + m_5 + m_6)$	s_{2_y}
p_5	$s_{2_x} + l_2(m_3 + m_4 + m_5 + m_6)$	s_{3_z}
p_6	$I_{3_{xx}} - I_{3_{yy}} + I_{4_{yy}} + 2l_4 s_{4_z} + (l_4^2 - l_3^2)(m_4 + m_5 + m_6)$	$I_{xx_E} + m_3 \sum_{j=1}^6 (r_{By_j}^2 + r_{Bz_j}^2)$
p_7	$I_{3_{xy}} - l_3 s_{4_z} - l_3 l_4 (m_4 + m_5 + m_6)$	$I_{yy_E} + m_3 \sum_{j=1}^6 (r_{Bx_j}^2 + r_{Bz_j}^2)$
p_8	$I_{3_{xx}} - I_{3_{yy}} + I_{4_{yy}} + 2l_4 s_{4_z} + (l_3^2 + l_4^2)(m_4 + m_5 + m_6)$	$I_{zz_E} + m_3 \sum_{j=1}^6 (r_{Bx_j}^2 + r_{By_j}^2)$
p_9	$s_{3_x} - l_3(m_4 + m_5 + m_6)$	$s_{z_E} + m_3 \sum_{j=1}^6 r_{Bz_j}$
p_{10}	$s_{3_y} - s_{4_z} - l_4(m_4 + m_5 + m_6)$	$m_E + 6m_3$
p_{11}	$I_{4_{xx}} - I_{4_{yy}} + I_{5_{yy}}$	—
p_{12}	$I_{4_{zz}} + I_{5_{yy}}$	—
p_{13}	s_{4_y}	—
p_{14}	$I_{5_{xx}} - I_{5_{yy}} + I_{6_{xx}}$	—
p_{15}	$I_{5_{zz}} + I_{6_{xx}}$	—
p_{16}	$s_{5_y} - s_{6_z}$	—
p_{17}	$I_{6_{zz}}$	—
p_{18}	I_{M_3}	—
p_{19}	I_{M_4}	—
p_{20}	I_{M_5}	—
p_{21}	I_{M_6}	—

Table 1. Minimal rigid-body parameter set for the 6-DOF robots *KUKA KR15* and *PaLiDA*.

4. Dynamics Model Identification

A main result retained from the last section is that the inverse dynamics of robotic manipulators can be written as

$$\mathbf{Q}_a = \mathbf{Q}_{a,rb} + \mathbf{Q}_{a,f} = \mathbf{A}_{rb}(\mathbf{z}, \dot{\mathbf{s}}, \ddot{\mathbf{s}})\mathbf{p}_{rb} + \mathbf{A}_f(\mathbf{z}, \dot{\mathbf{s}})\mathbf{p}_f = \mathbf{A}(\mathbf{z}, \dot{\mathbf{s}}, \ddot{\mathbf{s}})\mathbf{p}$$

with the vector \mathbf{p} containing the minimal rigid-body parameters and friction coefficients. With such an LP (*linear in the parameter*)-structure computational efficient linear estimators can be applied for identification. The formulation of the related problem is derived for an experiment producing N data vectors as follows

$$\underbrace{\begin{bmatrix} \mathbf{Q}_{a_1} \\ \vdots \\ \mathbf{Q}_{a_N} \end{bmatrix}}_{\boldsymbol{\Gamma}} = \underbrace{\begin{bmatrix} \mathbf{A}(\mathbf{z}_1, \dot{\mathbf{s}}_1, \ddot{\mathbf{s}}_1) \\ \vdots \\ \mathbf{A}(\mathbf{z}_N, \dot{\mathbf{s}}_N, \ddot{\mathbf{s}}_N) \end{bmatrix}}_{\boldsymbol{\Psi}} \mathbf{p} + \underbrace{\begin{bmatrix} \mathbf{e}_1 \\ \vdots \\ \mathbf{e}_N \end{bmatrix}}_{\boldsymbol{\eta}} \quad (37)$$

with the measurement vector $\boldsymbol{\Gamma}$, the information or regression matrix $\boldsymbol{\Psi}$ and the error vector $\boldsymbol{\eta}$ that accounts for disturbances. The solution of the over-determined equation system (37) yields Weighted Least-Squares estimate $\hat{\mathbf{p}}_{\text{WLS}}$ of the parameter vector (Walter & Pronzato, 1997)

$$\hat{\mathbf{p}}_{\text{WLS}} = (\boldsymbol{\Psi}^T \mathbf{W}^{-1} \boldsymbol{\Psi})^{-1} \boldsymbol{\Psi}^T \mathbf{W}^{-1} \boldsymbol{\Gamma} \quad (38)$$

where \mathbf{W} is a weight matrix. The classical Least-Squares (LS) estimator results from (38) by setting \mathbf{W} to a diagonal matrix with equal entries

$$\hat{\mathbf{p}}_{\text{LS}} = (\boldsymbol{\Psi}^T \boldsymbol{\Psi})^{-1} \boldsymbol{\Psi}^T \boldsymbol{\Gamma} \quad (39)$$

The quality of the estimation results depends on the so called experiment design that define how and which data has to be collected to guarantee good estimate. Here, two main categories exist: the direct and indirect approach. The first one suggests collecting the data along one single trajectory that excite all parameters. The second approach proposes collecting different measurements in single configurations within the workspace. Each approach has characteristic advantages and drawbacks that depend also on the regarded robot type. It can be stated generally, that the identification procedure for parallel robots is more difficult, because the necessary information about the minimal coordinates, velocities and accelerations can not be directly measured (Abdellatif et al., 2005c). A systematic comparison for both approaches is given in Table 2 as

well as an evaluation of its appropriateness for the studied manipulator types. The main drawback of indirect approaches is the time-consuming data collection procedure, since many measurements are necessary. Moreover, they require high quality measurements to enable handling narrow measurement intervals, otherwise the data quality will be poor (Grotjahn et al., 2004). These drawbacks are eliminated by direct approaches, since we obtain quick and fast identification trajectories (Abdellatif et al., 2005c). The availability of analytical form of the trajectories helps a highly efficient frequency-domain handling of the measurements. On the other hand, optimized trajectories are needed (Swevers et al., 1997) that can not be simply achieved by conventional industrial control setups. Additional programming or hardware interfaces are necessary. This is not the case with the indirect approach because simple PTP-motions with trapezoidal velocity profile are used, which can be directly programmed and executed. Furthermore, the indirect approach enables identification of submodels independently on each other, i.e. friction and rigid-body or inertial and gravitational parameters (Grotjahn et al., 2001; Grotjahn et al., 2004).

	direct approach		indirect approach	
	serial	parallel	serial	parallel
time cost	+	+	–	–
signal processing	+	+	0	–
interface requirement	–	–	+	+
sub-model identification	–	–	+	+
Refs	Swevers et al., 1997	Abdellatif et al., 2005c	Grotjahn et al., 2001	Grotjahn et al., 2004

Table 2. Direct vs. indirect identification approach: appropriateness for serial and parallel industrial robots

The recommended approach depends on the real setup and equipment. If the available control system allows the achievement of arbitrarily complex trajectories, it is recommended to measure optimized trajectories. Otherwise, long measuring processes have to be taken into account, if the indirect approach is chosen. E.g. the identification of friction models for the *KUKA KR15* and for *PaLiDA* required 45 min and 120 min measurement time, respectively. The validation of the identified dynamics models is achieved by considering reference trajectories that were not used for identification. For the industrial *KUKA KR15* robot, the *ISO-92833* standard trajectory is validated. The measured torques are compared with those calculated by the identified model. The results are illustrated in Figure 3.

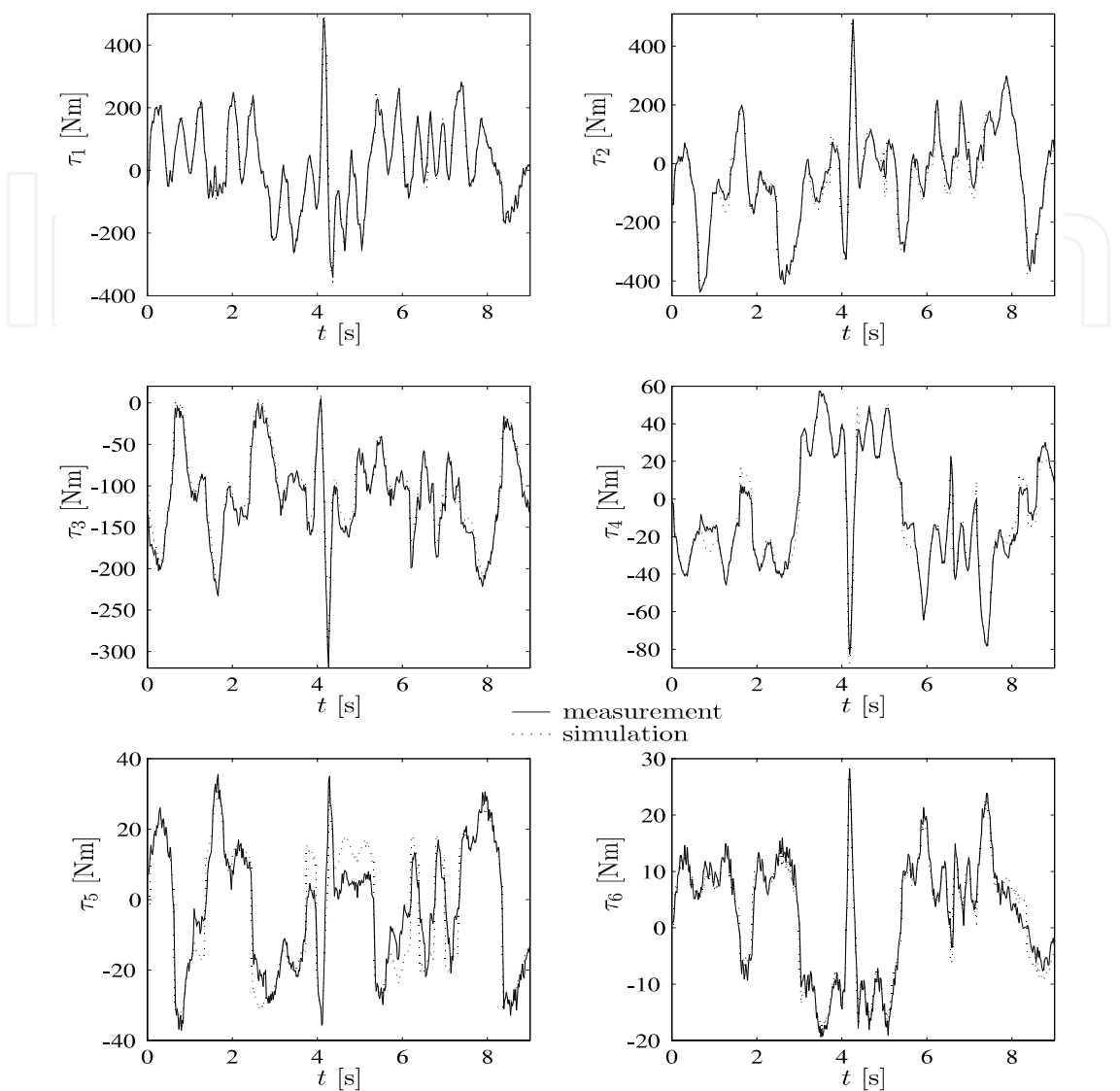


Figure 3. Accuracy validation of identified model of the KUKA KR15: Torques for ISO-9283 trajectory

Unfortunately, standard benchmarks are not defined for parallel robots yet and the ISO-92833 violates the workspace of the here studied robot. Thus, an inclined circular motion in the middle of the workspace is chosen as a benchmark test for validating the accuracy of the identified model for *PaLiDA*. The results are shown in Figure 4.

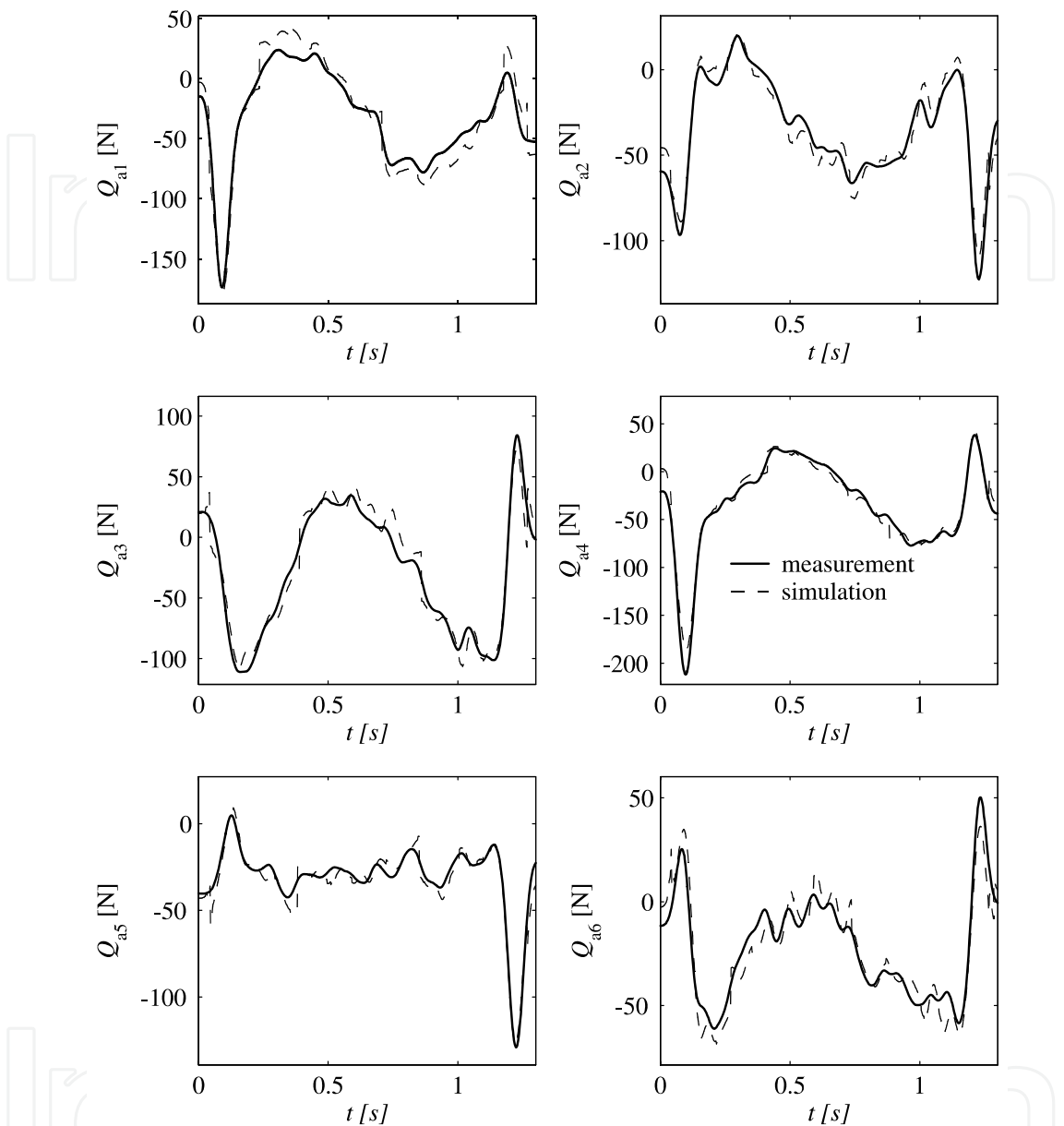


Figure 4. Accuracy validation of identified model of *PaLiDA*: Forces for a circular benchmark trajectory.

For both robot types, very good agreement between model output and experiment is noticeable. Of course, some deviations still remain, since a complete agreement of the model and the reality is quite impossible. Nevertheless, these results are an excellent base for the compensation of nonlinearities of both robots. For place reasons, the values of the identified parameters of the studied systems are not illustrated. We refer though to former publications with deeper insight and more discussion on dynamics identification of robots (Abdellatif et al., 2005b; Abdellatif et al., 2005d; Grotjahn et al., 2004; Grotjahn et al., 2001).

5. Model-based Feedforward control

The basics for implementing model-based feedforward control are now achieved. We have adequate modeled systems with accurate identified parameters. The next challenge is to use the attained knowledge for practical and industrial relevant control. The compensation of nonlinear dynamics can only be performed by using feedforward approaches because there is usually no possibility to change the feedback controller structure provided by the robot manufacturers. Anticipating for possible next commercial technologies, industrial control systems can have also a force/ torque interface. In that case, the controller input can be directly used by the robot's operator. However, the standard in industrial applications remains that only the desired path can be changed by the feedforward controller as other interfaces do not exist. The case of parallel robots is a prime example, that conventional and commercial control systems are less adequate for such new industrial application. It is believed, that the use of conventional technology widespread for serial robots or machine tools should be reconsidered. The paradigm of single-joint control do not regard highly nonlinear kinematic coupling and is one of the main reasons, that parallel robots still did not reach their promised potentials in practice. In the following we want to consider both possibilities of direct force/ torque input and the position pre-correction. Respecting the scope of this paper, it is more focused on the second case. First nonlinear feedforward control strategies are discussed, that are based on the friction and rigid-body model presented in section 3. Subsequently, compensation methods are presented which use linear models for improvement of path accuracy.

5.1 Nonlinear Compensation Control

5.1.1 Computed Force/Torque Feedforward Control

The computed force/ torque feedforward control is one of the most classic approaches in robotics (Sciavicco & Siciliano, 2000; Khalil & Dombre, 2002). A uniform scheme can be given according to the formalism defined in previous sections and is depicted in Fig 5. The global input consists in the minimal coordinates \mathbf{z} with an explicit or implicit dependency on time t , to enable the derivation of velocities and accelerations $\dot{\mathbf{s}}$ and $\ddot{\mathbf{s}}$. Only the nonlinear block ($\mathbf{z} \rightarrow \mathbf{q}_{a,d}$) depends on the robot's type. It consists in the trivial identity transformation for serial robots and the inverse kinematics (14) for parallel manipulators. For both systems, the desired forces (or torques) $\mathbf{Q}_{a,d}$ can be only derived from \mathbf{z} , $\dot{\mathbf{s}}$ and $\ddot{\mathbf{s}}$. Of course, such an approach depends on the presence of a motor-current interface to achieve the direct forward feeding of the calculated forces or torques.

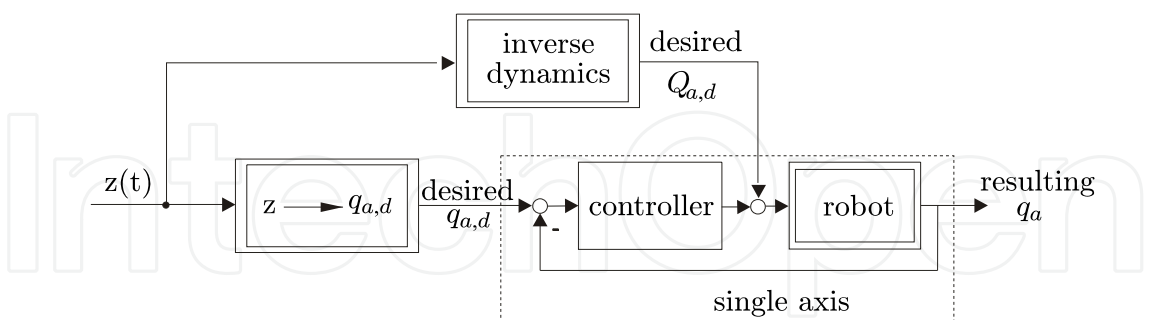


Figure 5. Uniform Scheme of Computed Force/ Torque Feedforward Control

The impact of the feedforward compensation of dynamics is depicted for the robot *PaLiDA* in Figure 6.

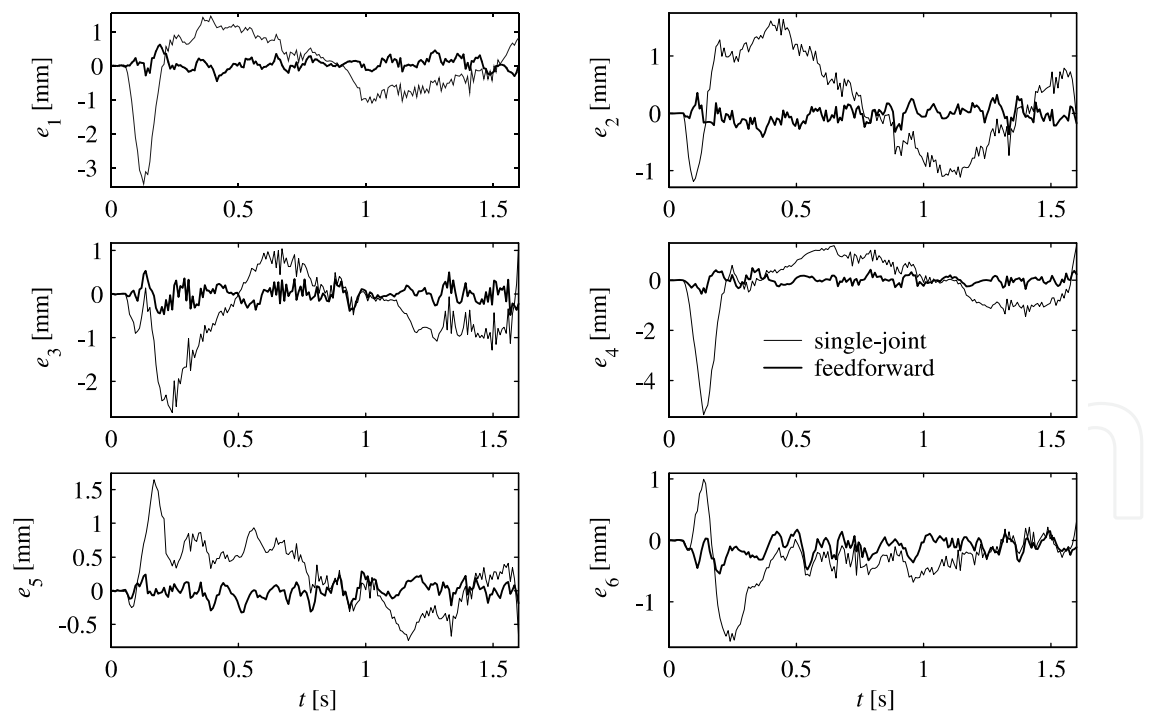


Figure 6. Tracking performance of the parallel robot *PaLiDA*. A Comparison between the single-joint and the feedforward approach.

The same circular motion presented in Figure 4 is investigated. Without doubt, the tracking errors for all six actuators were considerably decreased. As depicted, the control deviations resulting from the use of the simple decentralized single-joint control yields unacceptable errors (about 4 mm while acceleration). According to our experience, the compensation of the nonlinear dynamics is indispensable for operating parallel robots at high velocities and accelerations. Details on related experimental studies can be found in (Abdellatif et al., 2005a). A key role for the control improvement is assumed by the modeling of inverse dynamics and the appropriate identification of the related parameters.

5.1.2 Nonlinear Precompensation

By the absence of motor-current or force/ torque interface, a similar approach to the feedforward control can be applied, which we call *nonlinear pre-correction* (see Figure 7). The inverse dynamics model is computed the same way but is now provided to the inverse controller F_R^{-1} that yields the pre-correction terms $\Delta q_{a,d}$. The advantage of nonlinear trajectory pre-correction compared to the computed force/ torque method is that one only needs to convey path information to the robotic system. Force/ torque information are not necessary. Only a path interface is necessary. The approach is applicable to standard industrial robot systems since such an interface is usually provided for the integration of external sensor information into the control circuit. The proposed approach was implemented within the KRC1 standard control for the robot KUKA KR15. The improvements of tracking errors are depicted in Figure 8. The disadvantage of the nonlinear pre-correction is the necessity of a reliable controller model to be inverted. If there is no knowledge about the controller, experimental identification of the controller dynamics has to be carried out (Grotjahn & Heimann, 2002).

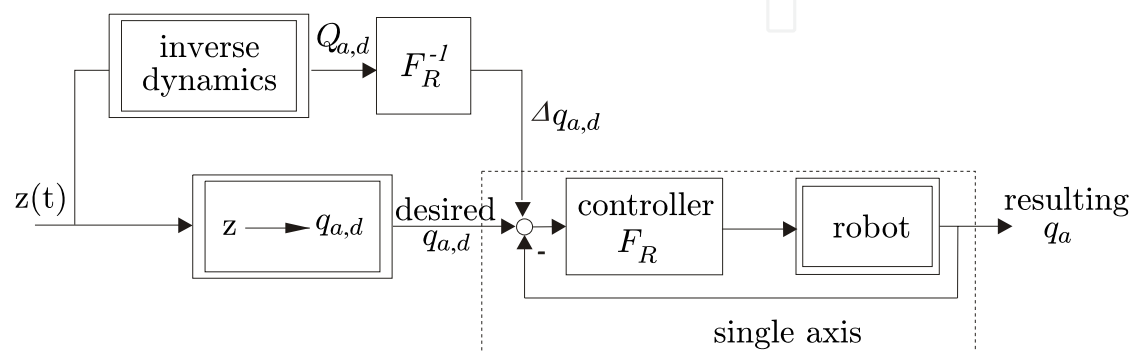


Figure 7. Uniform scheme of nonlinear pre-correction control.

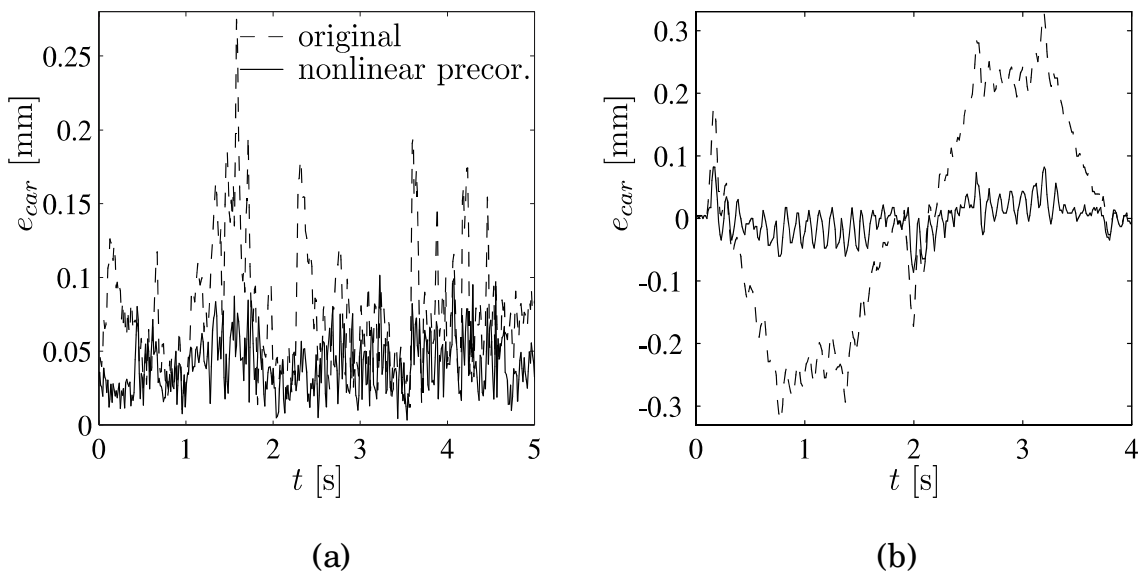


Figure 8. Path deviations of the *KUKA KR15*: (a) *ISO-9283* trajectory; (b) single motion of the second joint.

5.2 Feedforward Control by Linear Model

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 'iterative learning' algorithm that is presented in the following. Alternatively 'training' of a feedforward joint controller is explained. Both approaches are based on the actuation variables. Thus, their implementation is similar for parallel and serial robots.

5.2.1 Iterative Linear Learning

Learning control 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 (Moore, 1999; Longman, 2000). Thereby, the path accuracy is improved by iterative correction (see Figure 9).

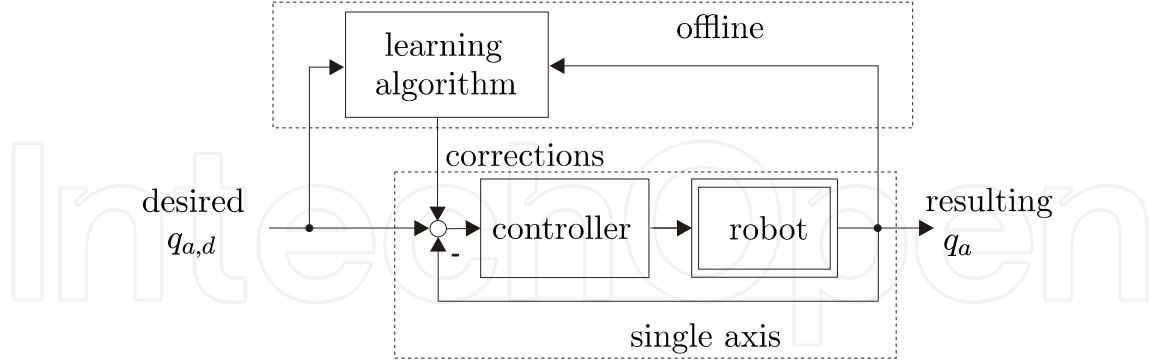


Figure 9. Principle of Iterative Learning Control.

For learning implementation, the robot is already under feedback-control, such that its closed-loop dynamics can be described in a linear discrete-time form by the state-space equations:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) + \mathbf{w}_1(k) \\ \mathbf{y}(k) &= \mathbf{C}(k)\mathbf{x}(k) + \mathbf{w}_2(k) \end{aligned} \quad (40)$$

with \mathbf{u} being the input and \mathbf{y} being the output. It is assumed that \mathbf{w}_1 represents some deterministic disturbance that appears every repetition and that \mathbf{w}_2 is the measurement disturbance. The aim of Learning is to change the command input every trial j using the learning control law:

$$\mathbf{u}_{j+1}(k) = \mathbf{f}_L(\mathbf{u}_j(k), \mathbf{y}_j(k), \mathbf{y}_d(k)) \quad (41)$$

such that the desired trajectory \mathbf{y}_d is tracked. Iterative learning control is called linear, when the learning law \mathbf{f}_L makes an iterative change in the input that is a linear combination of the error $\mathbf{e}_j = \mathbf{y}_j - \mathbf{y}_d$ measured in the previous repetition and the last input sequence

$$\mathbf{u}_{j+1} = \mathbf{u}_j + \mathbf{L}\mathbf{e}_j \Rightarrow \mathbf{q}_{a,j+1} = \mathbf{q}_{a,j} + \mathbf{L}(\mathbf{q}_{a,j} - \mathbf{q}_{a,d}) \quad (42)$$

The design of the learning gain matrix \mathbf{L} has to achieve desired properties. It is simple to derive the iterative error dynamics as

$$\mathbf{e}_{j+1} = (\mathbf{I} - \mathbf{P}\mathbf{L})\mathbf{e}_j \quad (43)$$

where \mathbf{I} is the identity matrix and

$$\mathbf{P} = \begin{bmatrix} g(1) & & & \\ g(2) & g(1) & & \\ \vdots & \vdots & \ddots & \\ g(N) & g(N-1) & \cdots & g(1) \end{bmatrix} \quad (44)$$

if a linear time-invariant actuator error dynamics is assumed, with an impulse response g . N is the length of the actual trajectory or input. Choosing \mathbf{L} to be \mathbf{P}^{-1} would lead to immediate zero tracking. The inversion of \mathbf{P} is equivalent to an inversion of the joint's closed-loop. As it can not be guaranteed that the identified 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, many methods can be used, like filtering suggested in (Norrlöf & Gunnarsson, 2002; Longman, 2000) etc. We propose here two methods. The first one is an extended LS-method or the Inverse-Covariance Kalman-Filter (Grotjahn & Heimann, 2002). It requires a good quality of measurement signals, which is the case for standard industrial robot. In the case when measurements of the actuator variables are importantly corrupted by noise, the inverse of \mathbf{P} is not adequate. Alternatively, a contraction mapping approach can be used (Longman, 2000), where the learning matrix is defined by: $\mathbf{L} = \Phi_1 \mathbf{P}^T$ for a learning gain Φ_1 that has to be chosen. The dynamics and therefore an estimate of the impulse response g of the closed-loop dynamics can be identified by applying standard procedures (Ljung, 1999).

The main advantage of learning control is the fact that it can compensate for influences and systematic deviations that are not captured by the model. This holds not only for non-modeled effects by the linear models used for learning, but also for effects not even reflected by the presented complex nonlinear model. Another advantage is that the actual deviations of end-effector position and orientation can be taken into account if they are measurable. The main disadvantage, however, is that every small change of the desired trajectory necessitates a new learning process. Each learning process needs some iterations until convergence, which can be undesirable by robot operators.

5.2.2 Training of Feedforward Controller

In order to avoid the disadvantages of learning control described in the previous section, Lange and Hirzinger proposed to 'train' a feedforward controller from the learnt behavior (Lange & Hirzinger, 1996). The feedforward controller model has to be identified by using e.g. the extended LS-method or the Inverse-Covariance Kalman-Filter (Grotjahn & Heimann, 2002). The scheme of training is given by Fig 10.

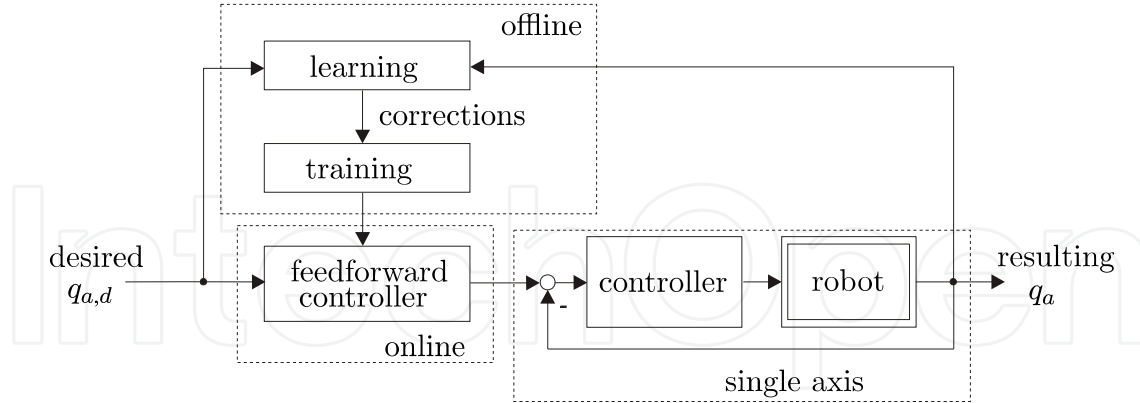


Figure 10. Training a Feedforward Controller.

As a matter of principal, 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 to transfer the learned behavior to similar trajectories. Consequently, small trajectory changes would not require a new learning process. The choice of the controller structure is a key issue. A fundamental condition is that the model can satisfactorily reproduce the learned behavior. In (Lange & Hirzinger, 1994), linear models like

$$\mathbf{q}_a(k) = \mathbf{q}_{a,d}(k) + \sum_{l=m_i}^m r(l)(\mathbf{q}_{a,d}(k+l) - \mathbf{q}_{a,d}(k)) \quad (45)$$

were proposed. This approach, however, cannot compensate nonlinear effects, like friction. Fig 11 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:

$$\begin{aligned} \mathbf{q}_a(k) = & \mathbf{q}_{a,d}(k) + \sum_{l=m_i}^m r(l)(\mathbf{q}_{a,d}(k+l) - \mathbf{q}_{a,d}(k)) \\ & + \sum_{l=p_i}^p s(l) \text{sign}(\dot{\mathbf{q}}_{a,d}(k+l) - \dot{\mathbf{q}}_{a,d}(k)) \end{aligned} \quad (46)$$

This auxiliary summand is suited to separate friction from inertial influences. After its consideration, learned corrections are reproduced much better than by the linear model. As expected, better tracking behavior can be obtained (see Fig 11). Although the performance of 'training' remains less than that of 'learn-

ing', it seems to be an alternative that combines good results with practical simplicity. The values m_t , m , p_t and p can be chosen by operators corresponding to the desired error dynamics. Using only positive values is equivalent to the consideration of posterior or future errors, only. It is recommended to choose negative m_t and s_t , such that the influence of previous dynamics, like high accelerations or velocities, can be taken into account.

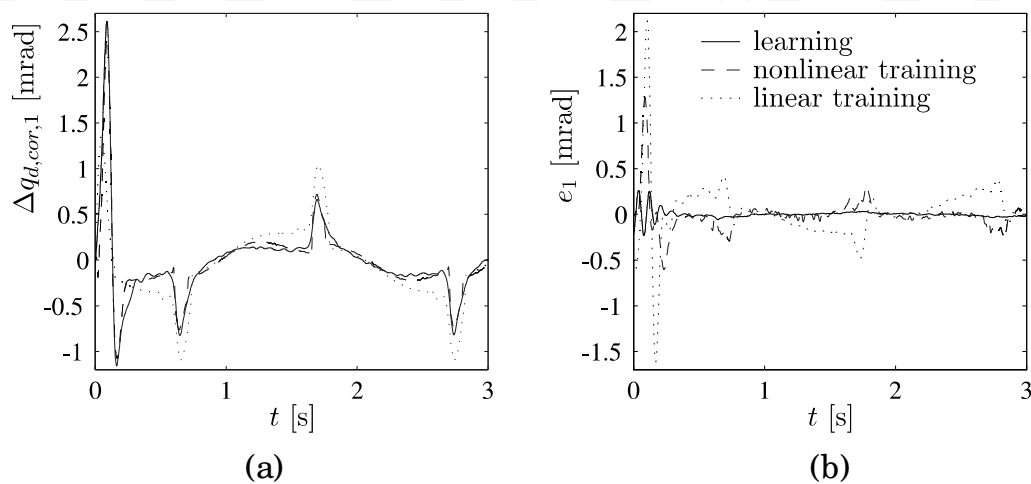


Figure 11. Behaviour of the first axis of the *manutec-r15* for a vertical circle: (a) learned and trained corrections; (b) comparison of resulting tracking errors

5.2.3 Application to serial robots

The presented compensation methods are investigated by experimental application to a classic serial manipulator: the *manutec-r15*. It is emphasized in the following on the resulting performance and on the applicability. Several different test trajectories are used. Here, the results are explained by regarding two trajectories: a vertical circle and a tracking of planar corner 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 are depicted in Figure 12. It shows some general results which can be reproduced for all tested trajectories. The 'training' yields the worst results of all methods. The nonlinear pre-correction is much better. Learning Control yields even further improvements. In order to numerically evaluate the results, the root mean square (RMS) of the cartesian errors are evaluated for the trajectories and for the different approaches. For all investigated trajectories, 'learning' leads to a decrease of at least 80 % after four iterations. The nonlinear pre-correction reduces the criterion by at least 60 %, whereas the 'training' leads to a minimum reduction of 35 %. Figure 12 depicts the tracking of the corner while changing the orientation by 90° for the *manutec-r15* robot. Although the

path velocity is only 0.02 m/ s, joint velocities and accelerations are quite high. Therefore, the couplings between different links have strong impact. These effects can not be compensated by the decoupled '*trained*' feedforward joint controllers.

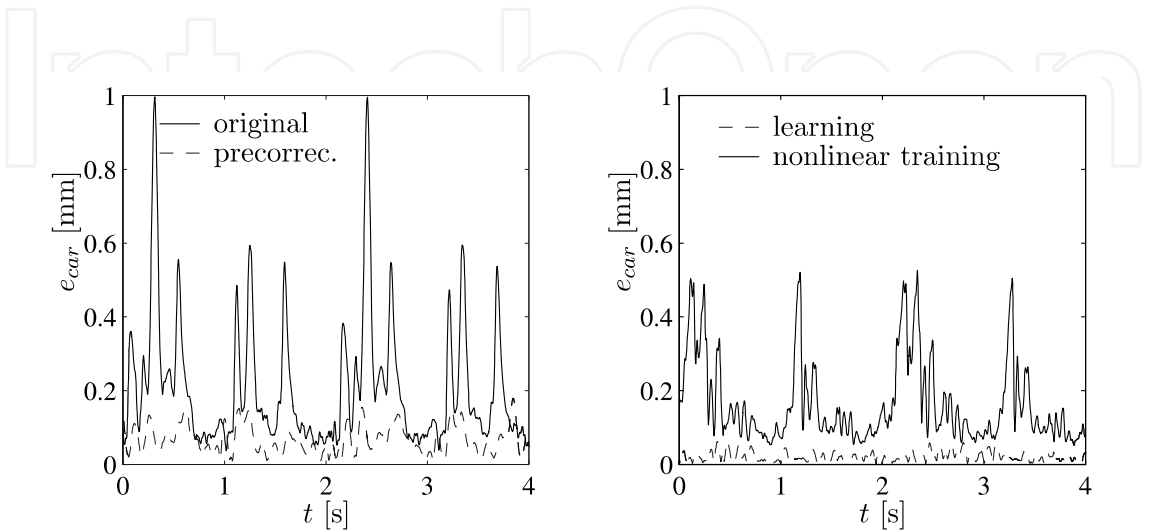


Figure 12. Comparison of cartesian path errors for a vertical circle of the *manutec-r15*

Nonlinear pre-correction is the only approach which combines efficient compensation of nonlinear deviations with practical applicability. '*Learning*' yields better results, but robustness and stability properties have to be 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. An extended experimental investigation can be found in (Grotjahn & Heimann, 2002).

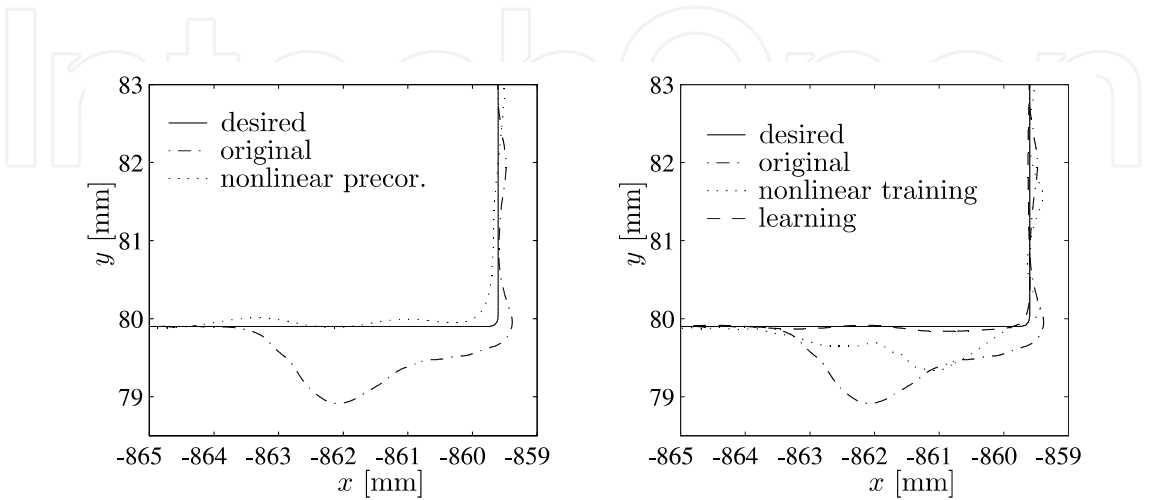


Figure 13. Tracking a Corner by the *Manutec-r15* Robot. Experimental Comparison of Different Feedforward Controllers

5.2.4 Application to parallel robots

In analogy to the case of serial robots, the compensation techniques were implemented to the parallel manipulator *PaLiDA*. It is important to notice, that the feedforward compensation techniques are all based on the actuation variables, for serial, as well as for parallel mechanisms. The procedure of experimental investigation and evaluation is very similar to that mentioned in the previous section. Figure 14 shows the improvement of tracking for the here considered robot. We compare the performances of simple single-joint control, Feedforward control and Learning control for a circular and a quadratic motion. The range of investigated velocity is here much higher than the case of serial robots. The average end-effector's velocity is equal to 1.5 ms^{-1} and 2 ms^{-1} for the circular and quadratic motion, respectively.

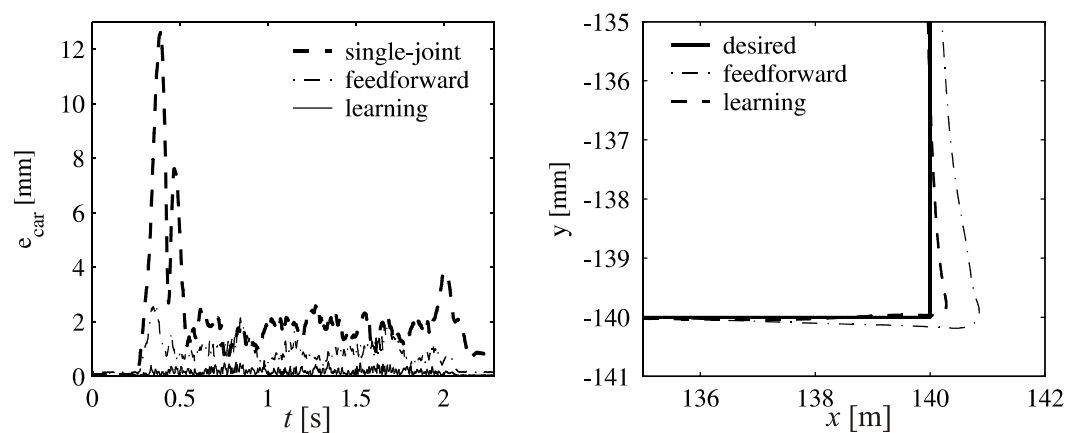


Figure 14. Comparison of Cartesian Path Errors for the Parallel Robot *PaLiDA*. Left: Circle Motion, Right: Quadratic Motion.

The feedforward control helps decreasing the cartesian RMS-error of about 60 %. Learning control decreased the errors of at least 90 %, which is very satisfactory. Such improvement can be clearly observed in Figure 14. Conventional control strategies are not acceptable for operating robots in an accurate manner. The Integration of model-based feedforward compensators, such inverse dynamics or learning controller yield impressive improvement of tracking accuracy.

6. Conclusions

This chapter presented a uniform and coherent methodology for model-based control of industrial robots. To take account of the technological evolution over the last years, classic approaches were extended to the class of parallel kinematic manipulators or parallel robots. A revision of the basics is necessary. Many assumptions that became common due to the reputation of classic open-chain robots were highlighted and revised. This is necessary to be able of developing uniform approaches for handling serial and parallel robots. The idea of this work was to exploit the similarities and to consider the differences between two types. The similarities can be provided by the same modules (calculation, control, etc.). The differences are considered by interfaces (transformations etc.) that account for robot inherent properties.

Already at the basic level of modeling the kinematics and dynamics, the uniformity of the methods can be achieved by considering the generalized coordinates and velocities to be the formal conjunction of serial and parallel robots. It is then possible to apply e.g. generic algorithms and efficient calculation of the inverse dynamics, such that the presented solutions remain valid for a wide class of robots. This was also the case for developing identification strategies of the model parameters. It was demonstrated, that with light adaptation, the same algorithms and experimental strategies can be applied for serial robots and for parallel manipulators.

In the praxis of control, it is the type of the control system that is more crucial for successful implementation, rather than the robot structure. If a force/ torque interface is provided, all feedforward strategies can be applied in the same way for parallel and serial robots, since the desired motions are given. If only a position interface is supplied, it is practicable to use correction techniques at actuator levels. The chapter presented nonlinear and linear approaches. The nonlinear pre-correction techniques are recommended for typical industrial control systems and have demonstrated impressive performance. Iterative learning and training algorithms offer the possibilities to use computational efficient linear models. Like substantiated by experimental results the improvement of the control accuracy was investigated for serial and parallel robots.

7. References

- Abdellatif, H.; Grotjahn, M. & Heimann, B. (2005a). High efficient dynamics calculation approach for computed-force control of robots with parallel structures, *Proceedings of 44th IEEE Conference on Decision and Control and the 2005 European Control Conference (CDC-ECC05)*, pp. 2024-2029, Seville, 2005.
- Abdellatif, H.; Heimann, B. & Grotjahn, M. (2005b). Statistical approach for bias-free identification of a parallel manipulator affected with high measurement noise,

- Proceedings of the 44th IEEE Conference on Decision and Control (CDC-ECC05)*, pp. 3357-3362, Seville, 2005.
- Abdellatif, H.; Heimann, B. & Holz, C. (2005c). Time-effective direct dynamics identification of parallel manipulators for model-based feedforward control, *Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM2005)*, pp. 777-782, Monterey, 2005.
- Abdellatif, H.; Heimann, B. & Holz, C. (2005d). Applying efficient computation of the mass matrix for decoupling control of complex parallel manipulators, *Preprints of the 16th IFAC World Congress*, Prag, 2005.
- Angeles, J. (2003). *Fundamentals of Robotic Mechanical Systems*, Springer, New York, Berlin, Heidelberg, 2003.
- Armstrong-Hélouvry, B. (1991). *Control of Machines with Friction*, Kluwer Academic Publishers, Boston.
- Bremer, H. (1988). *Dynamik und Regelung mechanischer Systeme*, Teubner, Stuttgart.
- Bona, B. & Indri, M. (2005). Friction compensation in robotics: an overview, *Proceedings of the 44th IEEE Conference on Decision and Control and the 2005 European Control Conference (CDC-ECC05)*, pp. 4360-4367, Seville, 2005.
- Bonev, I. (2002). *Geometric analysis of parallel mechanisms*, Faculté des études supérieures de l'Université Laval.
- Cheng, H.; Kuen, Y. & Li, Z. (2003). Dynamics and control of redundantly actuated parallel manipulators. *IEEE/ASME Transactions on Mechatronics*, 8, 4, (2003) 483-491.
- Fiset, P.; Raucourt, B. & Samin, J. C. (1996). Minimal Dynamic Characterization of Tree-Like Multibody Systems. *Nonlinear Dynamics*, 9, 1-2 (1996) 165-184.
- Gautier, M. & Khalil, W. (1990). Direct calculation of minimum set of inertial parameters of serial robots. *IEEE Transactions on Robotics and Automation*, 6, 3 (1990) 368-373.
- Gosselin, C. & Angeles, J. (1990). Singularity analysis of closed-loop kinematic chains. *IEEE Transactions on Robotics and Automation*, 6, 3, (1990) 281-290.
- Grotjahn, M. & Heimann, B. (2000). Determination of dynamic parameters of robots by base sensor measurements, *Proceedings of the sixth IFAC Symposium on Robot Control (SYROCO)*, Vienna, 2000.
- Grotjahn, M.; Daemi, M. & Heimann, B. (2001). Friction and rigid body identification of robot dynamics. *International Journal of Solids and Structures*, 38, (2001) 1889-1902.
- Grotjahn, M. & Heimann, B. (2002). Model-based feedforward control in industrial robotic. *International Journal of Robotics research*, 21, 1, (2002) 99-114.
- Grotjahn, M.; Heimann, B. & Abdellatif, H. (2004). Identification of friction and rigid-body dynamics of parallel kinematic structures for model-based control. *Multibody System Dynamics*, 11, 3, (2004) 273-294.
- Khalil, W. & Kleinfinger, J. (1986). A new geometric notation for open and closed-loop robots, *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, pp. 1174-1179, San Francisco, 1986.
- Khalil, W. & Kleinfinger, J.-F. (1987). Minimum Operations and Minimum Parameters of the Dynamic Models of Tree Structure Robots. *IEEE Transactions of Robotics and Automation*, 3, 6 (1987) 517-526.

- Khalil, W. & Dombre, E. (2002). *Modelling, Identification and Control of Robots*, Hermes, London
- Khalil, W. & Guegan, S. D. (2004). Inverse and direct dynamics modeling of gough-stewart robots. *IEEE Transactions on Robotics*, 20, 4, (2004) 754-762.
- Lange, F. & Hirzinger, G. (1994). Learning to improve the path accuracy of position controlled robots, *Proceedings of the Conference on Intelligent Robots and Systems*, pp. 494-501, Munich, 1994.
- Lange, F. & Hirzinger, G. (1996). Learning of a controller for non-recurring fast movements. *Advanced Robotics*, 10, 2 (1996) 229-244.
- Ljung, L. (1999). *System Identification: Theory for the User*, Prentice-Hall, New Jersey.
- Longman, R. W. (2000). Iterative learning control and repetitive learning control for engineering practice. *International Journal of Control*, 73, 10, (2000) 930-954.
- Meirovitch, L. (1970). *Methods of Analytical Dynamics*, McGraw-Hill, New York.
- Merlet, J.-P. (2000). *Parallel Robots*, Kluwer Academic Publishers, Dordrecht.
- Moore, K. L. (1999). Iterative learning control: An expository overview. *Applied and Computational Control, Signals and Circuits*, 1, (1999) 151-214.
- Norrlöf, M. & Gunnarsson, S. (2002). Experimental Results of Some Classical Iterative Learning Control Algorithms. *IEEE Transactions on Robotics and Automation*, 18, 4 (2002) 636-641.
- Sciavicco, L. & Siciliano, B. (2000). *Modeling and Control of Robot Manipulators*, Springer, London.
- Swevers, J.; Ganseman, C.; Tükel, D.; Schutter, J. d. & Brussel, H. v. (1997). Optimal robot excitation and identification. *IEEE Transactions on Robotics and Automation*, 13, 5, (1997) 730-740.
- Ting, Y.; Chen, Y.-S. & Jär, H.-C. (2004). Modeling and control for a gough-stewart platform cnc-machine. *Journal of Robotic Systems*, 21, 11, (2004) 609-623.
- Tsai, L.-W. (1999). *Robot Analysis*, Wiley-Interscience, New York.
- Walter, E. & Pronzato, L. (1997). *Identification of Parametric Models from Experimental Data*, Springer, London.



Industrial Robotics: Theory, Modelling and Control

Edited by Sam Cubero

ISBN 3-86611-285-8

Hard cover, 964 pages

Publisher Pro Literatur Verlag, Germany / ARS, Austria

Published online 01, December, 2006

Published in print edition December, 2006

This book covers a wide range of topics relating to advanced industrial robotics, sensors and automation technologies. Although being highly technical and complex in nature, the papers presented in this book represent some of the latest cutting edge technologies and advancements in industrial robotics technology. This book covers topics such as networking, properties of manipulators, forward and inverse robot arm kinematics, motion path-planning, machine vision and many other practical topics too numerous to list here. The authors and editor of this book wish to inspire people, especially young ones, to get involved with robotic and mechatronic engineering technology and to develop new and exciting practical applications, perhaps using the ideas and concepts presented herein.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Houssem Abdellatif and Bodo Heimann (2006). Model-Based Control for Industrial Robots: Uniform Approaches for Serial and Parallel Structures, *Industrial Robotics: Theory, Modelling and Control*, Sam Cubero (Ed.), ISBN: 3-86611-285-8, InTech, Available from:

http://www.intechopen.com/books/industrial_robotics_theory_modelling_and_control/model-based_control_for_industrial_robots__uniform_approaches_for_serial_and_parallel_structures

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2006 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen