19

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 tricpet 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 filed.

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-

Source: Industrial-Robotics-Theory-Modelling-Control, ISBN 3-86611-285-8, pp. 964, ARS/plV, Germany, December 2006, Edited by: Sam Cubero

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 $\boldsymbol{q}_\mathrm{a}$, that defines the actuation space A. Additionally, passive joints are denoted by $\boldsymbol{q}_\mathrm{p}$. Both vectors can be grouped in the joint vector $\boldsymbol{q} = \left[\boldsymbol{q}_\mathrm{a}^\mathrm{T} \boldsymbol{q}_\mathrm{p}^\mathrm{T}\right]^\mathrm{T}$ 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 $\boldsymbol{x}$ containing the position and orientation of the end-effector (EE) with respect to the inertial frame. Let the vector $\boldsymbol{z}$ 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 DOFs (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 $\boldsymbol{x}$ 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 than 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 $\boldsymbol{z}$. 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 $(KS)_\mathrm{E}$, each link i is associated with body-fixed frame $(KS)_i$. For efficient formulation and calculation of the homogenous transformations (given by $\boldsymbol{T}_i^{i-1}$) 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:

$$\boldsymbol{T}_i^{i-1} = \begin{bmatrix} \boldsymbol{R}_i^{i-1} & {}_{(i-1)}\boldsymbol{r}_i^{i-1} \\ 000 & 1 \end{bmatrix} = \begin{bmatrix} c_{\vartheta i} & s_{\vartheta i} & 0 & a_i \\ s_{\vartheta i} c_{\alpha i} & c_{\vartheta i} c_{\alpha i} & -s_{\alpha i} & -d_i s_{\alpha i} \\ s_{\vartheta i} s_{\alpha i} & c_{\vartheta i} s_{\alpha i} & c_{\alpha i} & d_i c_{\alpha i} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

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 $\boldsymbol{R}_{\boldsymbol{i}}^{\boldsymbol{i-1}}$ and the vector ${}_{(i-1)}\boldsymbol{r}_i^{i-1}$ 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 ${}_{(i)}\boldsymbol{v}_{\boldsymbol{i}}$ and angular velocity ${}_{(i)}\boldsymbol{\omega}_i$ of each link i and the corresponding accelerations can be calculated recursively by the following equations:

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

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

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

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

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

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:

$$\boldsymbol{f}: A \rightarrow W$$
$$\mathrm{q}_a \rightarrow \boldsymbol{x} = \mathrm{f}\left(\boldsymbol{q}_{\boldsymbol{a}}\right)$$

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

$$\boldsymbol{g}: \mathrm{W} \rightarrow \mathrm{A}$$
$$\boldsymbol{x} \rightarrow \boldsymbol{q}_{\mathrm{a}} = \boldsymbol{g}(\boldsymbol{x})$$

We mentioned above, that only the minimal coordinates describe the system uniquely. Consequently, only the transformations having the argument set being 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 $(\mathrm{C} \equiv \mathrm{W})$. 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 $(\mathrm{C} \equiv \mathrm{A})$, 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 $\boldsymbol{z} = \boldsymbol{x}$ can not be measured.

For both robotic types the pose vector is defined as:

$$\boldsymbol{x} = \begin{bmatrix} x & y & z & \alpha & \beta & \gamma \end{bmatrix}^{\mathrm{T}},$$

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

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

2.3 Differential Kinematics

The differential kinematics maps the velocity of the end-effector into the velocity of the actuated joints $\dot{\boldsymbol{q}}_{\mathrm{a}}$ 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

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \frac{\partial \boldsymbol{f}_1}{\partial q_{\mathrm{a},1}} & \cdots & \frac{\partial f_1}{\partial q_{\mathrm{a},\mathrm{n_a}}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial q_{\mathrm{a},1}} & \cdots & \frac{\partial f_n}{\partial q_{\mathrm{a},\mathrm{n_a}}} \end{bmatrix} \dot{\boldsymbol{q}}_{\mathrm{a}} \tag{7}$$

or simply

$$\dot{\boldsymbol{x}} = \boldsymbol{J}_{\mathrm{A}} \dot{\boldsymbol{q}}_{\boldsymbol{a}}. \tag{8}$$

The analytical Jacobian $\boldsymbol{J}_A$ 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: $\boldsymbol{\omega}_\mathrm{E} = \begin{bmatrix}\omega_x & \omega_y & \omega_z\end{bmatrix}^\mathrm{T}$, such that

$$\boldsymbol{\omega}_\mathrm{E} = \begin{bmatrix}\omega_x \\ \omega_y \\ \omega_z\end{bmatrix} = \underbrace{\begin{bmatrix}1 & 0 & s_\beta \\ 0 & c_\alpha & -s_\alpha c_\beta \\ 0 & s_\alpha & c_\alpha c_\beta\end{bmatrix}}_{\boldsymbol{R}_\mathrm{K}(\alpha,\beta)} \begin{bmatrix}\dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma}\end{bmatrix} \tag{9}$$

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

$$\boldsymbol{v}_\mathrm{E} = \begin{bmatrix}\dot{\boldsymbol{r}}_\mathrm{E} \\ \boldsymbol{\omega}_\mathrm{E}\end{bmatrix} = \boldsymbol{J}\dot{\boldsymbol{q}}_\mathrm{a}, \text{ with } \boldsymbol{J} = \begin{bmatrix}\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_\mathrm{K}\end{bmatrix}\boldsymbol{J}_A \tag{10}$$

By regarding eq. (7) it is obvious that the analytic derivation of the jacobian is only available, when the direct kinematic solution $\boldsymbol{f}(\boldsymbol{q}_\mathrm{a})$ is given in a closed form. This is the case for classic open-chain robots, whereas for parallel robots, the inverse jacobian $\boldsymbol{J}^{-1}$ 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 $(\det(\boldsymbol{J}) = 0)$. 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 $\dot{\boldsymbol{s}}$. They always present a linear combination of the time-derivatives of the generalized coordinates $\dot{\boldsymbol{z}}$. The simplest case is when these combinations correspond to the identity:

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

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

$\dot{\boldsymbol{s}} = \boldsymbol{v}_{\mathrm{E}} \neq \dot{\boldsymbol{x}} = \dot{\boldsymbol{z}}$. 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 $\boldsymbol{J}_{T_i}$ and $\boldsymbol{J}_{\boldsymbol{R}_i}$ that relate its translational and its angular velocities to the generalized velocities of the robot, respectively:

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

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

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

$$\boldsymbol{J}_{R_i} = \boldsymbol{R}_{\boldsymbol{i-1}}^{i} \boldsymbol{J}_{\boldsymbol{R}_{\boldsymbol{i-1}}} + \boldsymbol{e}_z \frac{\partial \dot{\vartheta}_i}{\partial \dot{\boldsymbol{s}}} \tag{12}$$

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 $\boldsymbol{T}_i^{i-1}$. By solving

$$\boldsymbol{T}_{\mathrm{E}}^{0}(\boldsymbol{q}_{\mathrm{a}}) = \prod\nolimits_{i=0}^{n} \boldsymbol{T}_{\boldsymbol{i}}^{\boldsymbol{i-1}}(q_{a,i}) \stackrel{!}{=} \boldsymbol{T}_{\mathrm{E}}^{0}(\boldsymbol{x}),$$

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

$$\boldsymbol{J}(\boldsymbol{q}) = \left[\boldsymbol{J}_1 \mid \boldsymbol{J}_2 \mid \ldots \mid \boldsymbol{J}_n\right] \text{ with } \boldsymbol{J}_i = \begin{bmatrix} {}_{(0)}\boldsymbol{e}_{\boldsymbol{z}}^{\boldsymbol{i}} \times {}_{(0)}\boldsymbol{r}_{\mathrm{E}}^{i} \\ {}_{(0)}\boldsymbol{e}_z^{i} \end{bmatrix} \tag{13}$$

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

$$\boldsymbol{r}_{B_j}^{A_j} = \begin{bmatrix} x_j & y_j & z_j \end{bmatrix}^{\mathrm{T}} = -\boldsymbol{r}_{\boldsymbol{A}_j}^{0} + \boldsymbol{r}_{\mathrm{E}}^{0} + \boldsymbol{R}_{\mathrm{E}\,(\mathrm{E})}^{0} \boldsymbol{r}_{\boldsymbol{B}_j}^{E} . \tag{14}$$

Thus, every chain can now be regarded separately as a conventional open-chain robot with a corresponding end-effector position at $\boldsymbol{r}_{B_j}^{A_j}$. 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

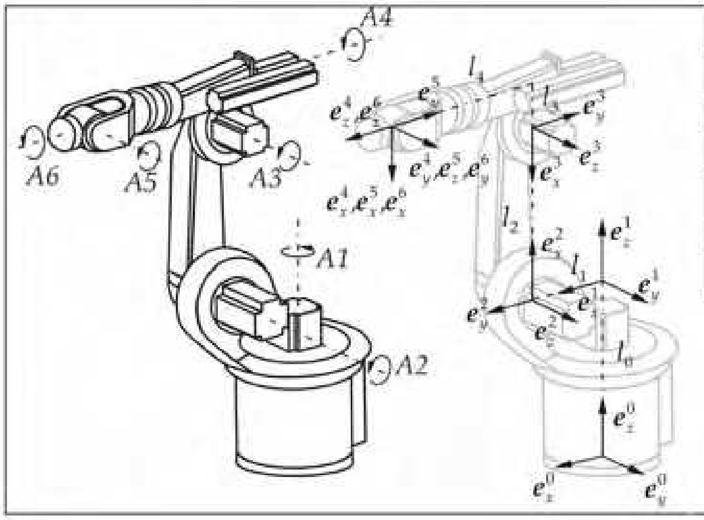
$$q_{\mathrm{a}_j} = l_j = \sqrt{x_j^2 + y_j^2 + z_j^2} \tag{15}$$

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

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

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

$$\boldsymbol{J}(\boldsymbol{x})^{-1} = \frac{\partial \dot{\boldsymbol{q}}_{\mathrm{a}}}{\partial \dot{\boldsymbol{s}}} = \begin{bmatrix} \dfrac{\partial \dot{\boldsymbol{q}}_{\mathrm{a}}}{\partial \dot{\boldsymbol{r}}_{\mathrm{E}}} & \dfrac{\partial \dot{\boldsymbol{q}}_{\mathrm{a}}}{\partial \dot{\boldsymbol{\omega}}_{\mathrm{E}}} \end{bmatrix} \tag{18}$$



i	θ_i	d_i	a_i	α_i
1	$-q_1$	l_0	0	0
2	q_2	0	l_1	$-\frac{\pi}{2}$
3	$q_3+\frac{\pi}{2}$	0	l_2	0
4	$-q_4$	l_4	$-l_3$	$\frac{\pi}{2}$
5	q_5	0	0	$-\frac{\pi}{2}$
6	q_6	0	0	$\frac{\pi}{2}$

$l_0 = 675\,\text{mm}$
$l_1 = 300\,\text{mm}$
$l_2 = 650\,\text{mm}$
$l_3 = 155\,\text{mm}$
$l_4 = 600\,\text{mm}$

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

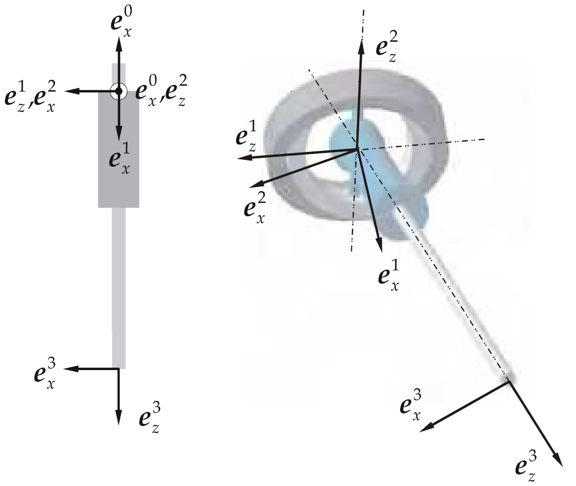
Many methods are proposed in the literature for calculating the inverse jacobian. We propose here the most straight-forward way in our case. Every single chain j corresponds to the j^{th} raw of the inverse jacobian:

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

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

$$\boldsymbol{v}_{\boldsymbol{B}_j} = \dot{\boldsymbol{r}}_{\text{E}} + \tilde{\boldsymbol{\omega}}_{\text{E}} r_{B_j}^{\text{E}} = \dot{\boldsymbol{r}}_{\text{R}} + \tilde{\boldsymbol{r}}_{\boldsymbol{B}_j}^{\text{E}} \omega_{\text{E}} \tag{20}$$

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 $\boldsymbol{v}_{B_j}$ that is needed for (19).



i	ϑ_i	d_i	a_i	α_i
1	$\alpha_i - \frac{\pi}{2}$	0	0	$\frac{\pi}{2}$
2	$\beta_i - \frac{\pi}{2}$	0	0	$-\frac{\pi}{2}$
3	0	l_i	0	$-\frac{\pi}{2}$

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 $\boldsymbol{Q}_\mathrm{a}$ 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:

$$\boldsymbol{Q}_\mathrm{a} = \boldsymbol{B}_\mathrm{a}(\boldsymbol{z}, \dot{\boldsymbol{s}}, \ddot{\boldsymbol{s}}) \tag{21}$$

$$\boldsymbol{Q}_\mathrm{a} = \boldsymbol{M}_\mathrm{a}(\boldsymbol{z})\ddot{\boldsymbol{q}}_\mathrm{a} + \boldsymbol{N}(\boldsymbol{z}, \dot{\boldsymbol{s}}) \tag{22}$$

$$\boldsymbol{Q}_\mathrm{a} = \boldsymbol{M}_\mathrm{a}(\boldsymbol{z})\ddot{\boldsymbol{q}}_\mathrm{a} + \boldsymbol{c}_\mathrm{a}(\boldsymbol{z}, \dot{\boldsymbol{s}}) + \boldsymbol{g}_\mathrm{a}(\boldsymbol{z}) \tag{23}$$

$$\boldsymbol{Q}_\mathrm{a} = \boldsymbol{A}_\mathrm{a}(\boldsymbol{z}, \dot{\boldsymbol{s}}, \ddot{\boldsymbol{s}})\boldsymbol{p}_\mathrm{min} \tag{24}$$

where $\boldsymbol{Q}_{\mathrm{a}}$ being the vector of actuation forces. The massmatrix is denoted by $\boldsymbol{M}$. The vectors $\boldsymbol{c}$ and $\boldsymbol{g}$ contain the centrifugal and Coriolis, and the gravitational terms, respectively. The vector $\boldsymbol{N}$ includes implicitly $\boldsymbol{c}$ and $\boldsymbol{g}$. Analogically, the vector $\boldsymbol{B}(\boldsymbol{z},\dot{\boldsymbol{s}},\ddot{\boldsymbol{s}})$ 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 $\boldsymbol{z}$, $\dot{\boldsymbol{s}}$ and $\ddot{\boldsymbol{s}}$, whereas the outputs are always given in the actuation space: $\boldsymbol{Q}_{\mathrm{a}}$. 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 efficient[1] 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

$$\partial\dot{\boldsymbol{s}}^{\mathrm{T}}\boldsymbol{\tau} = \partial\dot{\boldsymbol{q}}_{\mathrm{a}}^{\mathrm{T}}\boldsymbol{Q}_{\mathrm{a}} \Leftrightarrow \boldsymbol{\tau} = \left(\frac{\partial\dot{\boldsymbol{q}}_{\mathrm{a}}}{\partial\dot{\boldsymbol{s}}}\right)^{\mathrm{T}}\boldsymbol{Q}_{\mathrm{a}} \tag{25}$$

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:

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

---

[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:

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

with $\boldsymbol{a}_{\mathrm{S}_i} = \dot{\boldsymbol{v}}_{\mathrm{S}_i} - \boldsymbol{g}$ 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 $\dot{\boldsymbol{v}}_{\mathrm{S}_i}$, m_i and $\boldsymbol{I}_{\boldsymbol{i}}^{(S_i)}$, respectively. To be able of using the recursion calculation of kinematic quantities (2-5, 11), eq. (27) is transformed to

$$\boldsymbol{\tau} = \sum_{\iota=1}^{N_K}\left[\underbrace{\left(\frac{\partial\, {}_{(i)}\boldsymbol{v}_i}{\partial \dot{\boldsymbol{s}}}\right)^{\mathrm{T}}}_{\boldsymbol{J}_{T_i}^{\mathrm{T}}} \left(m_{i\,(i)}\boldsymbol{a}_i + {}_{(i)}\dot{\tilde{\boldsymbol{\omega}}}_i \boldsymbol{s}_i + {}_{(i)}\tilde{\boldsymbol{\omega}}_{i\,(i)}\tilde{\boldsymbol{\omega}}_i \boldsymbol{s}_i\right)\right.$$

$$\left. + \underbrace{\left(\frac{\partial\, {}_{(i)}\boldsymbol{\omega}_i}{\partial \dot{\boldsymbol{s}}}\right)^{\mathrm{T}}}_{\boldsymbol{J}_{R_i}^{\mathrm{T}}} \left({}_{(i)}\boldsymbol{I}_{\boldsymbol{i}}^{(i)}\,{}_{(i)}\dot{\boldsymbol{\omega}}_i + {}_{(i)}\tilde{\boldsymbol{\omega}}_i \left({}_{(i)}\boldsymbol{I}_{\boldsymbol{i}}^{(i)}\,{}_{(i)}\boldsymbol{\omega}_i\right) + \tilde{\boldsymbol{s}}_{i\,(i)}\boldsymbol{a}_i\right)\right] \tag{28}$$

with $\boldsymbol{s}_i$ being the vector of the i^{th} body's first moment[2] $\boldsymbol{s}_i = \begin{bmatrix} s_{i_x} & s_{i_y} & s_{i_z}\end{bmatrix}^{\mathrm{T}} = m_{i\,(i)}\boldsymbol{r}_{\boldsymbol{S_i}}^{\boldsymbol{i}}$ ($\boldsymbol{r}_{\boldsymbol{S_i}}^{\boldsymbol{i}}$: location of S_i with respect to the limb-fixed coordinate frame) and ${}_{(i)}\boldsymbol{I}_i^{(i)}$ being the inertia tensor about the same coordinate frame.

It is obvious, that the calculation of $\boldsymbol{\tau}_{\mathrm{rb}}$ 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 $(\dot{\boldsymbol{s}} \equiv \dot{\boldsymbol{q}}_{\mathrm{a}})$

$$\left(\frac{\partial \dot{\boldsymbol{q}}_{\mathrm{a}}}{\partial \dot{\boldsymbol{s}}}\right)^{-\mathrm{T}} = \left(\frac{\partial \dot{\boldsymbol{q}}_{\mathrm{a}}}{\partial \dot{\boldsymbol{q}}_{\mathrm{a}}}\right)^{-\mathrm{T}} = \boldsymbol{I}$$

[2] not to confuse with generalized velocities $\dot{\boldsymbol{s}}$

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

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

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 $\boldsymbol{p}_{\min}$.

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, $\boldsymbol{s}_i$ and $\boldsymbol{I}_i$. To regroup such parameters, the definition of two new operators $(.)^*$ and $(.)^\Diamond$ are required first:

$$\boldsymbol{\omega}_i^* \boldsymbol{I}_i^\Diamond := {}_{(i)}\boldsymbol{I}_i^{(i)}\, {}_{(i)}\boldsymbol{\omega}_i \tag{29}$$

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

The inverse dynamics (28) can be written as

$$\begin{aligned}
\boldsymbol{\tau}_{\mathrm{rb}} &= \sum_{i=1}^{N_K} \underbrace{\begin{bmatrix} \boldsymbol{J}_{\mathrm{T}_i}^{\mathrm{T}} & \boldsymbol{J}_{\mathrm{R_i}}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \boldsymbol{0} & {}_{(i)}\dot{\tilde{\boldsymbol{\omega}}} + {}_{(i)}\tilde{\boldsymbol{\omega}}_i\, {}_{(i)}\tilde{\boldsymbol{\omega}}_i & {}_{(i)}\boldsymbol{a}_i \\ {}_{(i)}\dot{\boldsymbol{\omega}}_i^* + {}_{(i)}\tilde{\boldsymbol{\omega}}_i\, {}_{(i)}\boldsymbol{\omega}_i^* & -{}_{(i)}\tilde{\boldsymbol{a}}_i & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{H}_i} \underbrace{\begin{bmatrix} \boldsymbol{I}_i^\Diamond \\ \boldsymbol{s}_i \\ m_i \end{bmatrix}}_{\boldsymbol{p}_{\mathrm{rb},i}} \\
&= \underbrace{\begin{bmatrix} \boldsymbol{H}_1 & \boldsymbol{H}_2 & \dots & \boldsymbol{H}_{N_{\mathrm{K}}} \end{bmatrix}}_{\boldsymbol{H}(\boldsymbol{z},\dot{\boldsymbol{s}},\ddot{\boldsymbol{s}})} \underbrace{\begin{bmatrix} \boldsymbol{p}_1^{\mathrm{T}} & \boldsymbol{p}_2^{\mathrm{T}} & \dots & \boldsymbol{p}_{N_{\mathrm{K}}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}}_{\boldsymbol{p}_{\mathrm{rb}}}
\end{aligned} \tag{30}$$

which is now linear in respect to the parameter set $\boldsymbol{p}_{\mathrm{rb}}$, 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 inertiatensor 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 $\boldsymbol{H}_i$. The dimension of $\boldsymbol{p}_{\mathrm{rb}}$ 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; Gauier & Khalil, 1990; Fisette 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 $\boldsymbol{H}_i$ in (30) can be grouped for single serial kinematic chains, such that a recursive calculation:

$$\boldsymbol{H}_i = \boldsymbol{H}_{i-1}\boldsymbol{L}_i + \boldsymbol{K}_i \tag{31}$$

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

$$\boldsymbol{h}_j = \sum_{l=1}^{k} a_{lj}\boldsymbol{h}_{lj} \tag{32}$$

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

$$p_{\mathrm{rb},lj,\mathrm{new}} = p_{\mathrm{rb},lj} + a_{lj}p^*_{\mathrm{rb},j} \tag{33}$$

The recursive relationship given in (31) can be used for parameter reduction. If one column or a linear combination of columns of $\boldsymbol{L}_i$ is constant with respect to the joint variable and the corresponding columns of $\boldsymbol{K}_i$ are zero columns, the parameters can be regrouped. This leads to the rules which are formulated in (Gautier & 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 9^{th}, the 10^{th} and the sum of the 1^{st} and 4^{th} columns of $\boldsymbol{L}_i$ and $\boldsymbol{K}_i$ 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 $\boldsymbol{B}_j$ 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 Q, usually as nonlinear characteristics $Q_{f_i}(\dot{q}_i) = f(\dot{q}_i)$ with respect to the joint velocity, i.e.

$$Q_{f_i}(\dot{q}_i) = r_{1_i}\,\mathrm{sign}(\dot{q}_i) + r_{2_i}\dot{q}_i \tag{34}$$

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

$$\boldsymbol{Q}_{\mathrm{a,f}} = \left(\frac{\partial\dot{\boldsymbol{q}}}{\partial\dot{\boldsymbol{q}}_{\mathrm{a}}}\right)^{\mathrm{T}} \boldsymbol{Q}_{\mathrm{f}} = \boldsymbol{J}^{\mathrm{T}} \left(\frac{\partial\dot{\boldsymbol{q}}}{\partial\dot{\boldsymbol{s}}}\right)^{\mathrm{T}} \boldsymbol{Q}_{\mathrm{f}} \tag{35}$$

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 $\partial\dot{\boldsymbol{q}}/\partial\dot{\boldsymbol{q}}_{\mathrm{a}}$ 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 $\boldsymbol{Q}_f$ is only possible indirectly via the projected forces $\boldsymbol{Q}_{\mathrm{a,f}}$ 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:

$$\boldsymbol{Q}_{\mathrm{a,f}} = \boldsymbol{A}_f(\boldsymbol{z}, \dot{\boldsymbol{s}})\boldsymbol{p}_f \tag{36}$$

where the friction coefficients are grouped in a corresponding parameter vector $\boldsymbol{p}_f$.
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 ${}_{(E)}\boldsymbol{r}_{\boldsymbol{B}_j}^{E} = \begin{bmatrix} r_{Bx_j} & r_{By_j} & r_{Bz_j} \end{bmatrix}^{\mathrm{T}}$ into account (see also eq. (14)). The derivation of minimal parameters is of a major interest, since they constitute the set of identifiable ones (Gautier & Khalil, 1990). Following section discusses the experimental identification of parameters and the implementation of identified inverse models in control.

	KUKA KR15	*PaLiDA*
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	$I_{3_{xx}} - I_{3_{yy}} + I_{4_{yy}} + 2l_4 s_{4_z} + (l_4^2 - l_3^2)(m_4 + m_5 + m_6)$	$I_{xx_E} + m_3 \sum_{j=1}^{6}\left(r_{By_j}^2 + r_{Bz_j}^2\right)$
p_7	$I_{3_{xy}} - l_3 s_{4_z} - l_3 l_4(m_4 + m_5 + m_6)$	$I_{yy_E} + m_3 \sum_{j=1}^{6}\left(r_{Bx_j}^2 + r_{Bz_j}^2\right)$
p_8	$I_{3_{xx}} - I_{3_{yy}} + I_{4_{yy}} + 2l_4 s_{4_z} + (l_3^2 + l_4^2)(m_4 + m_5 + m_6)$	$I_{zz_E} + m_3 \sum_{j=1}^{6}\left(r_{Bx_j}^2 + r_{By_j}^2\right)$
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*.