Chapter 11
Analysis of Industrial Robot Structure and Milling Process Interaction for Path Manipulation

J. Bauer, M. Friedmann, T. Hemker, M. Pischan, C. Reinl,
E. Abele, and O. von Stryk

Abstract. Industrial robots are used in a great variety of applications for handling, welding, assembling and milling operations. Especially for machining operations, industrial robots represent a cost-saving and flexible alternative compared to standard machine tools. Reduced pose and path accuracy, especially under process force load due to the high mechanical compliance, restrict the use of industrial robots for machining applications with high accuracy requirements. In this chapter, a method is presented to predict and compensate path deviation of robots resulting from process forces. A process force simulation based on a material removal calculation is presented. Furthermore, a rigid multi-body dynamic system's model of the robot is extended by joint elasticities and tilting effects, which are modeled by spring-damper-models at actuated and additional virtual axes. By coupling the removal simulation with the robot model the interaction of the milling process with the robot structure can be analyzed by evaluating the path deviation and surface structure. With the knowledge of interaction along the milling path a general model-based path correction strategy is introduced to significantly improve accuracy in milling operations.

11.1 Introduction

Major fields of machining applications for industrial robots are automated premachining, deburring and fettling of cast parts or trimming of carbon-fiber-reinforced laminate. Due to its kinematic structure with 6 axes the robot can cover a large working space and is able to reach difficult workpiece positions so that it can be applied to perform complex machining operations. Therefore, compared to standard machine tools, industrial robots offer an economic and flexible machining alternative. However, industrial robots do not provide a high absolute and repeat accuracy. Current industrial robot systems reach a repeat accuracy of 0.06 mm. Under process load, e. g. in milling operation, an additional deflection of the tool center point (TCP) occurs. Measured deflections of 0.25 mm under process loads of 100 N in earlier tests [1] have confirmed the expected compliance. Therefore, when using an industrial robot for milling applications, inaccuracies of the

B. Denkena and F. Hollmann (Eds.): Process Machine Interactions, LNPE, pp. 245–263.
springerlink.com © Springer-Verlag Berlin Heidelberg 2013

serial robot kinematic, the low structural stiffness and the effective process forces lead to path deviations. The results are unwanted trajectory deviations, which lead to errors in dimension and a reduced surface quality of the workpiece. These deviations mainly consist of a static offset overlaid with a low frequency oscillation of the tool [2].

In order to increase the milling accuracy with present process forces of the milling operation a model-based path manipulation module for robots is developed. This module consists of a robot and a milling force model to predict path errors in advance of a real milling operation. Additionally, the simulation module allows the investigation and analysis of the interaction of milling force and motion of the robot. Finally, the proposed strategy for modeling and simulation serves as the basis for an efficient offline correction of path deviations.

Overall Objective

The machining operation induces process forces, which act on the machine structure. The high compliance of the structure causes a deviation of the programmed path at the TCP. This path displacement causes a new cutting condition of the cutter including a variation of the process force (cf. Fig. 11.1). Thus, milling robots are characterized by a close interaction of the cutting process with the mechanical robot structure. An early consideration of the interactions of the milling robot in the production process can reduce manufacturing failures as well as efforts to manually re-teach the robot's path.

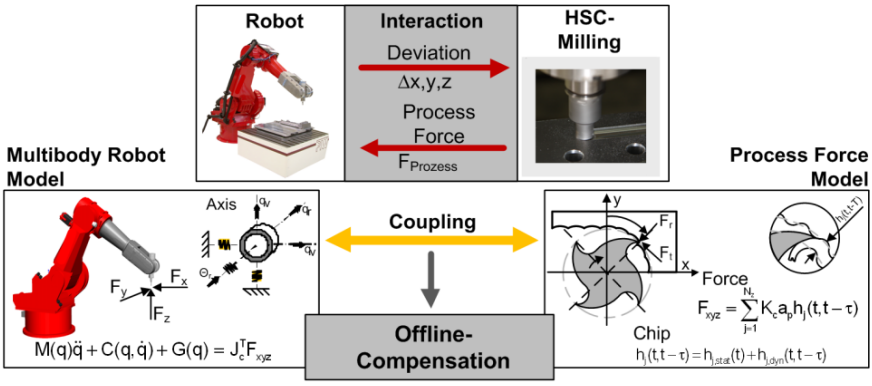

Fig. 11.1. Model based tool path compensation

The general aim is the development of a strategy to improve the accuracy in robot based milling operations by prediction and offline compensation of static and low frequency path deviations. Furthermore, the optimization of cutting parameters and a prediction of milling properties (accuracy in dimension, surface structure) is in the focus.

In order to simulate the milling force a standard cutting force model presented in [3] is implemented and adapted for industrial robots. The simulation of the

robot's motion dynamics is based on the Newton-Euler-formulation. The model is based on a fine granular description of the kinematic structure and dynamical properties. It allows the introduction of arbitrary rotational axes to model elastic deformation. Additional properties are added to consider tilting of the axes and backlash of the gears.

For both sub-models, different experimental investigations are conducted to determine the model parameters describing the physical behavior of the robot and the machining process. Both models are initially tested independently and then coupled to simulate the machine process interaction. Two methods of model-based offline path compensation - based on ideal milling forces and deviation mirroring - are presented. Finally, robot milling experiments are carried out to proof the compensation concepts.

11.2 Robot and Milling Force Modeling

In order to study the interaction of machine structure, i. e. the industrial robot and the removal process, a fine granular robot model and a milling force model are developed.

11.2.1 Extended Robot Kinematic and Dynamics Modeling

In this section, the modeling and simulation of the kinematic and dynamics of the robot's motion are discussed. Special interest is taken in the modeling of elasticities of the robot, which have a high impact on the robot's motion, if the robot operates under present process forces. The presented approach is based on a modular modeling methodology, which allows the integration of arbitrary axes of motion without the need to re-implement the equations of the robot dynamics.

Modeling the Kinematical Structure

As elastic motions of the robot may not only appear around the axes of the robots drives but also around additional axes, an extended kinematical model has been derived. In this section, the kinematic model is discussed using homogeneous transformations in a 4x4-matrix representation.

For better readability, the abbreviations

$$Trans(x,y,z) := \begin{pmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{pmatrix}, Rot(x;\alpha) := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$Rot(y;\alpha) := \begin{pmatrix} \cos(\alpha) & 0 & \sin(\alpha) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\alpha) & 0 & cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, Rot(z;\alpha) := \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

are used to represent translations and rotations. In standard Denavit-Hartenberg-notation (e. g. [4]), a rigid link with a revolute joint is described by

$$link_{dh,i} = Rot(z;\theta_i)\cdot Trans(0,0,d_i)\cdot Trans(0,0,a_i)\cdot Rot(x;\alpha_i)$$

with θ_i being the current position of the joint and the constants d_i, a_i and α_i describing the relative position of the next joint. These joints, which are driven to move the robot, are referred to as *actuated joints*. It should be noted that this convention does not contain information on the precise placement of the joint. As

$$Trans(0,0,p_i)\cdot Rot(z;\theta_i)\cdot Trans(0,0,d_i - p_i) = Rot(z;q_i)\cdot Trans(0,0,d_i)$$

is valid, joints can be anywhere on the z-axis of the previous link's coordinate frame. To allow for a precise placement of the joint the model is extended by the joint displacement p_i, leading to

$$\begin{aligned} link_{wrep,i} = \; & Trans(0,0,p_i)\cdot Rot(z;\theta_i)\cdot Trans(0,0,d_i - p_i)\cdot \\ & \cdot Trans(0,0,a_i)\cdot Rot(x;\alpha_i). \end{aligned}$$

This model is extended by two variable rotations around axes, orthogonal to the joint's axis, leading to the final extended kinematics model

$$\begin{aligned} link_{ext,i} = \; & Trans(0,0,p_i)\cdot Rot(z;\theta_i)\cdot Rot(x;\theta_{x,i})\cdot Rot(y;\theta_{y,i})\cdot Trans(0,0,d_i - p_i)\cdot \\ & \cdot Trans(0,0,a_i)\cdot Rot(x;\alpha_i) \end{aligned}$$

with $\theta_{x,i}$ and $\theta_{y,i}$ being the rotation angles caused by elasticities around the additional axes. These additional revolute axes are referred to as *virtual joints* (cf. Fig. 11.2). It should be noted that these virtual joints are optional. One or both virtual joints can be added to each link, if additional elasticities need to be modeled for this link.

Additional Dynamics Properties

Simulating the motion dynamics of the robot requires additional parameters for each link of the robot. Namely, these are the mass m_i, the inertia tensor $\mathbf{I_i}$ and the center of mass com_i for each link. In the implementation used for this work, the center of mass is described with respect to the coordinate frame defined by $link_{ext,i}$. The inertia tensor is resolved at the center of mass in a coordinate frame using the same orientation as the frame defined by the link.

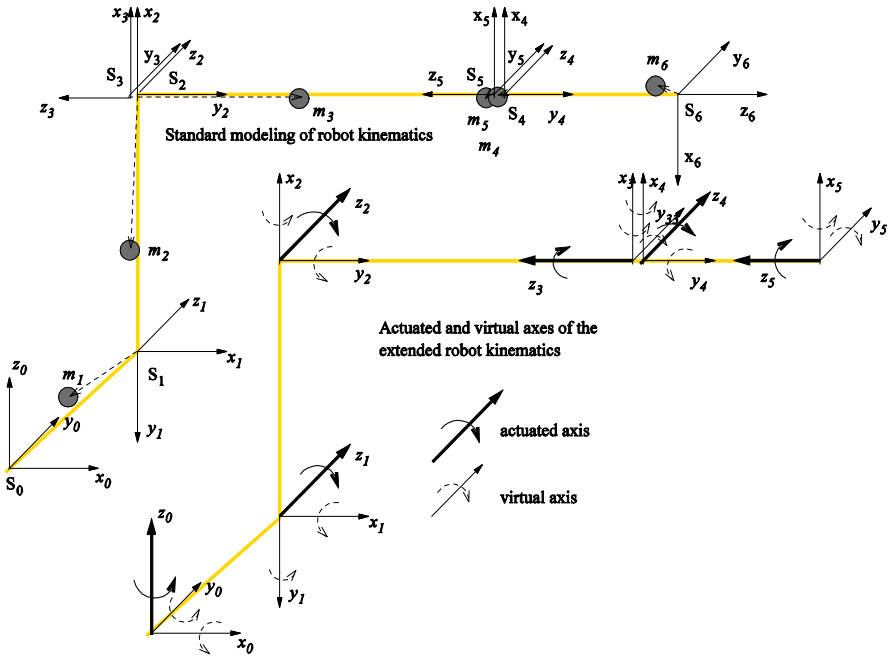

Fig. 11.2 Modeling of the robot with tilting at virtual axes

Modeling the Drives and Elasticities

Simulation of the robot's motion requires information of the torques acting in the joints (either real or virtual) of the robot. As no information of the robot's drivetrain nor the control strategy of the motors are known, the assumption is made that each motor is at its desired position q_i and moves with the desired rate $\dot{q}_i$. The motors are coupled to the joints by means of a spring-damper-system with the spring stiffness K_i, the damping D_i and s_i the backlash of the gear leading to the equation

$$\tau_i = D_i \cdot (\dot{\theta}_i - \dot{q}_i) + K_i \cdot \begin{cases} ((q_i - s_i) - \theta_i) & \text{, if } (q_i - \theta_i) \geq s_i \\ ((q_i + s_i) - \theta_i) & \text{, if } (q_i - \theta_i) \leq -s_i \\ 0 & \text{, else} \end{cases}$$

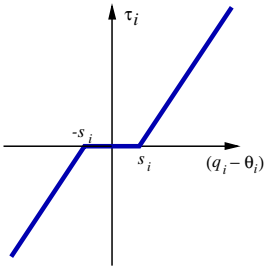

Fig. 11.3 Modeling of backlash

for the torques in the actuated joints (cf. Fig. 11.3). Note that this convention also allows for elasticities and damping around the actuated axes, which may occur because of the gears used in the drivetrain. Torques acting on virtual joints are calculated in the same way with the exception that the desired position and rate always are zero and that there is no backlash,

$$\tau_{x|y,i} = K_{x|y,i} \cdot \theta_{x|y,i} + D_{x|y,i} \cdot \dot{\theta}_{x|y,i} \ .$$

Forward Dynamics Simulation

The dynamics of a robot's motion is described by

$$\mathbf{M}(\boldsymbol{\theta})\cdot\ddot{\boldsymbol{\theta}} + \mathbf{C}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) + \mathbf{G}(\boldsymbol{\theta}) = \boldsymbol{\tau} + \mathbf{S}(\mathbf{F}_{\mathbf{xyz,tool}},\boldsymbol{\theta})$$

with $\boldsymbol{\theta} = (\theta_1, \theta_{x,1}, \theta_{y,1}, ..., \theta_n, \theta_{x,n}, \theta_{y_n})^T$ being the positions of all joints (actuated or virtual), $\boldsymbol{\tau} = (\tau_1, \tau_{x,1}, \tau_{y,1}, ..., \tau_n, \tau_{x,n}, \tau_{y_n})^T$ likewise being the torques in all joints and **S** being the torques caused by milling projected into the respective joints.

Different ways exist to calculate the mass-matrix **M**, vector of coriolis forces **C** and gravitational forces **G** in this model. The modular approach chosen describing the robot's structure is well-suited for algorithms based on the Newton-Euler-formulation of robot dynamics. Two widely used algorithms of this group are the Composite-Rigid-Body-Algorithm (CRBA, see [5]) and the Articulated-Body-Algorithm (ABA, [6]). Both algorithms yield the same solution but differ in runtime with the CRBA performing better than the ABA for systems with few joints.

Implementation

To allow the efficient simulation of the robot's motion dynamics an object-oriented framework has been developed using C++. Within this framework, the robot's structure is modeled as a chain of modeling entities consisting of the robot's base, variable and fixed rotations and rigid bodies (consisting of a fixed translation in combination with the body's mass, center of mass and inertia tensor). Additional modeling entities not used for this research include variable translations (to describe prismatic joints) and forks (to build tree-shaped structures beyond the kinematic chain). To allow for arbitrary structures these modeling entities can be combined in any order so that one is not limited to the structure described above. Similar approaches have been used successfully for the simulation of industrial robots [7], biomechanical systems [8] and autonomous mobile robots [9].

The framework provides methods to solve the robot's kinematics and dynamics equations, yielding solutions for the direct kinematics, the inverse dynamics and the forward dynamics. Currently, the forward dynamics is based on the CRBA and ABA to allow the selection of the better performing algorithm, depending on the complexity of the robot.

Due to this modular description of the robot the structure of the simulated robot can be exchanged easily without the need to derive new equations of motion. Thus, it is possible to select (and re-select) the virtual joints required for a specific milling simulation, depending on the concrete robot. For the purpose of parameter estimation and trajectory optimization the framework also allows the calculation of derivatives of the simulated robot's motion with respect to any modeling parameters.

This feature is based on the ADOL-C library [10] for automated derivation of C++-functions. Depending on the current use of the developed framework one can either use the special types provided by ADOL-C (if derivatives are required) or the standard floating-point types of the machine (no derivatives available, but

faster execution). By this feature, it is possible to use the same implementation of the model either for parameter estimation or for the simulation of the milling process as well as for the application of numerical optimal control methods [11], without changes in the source code.

11.2.2 Process Force Calculation

The calculation of milling forces is based on a material removal simulation that also calculates complex cutter workpiece engagement conditions and therefore the chip geometry. Based on the chip geometry the milling forces are calculated using a standard model presented in [3]. While other methods for cutter workpiece engagement simulations are introduced in [12, 13, 14, 15] the dexel representation used in this work enables an efficient computation of the engagement condition. Especially in applications of milling with industrial robots the accuracy of the dexel-based method is considered to be appropriate. In order to increase the calculation accuracy, [16] recommends the usage of a multi-dexel model for the representation of the workpiece (cf. Fig. 11.4).

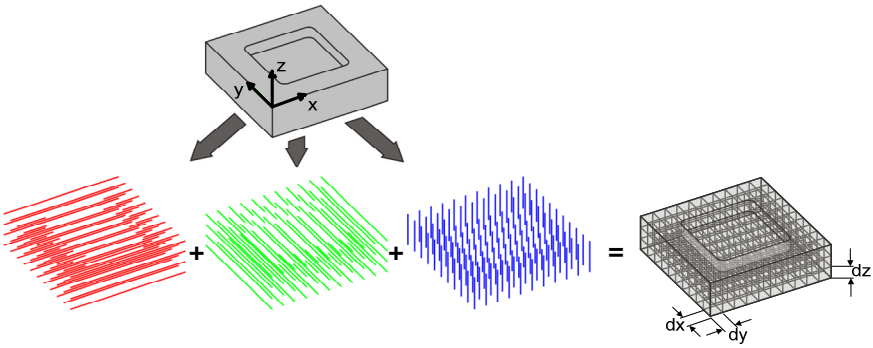

Fig. 11.4 Dexel representation of a workpiece in three directions in space

Thereby, the description of a single dexel is made by line equations,

$$\mathbf{p} = \mathbf{d}t + \mathbf{s}$$

Here, $\boldsymbol{p}$ is a point on the line, $\boldsymbol{d}$ is the line direction, t is the scaling factor and s the starting point in space. The workpiece is modeled using a multi-dexel representation compromising dexel directions in x, y and z. The discretization of the workpiece is defined by the dexel distances dx, dy and dz. In order to receive a sufficient accuracy of the force calculation the relation k_d between the dexel discretization and the tool radius R is considered,

$$k_d = \frac{dx}{R} = \frac{dy}{R} = \frac{dz}{R} = 0.05 \ .$$

During a simulated milling operation the tool moves through the workpiece in discrete time steps t_i. The incremental size of the time step is dt.

$$dt = \frac{dt_{tooth2tooth}}{n_{int}}.$$

In every time step $t_i=1,\ldots, n_{int}$ the material removal is calculated, where $dt_{tooth2tooth}$ is the tooth-passing time calculated by

$$dt_{tooth2tooth} = \frac{60}{N_z \cdot n}$$

with the number of revolutions n and the number of teeth N_z.

11.2.2.1 Calculation of the Chip Geometry

The material removal simulation calculates a point cloud representing the outer volume of a single chip, which the discrete chip geometry is extracted from. The chip geometry is described by the angular discretized chip thickness $h(\varphi,z)$ with the entry and exit angles φ_{in}, φ_{out}. As the chip thickness varies along the cutting edge the chip gets subdivided into discs of the height dz and in $d\varphi$ in angular direction (cf. Fig. 11.5).

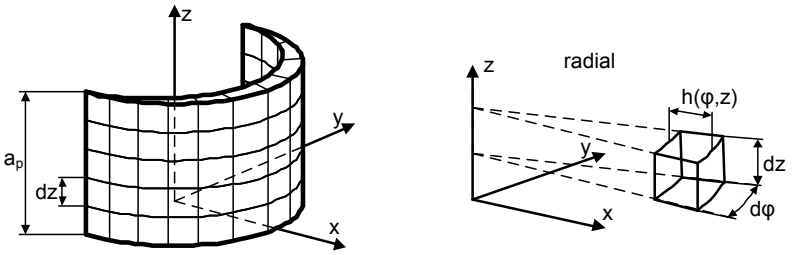

Fig. 11.5 Discretized chip geometry

The chip thickness is calculated in three steps. In the first step, the set of points is subdivided into disks of the height dz. As a next step, inside each disc the maximum angular points are defined as restricting points by φ_{in} and φ_{out} (cf. Fig. 11.6a). Due to the complex contact conditions and the complex chip geometries the entry and exit angles vary over the depth of cut a_p. The cross-sectional geometry of each disk is calculated based on the cylindrical form of the cutter at the discrete time of t_i and t_{i-1}. The area between the input angle φ_{in} and exit angle φ_{out} is discretized into multiple sub-areas with the angular distance $d\varphi$. The value of the discrete chip thickness $h(\varphi,z)$ with respect to φ is calculated by a line intersection with the two cylinders in the final step (cf. Fig. 11.6b).

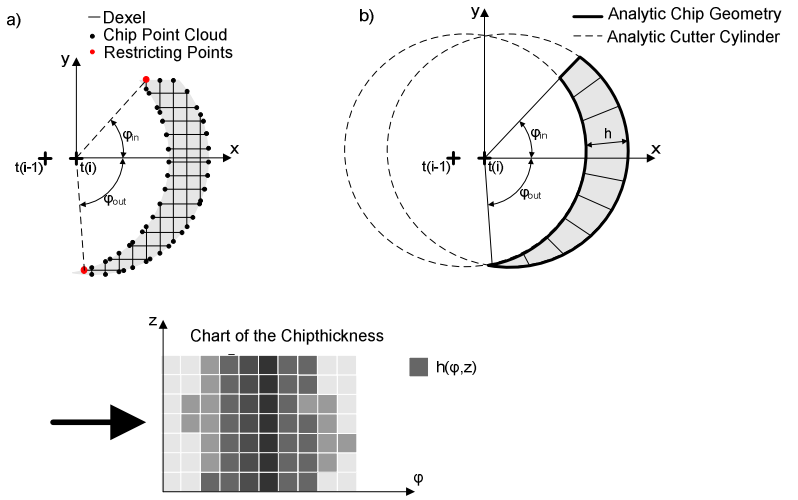

Fig. 11. 6 Calculation of the chip thickness *h(φ,z)*

Adding all disc levels together, a chart of the chip thickness *h(φ,z)* is extracted, which forms the base of the force calculation. This is the most frequently performed arithmetic operation during the removal simulation.

11.2.2.2 Cutting Force Model

For the prediction of the cutting forces a standard cutting force model based on Altintas [3] is implemented, where the time delay responsible for the chatter is neglected. According to the chip discretization the cutter is sectioned into discs of height *dz* (cf. Fig. 11.7).

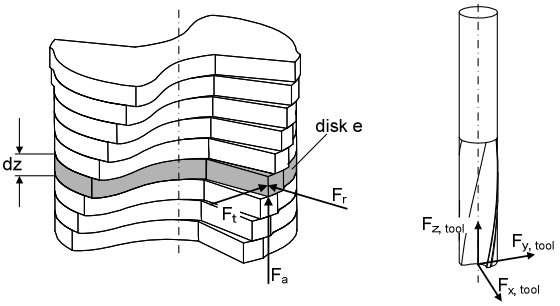

Fig. 11.7 Cutting force calculation

In each disc *e*, $\boldsymbol{F}_{rta,j}$ represents the force per tooth *j* in radial, tangential and axial direction,

$$\mathbf{F_{rta,j,e}} = \mathbf{K_c} \cdot dz \cdot h_j(\varphi, z) + \mathbf{K_e} \cdot dz \,.$$

Depending on the angular position of the tooth j of a disc, the corresponding chip thickness $h_j(\varphi,z)$ is inserted. The cutting force coefficients $\boldsymbol{K}_c=[K_{rc}, K_{tc}, K_{ac}]$ and $\boldsymbol{K}_e=[K_{re}, K_{te}, K_{ae}]$ need to be identified in advance.

A transformation of $\boldsymbol{F}_{rta,j,e}$ with $\boldsymbol{T}(\varphi)$ and the subsequent summation over all teeth N_z and discs N_e results in the process force $\boldsymbol{F}_{xyz,tool}$, given in a non-rotating tool coordinate frame,

$$\mathbf{F}_{\mathbf{xyz,tool}} = \sum_{e=1}^{N_e}\sum_{j=1}^{N_z} \mathbf{T}_{\mathbf{j}}(\varphi)\cdot\mathbf{F}_{\mathbf{rta,j,e}} \,.$$

The force $\boldsymbol{F}_{xyz,tool}$ acts at the TCP of the robot and thus represents the interface to the robot model.

11.3 Analysis of the Mechanical Robot Structure

In order to identify the mechanical model parameters of the robot several experiments are performed including

- modal analysis,
- stiffness measurement within the working volume of the robot,
- stiffness measurement of the structural robot components.

In the following sub-sections, the investigations are presented and the consequential model adaptations are implemented.

11.3.1 Modal Analysis

The modal analysis is used to determine eigenfrequencies, eigenforms and modal damping of the robot structure. The robot structure is excited by a hammer impulse at a defined position. The response is recorded with a tri-axial acceleration sensor, which the transfer functions are derived from for all 109 measuring points. Figure 11.8 shows the robot structure and the measuring grid.

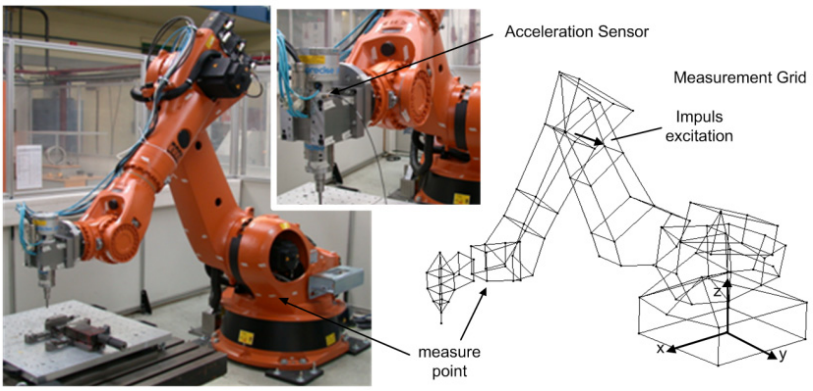

Fig. 11.8 Modal analysis of the robot

Based on the transfer functions, the eigenforms, eigenfrequencies and modal damping is extracted. The robot KUKA KR 210 [17] has six dominant eigenfrequencies below 100 Hz. In Table 11.1, these frequencies are summarized and explained briefly.

Table 11.1 Eigenfrequencies and eigenforms of the robot

Eigenform	Eigenfrequency [Hz]	Modal Damping [%]	Description
1	8.4	6.37	Oscillation about axis 1
2	11.1	1.02	Tilting of axis 1 around *y*
3	16.9	0.75	Tilting of axis 1 around *y*
4	20.6	0.16	Oscillation about axis 3
5	24.1	0.11	Oscillation about axis 2 and 3 plus deformation of fork (axis 5)
6	57.9	0.64	Tilting of axis 1, 2 and 3 plus torsion of the whole structure

11.3.2 Static Stiffness within Working Space

Measuring of the static stiffness in the working space is performed in *x*, *y*, and *z*-direction at nine positions within the relevant working area of the robot. Static forces F_x, F_y, and F_z are applied by a force measurement rod. Laser distance sensors are mounted to an external measurement frame to capture the robot's displacements Δx, Δy and Δz (cf. Fig. 11.9). The static stiffness k_{xx}, k_{yy}, k_{zz} is calculated by

$$F_x = k_{xx}\Delta x, \quad F_y = k_{yy}\Delta y, \quad F_z = k_{zz}\Delta z\,.$$

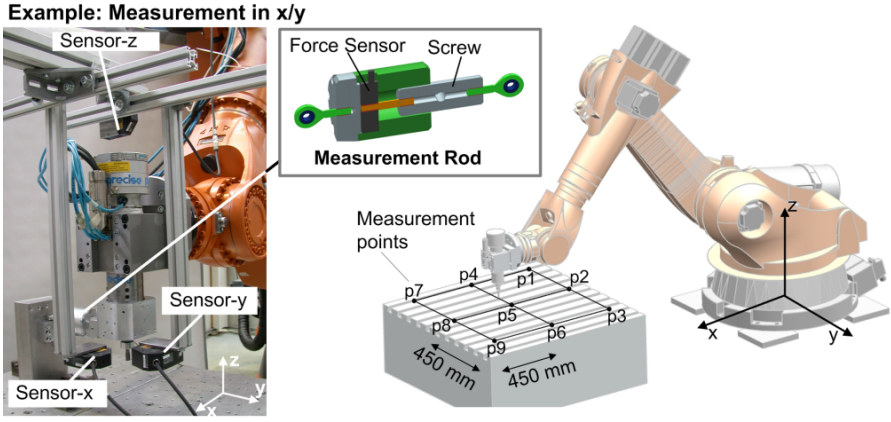

Fig. 11.9 Test-rig of the stiffness measurement

While evaluating the stiffness at a single measuring point by 3 cycles of tensile and compressive force of $F \in [-1500\,,\,1500]\,\mathrm{N}$, the robot's pose remains fixed. Figure 11.10 illustrates the robot's compliance - calculated by $h_{xx}=(k_{xx})^{-1}$, $h_{yy}=(k_{yy})^{-1}$ *and* $h_{zz}=(k_{zz})^{-1}$ - at each point of the measuring grid. A global trend in all directions towards a higher compliance is observed at those measuring points, which have a larger distance to the robot base.

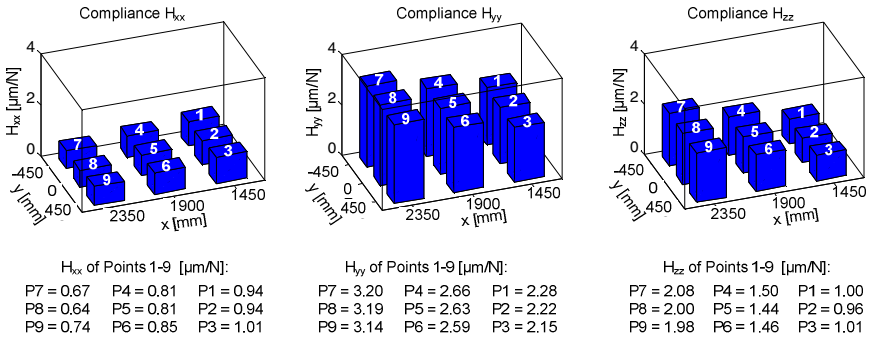

Fig. 11.10 Direct compliance within the working area (z = 900 mm)

In a similar way, the rotational and tilting stiffness are experimentally measured. While the force load is applied in defined directions at the spindle the distance sensors measure the tilting and rotation at each axis sequentially. In Table 11.2, the measured stiffnesses are summarized.

Table 11.2 Rotational robot stiffness at axes 1 – 6

Mech. Robot Component	Rotational Stiffness [Nm/rad]
Axis 1	8,937e-6
Axis 2	6,343e-6
Axis 3	2,357e-5
Axis 4	7,804e-5
Axis 5	1,047e-4
Axis 6	2,052e-4

11.3.3 Parameter Identification

For a full parameterization of the robot model, mass, stiffness, and damping matrices **M**, **K** and **D** need to be determined. The model adjustment is split into the static and the dynamic adaption. Within the static model adjustment the measured component stiffnesses (Sect. 11.3.2) are deployed in the robot model. The virtual robot is positioned at points p1 to p9 corresponding to the experiment and a static load is applied in the simulation. Hence, the deviations are simulated and the stiffness of the full mechanical structure is determined. By adjusting the model's

rotational and tilting stiffness values in order to compromise the measured stiffness, the model gets empirical updated. The simulated and the measured stiffnesses within the working area are compared in Figure 11.11.

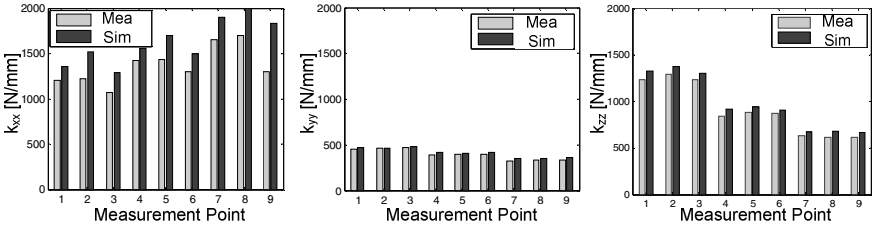

Fig. 11.11 Comparison of simulated and measured working area stiffness

For the model dynamics adaption, the mass and damping matrix have to be determined. The mass and the location of the centers of mass for each link are extracted from the technical documentation. The inertia tensors of all rigid links are derived from CAD-representations. By calculating the frequency response functions (FRF) of the virtual robot compliance H_{xx}, H_{yy}, and H_{zz} and comparing them to the corresponding robot's FRF, the dynamical correlations of the robot model are evaluated (cf. Fig. 11.12). In order to improve correlation, the mass and stiffness model parameters are slightly adjusted. Finally, the peak height of selected eigenfrequencies is adjusted by modifying the damping parameters to confirm the measured FRFs.

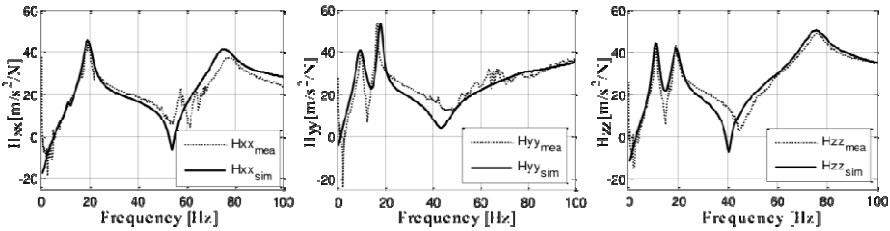

Fig. 11.12 Frequency response function of the compliance H

After updating the model parameters, the simulated robot compliance results in a reasonable correlation to the measured compliance frequency response function in the frequency band of $f = 0$ to 50 Hz. Especially the frequency and amplitude of the resonance peaks correlate well.

11.3.4 Simulation and Validation

The validation of the process force model is carried out separately from the robot model. Figure 11.13 shows results of machining a test workpiece of aluminum 3.1325 with a portal robot. The used cutter is a two-fluted poly-crystalline cutter with $d = 10$ mm. The milling parameters are $n = 8000$ rpm, $v_f = 50$ mm/s, $a_p = 1$ mm.

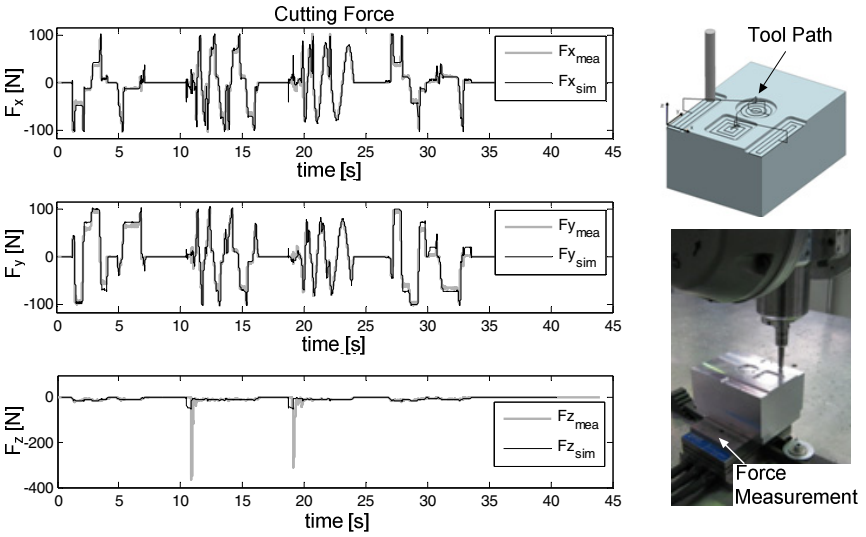

Fig. 11.13 Validation of the process force model

The cutting forces are recorded by a Kistler force dynamometer during the machining operation. In order to calculate the mean cutting forces, the force signal is filtered by a low-pass filter with a cut-off frequency of f_{corner} = 100 Hz. As the main eigenfrequencies of the robot structure are lower than 60 Hz (cf. Table 11.1) the consideration of frequency bands above 100 Hz are neglected for coupled robot milling simulations. The comparison of measured and simulated cutting forces F_x, F_y, F_z (Fig. 11.13) shows a sufficient congruence of an overall quadratic mean below 20 % of each force direction.

11.4 Model-Based Compensation

The two compensation strategies are characterized by a model-based approach using ideal milling forces (Sect. 11.4.1) and by geometrically mirroring the deviation path (Sect. 11.4.2). The aim is the offline correction of static and low-frequency deviations before the real milling process is started. Furthermore, this allows an improvement of machining parameters as well as estimations concerning dimensional accuracy or surface quality.

11.4.1 Compensation by Ideal Milling Forces

The proposed model-based compensation strategy consists of three steps: (1) A simulation run assuming an idealized robot, (2) the computation of reference joint positions and (3) a transformation into compensating track points.

Assuming a known ideal TCP trajectory, this idealized path is initially simulated without considering that elasticities in the robot's dynamics, i. e. real joint

positions, match the ideal set values ($q_{real} = q_{set}$ and $\dot{q}_{real} = \dot{q}_{set}$) for virtual and actuated joints. Thus, a path through a sequence of points on the workpiece is simulated using the robots kinematics. This generates corresponding trajectories of the process forces at the TCP. As the computation is based on a discretization of the workpiece and an infinitesimal stepsize Δt is deployed in an accurate simulation, force signals contain significant noise components. Therefore, low-pass filtering is applied to get smooth trajectories (cf. Fig. 11.14) for further use.

Using the recursive Newton-Euler-method [4], the inverse dynamics is solved to compute idealized torques $\tau_{ideal}(t_k)$ for a selection of ideal joint positions $\{q_{ideal}(t_k) \mid k = 1, \ldots, k_{max}\}$. Then, with τ_{ideal} and with the assumption $\dot{q}_{comp} = \dot{q}_{ideal}$,

$$\mathbf{K} \cdot (q_{comp} - q_{ideal}) + \mathbf{D} \cdot (\dot{q}_{comp} - \dot{q}_{ideal}) = \tau_{ideal}$$

is solved for each t_k as a linear equation system. This results in compensational joint positions q_{comp}, which includes virtual joints as well as the actuated ones.

As the robot input requires for certain positions $p_{comp}(t_k)$, the calculated compensational joint positions $q_{comp}(t_k)$ are transformed into base coordinates by solving the direct-kinematics for each t_k. Thus, the deviation according to the stiffness of actuated and virtual joints is considered in the resulting sequence of compensating set points $\{q_{comp}(t_k) \mid k = 1, \ldots, k_{max}\}$.

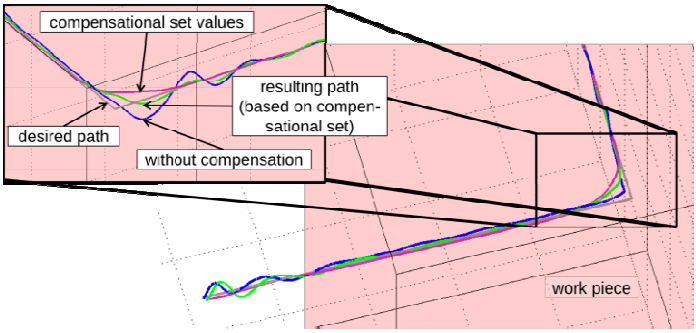

Fig. 11.14 Deviation, correction and resulting path in simulation

The proposed model-based strategy is computed very efficiently as only one simulation run with an idealized, kinematic robot model is performed. The eventual solution of one linear equation system per interpolation point can be carried out in real-time.

11.4.2 Compensation by Deviation Mirroring

The second approach of path compensation is based on a geometrical consideration of deviations caused by milling forces acting on the cutter (cf. Fig. 11.15(1.)). The compensation strategy is applied in radial and axial direction of the cutter.

Figure 11.15(2.) illustrates the idea of calculating a new NC-code, based on mirroring path points on the planned NC-path. Thus, when the milling operation is conducted with the modified NC-code, a coincidence of planned and the resulting TCP position is expected (cf. Fig. 11.15(3.)).

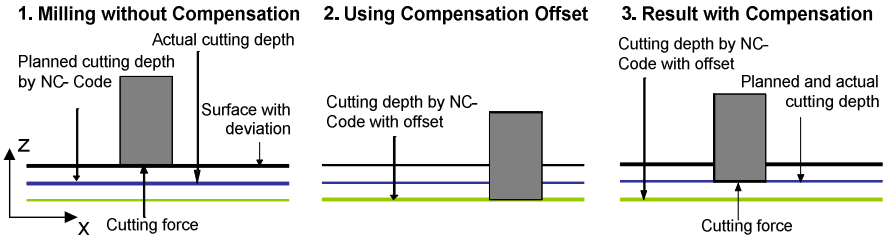

Fig. 11.15 Concept of path compensation by mirroring the cutter position.

To calculate a compensated path the machining operation is initially simulated using the original NC-code. Thus, corresponding deviations of the TCP become available and the offset values for selected points on the deviated path to corresponding NC-path points are calculated. By mirroring these offsets at the planned, ideal NC-path the compensational path points are computed. In order to increase the overall path accuracy additional path points may be inserted into the original NC-path for compensation.

11.4.3 Experimental Investigation

During the experiment a milling path describing a 90-degree corner is machined. Within this test, static deviation and dipping in the corner are investigated. Based on the ideal milling path the simulation initially returns deviations of 0.35 mm in *y* and 0.06 mm in *x*-direction. The compensational NC-path is generated by mirroring certain points on the ideal NC-path. The simulation with this compensational NC-path shows a significant reduction of the path deviation under present process force (cf. Fig. 11.16).

To validate these results from simulation a low feed rate of 1.5 mm/s and a milling depth of a_p = 0.5 mm is initially chosen for the experiments as a full slotting operation. This setting allows the neglecting of process forces and the manufacturing of a workpiece without recognizable deviations. Subsequently, a milling operation with a feed rate of $v_f = 50$ mm/s and $a_p = 1.5$ mm is performed. Due to the process forces the TCP deflects and the resulting deviation at the workpiece is measured by a microscope relative to the previous reference operation. Deviations between *0.03* mm and *0.44* mm are measured on 7 positions along the path (cf. Fig. 11.17). In the corner, a large difference of *1.69* mm between the planned and the resulting path is measured due to path blending effects. The root mean square error of all seven measured points equals $e_{rms} = 0.72$mm.

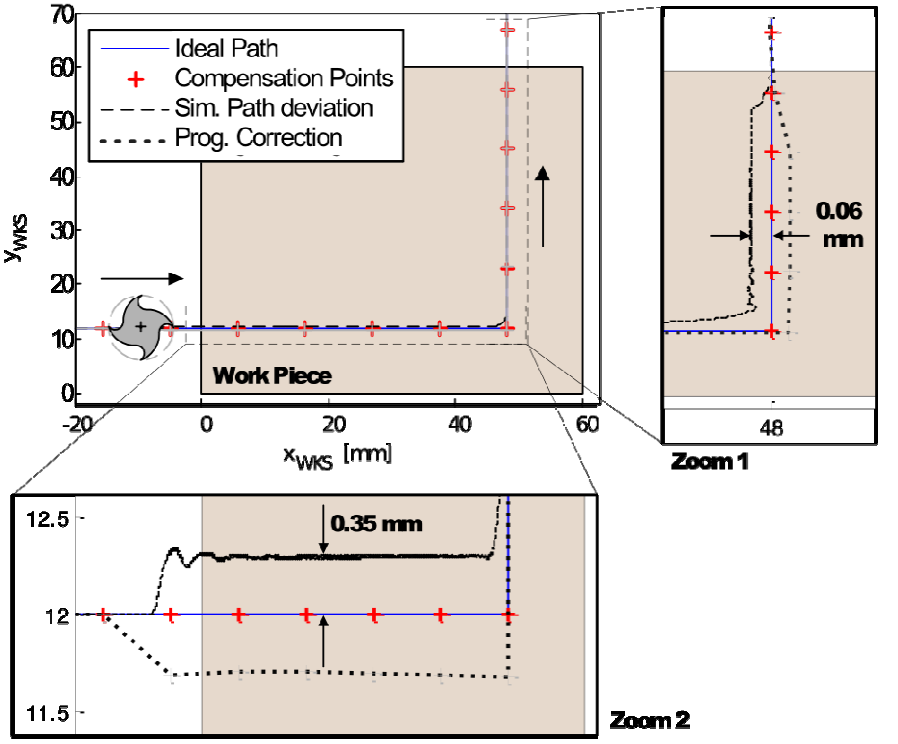

Fig. 11.16 Simulation of path deviation and robot program modification

The experimental milling operation is repeated with the modified robot program including the compensating path points. The deviation is reduced primarily within the strait path segment. The root mean square error of $e_{rms} = 0.72$ mm is now reduced to $e_{rms} = 0.15$ mm. In the corner, only a moderate error reduction was achieved because of the robot's internally-controlled path blending.

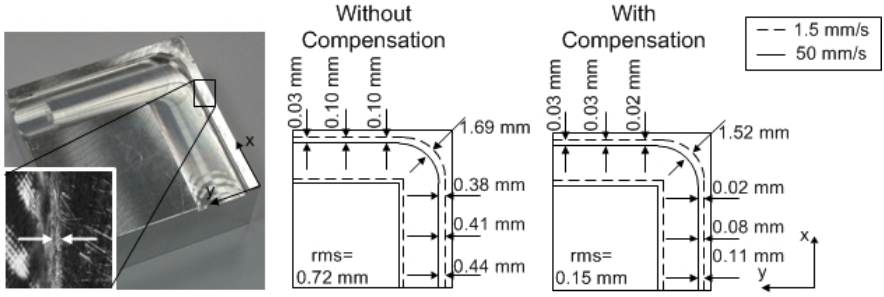

Fig. 11.17 Test workpiece without/with compensation

In z-direction, the profile of the machined part with and without compensation is measured on a coordinate measuring machine. The compensated path is slightly closer to the supposed NC-path on the straight path segments. In the corner, the dipping of the cutting tool is reduced by $\Delta z = 0.029$ mm (cf. Fig. 11.18).

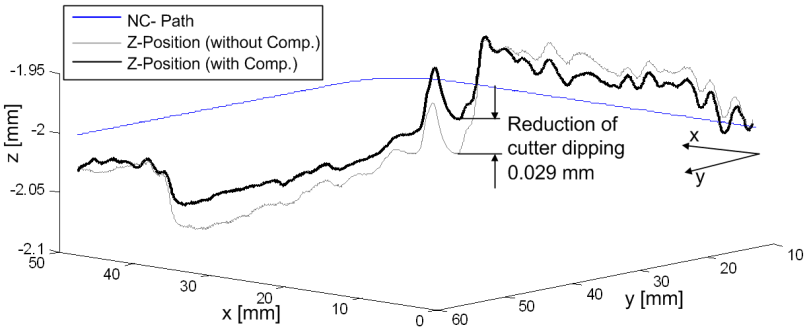

Fig. 11.18 Z-deviation of the machined workpiece

The main reason for only achieving a moderate compensation in z-direction is related to the relatively low forces acting on the cutter in z-direction. While mean forces $F_{x,1} = -130$ N, $F_{y,1} = 58$ N in the first and $F_{x,2} = 108$ N, $F_{y,2} = 59$ N in the second straight path segment were observed, the z-force is only $F_{z,1} = 24$ N and $F_{z,2} = 39$ N. The simulated mean force in both segments is $F_z = 35$ N, which the compensation is calculated on. Thus, the path correction off-set based on the calculated deviation is small compared to x and y- direction. Furthermore, the internal numerical control of the available KUKA robot additionally causes path errors, which cannot be described by the proposed robot model. This effect is even more dominant when only a small deviation appears due to the low force level.

11.5 Conclusion

Industrial robots are a cost-saving and flexible alternative for machining applications in a low-accuracy area. A reduced position and path accuracy, especially under process force load due to the high mechanical compliance, restricts the use of industrial robots for further machining applications.

The proposed strategy is built on a coupled simulation module consisting of a process force model and a robot model. This validated simulation enables the prediction of a force-induced path deviation in robot milling operations. Experiments have shown that a milling error occurs as a summation of robot control-contouring error and the process force deviation of up to *0.44* mm. Depending on the control system and the level of cutting force, the presented approach compensates the error caused by process force deviation to a large extend. The influence of the control-contouring error on the overall path deviation is more dominant in the case of only small process forces. Since this control-contouring error could not be covered in the simulation by the so-far implemented robot model, the path correction

approach results in only moderate improvements for this case of small process forces. For further improvements, these effects of the robot's internal path planning and control algorithms should be identified and implemented into model.

However, in the case of dominant process forces the path compensation concept is able to reduce the milling error significantly.

Acknowledgments. The authors thank Max Stelzer, who carried out very helpful proofs of concept in the first phase of the project.

References

[1] Abele, E., Bauer, J., Rothenbücher, S., Stelzer, M., Stryk, O.: Prediction of the Tool Displacement by Coupled Models of the Compliant Industrial Robot and the Milling Process. In: Conference on Process Machine Interaction, Hannover (2008)

[2] Weigold, M.: Kompensation der Werkzeugabdrängung bei der spanenden Bearbeitung mit Industrierobotern. PhD Thesis TU Darmstadt (2008) ISBN 978-3-8322-7178-7

[3] Altintas, Y.: Manufacturing Automation: Metal Cutting Mechanics, Machine Tool Vibrations and CNC Design. Cambridge University Press (2000)

[4] Craig, J.J.: Robotics. Addison-Wesley (1989)

[5] Walker, M.W., Orin, D.E.: Efficient Dynamics Computer Simulation of Robotic Mechanisms. Journal of Dynamic Systems, Measurement, and Control 104, 205–211 (1982)

[6] Featherstone, R.: Rigid Body Dynamics Algorithms. Springer (2008)

[7] Höpler, R.: A unifying object-oriented methodology to consolidate multibody dynamics computations in robot control, PhD Thesis TU Darmstadt (2004)

[8] Stelzer, M.: Forward Dynamics Simulation and Optimization of Walking Robots and Humans, PhD Thesis TU Darmstadt (2007)

[9] Friedmann, M.: Simulation of Autonomous Robot Teams with Adaptable Levels of Abstraction, PhD Thesis TU Darmstadt (2009)

[10] Walther, A., Griewank, A.: ADOL-C: A Package for the Automatic Differentiation of Algorithms Written in C/C++. Version 2.1.12, Documentation, `https://projects.coin-or.org/ADOL-C`

[11] von Stryk, O.: User's Guide for DIRCOL (Version 2.1): a direct collocation method for the numerical solution of optimal control problems, TU Darmstadt (2000), `http://www.sim.informatik.tu-darmstadt.de/sw/dircol`

[12] Rehling, S.: Technologische Erweiterung der Simulation von NC-Fertigungsprozessen, PhD Thesis Universität Hannover (2009)

[13] Surmann, T.: Geometrisch-physikalische Simulation der Prozessdynamik für das fünfachsige Fräsen von Freiformflächen, PhD Thesis TU Dortmund (2005)

[14] Stautner, M.: Simulation und Optimierung der mehrachsigen Fräsbearbeitung, PhD Thesis TU Dortmund, München (2002) ISBN 3-8027-8732-3

[15] Selle, J.: Technologiebasierte Fehlerkorrektur für das NC-Schlichtfräsen, PhD Thesis Universität Hannover (2005)

[16] Weinert, K., Müller, H., Kreis, W., Surmann, T., Ayasse, J., Schüppstuhl, T., Kneupner, K.: Diskrete Werkstückmodellierung zur Simulation von Zerspanprozessen. In: Zeitschrift für wirtschaftlichen Fabrikbetrieb: ZWF, München, pp. 385–389 (2002) ISSN 0932-0482

[17] KUKA Roboter GmbH. Data sheet KR 210-2. Gersthofen - Germany (July 2009), `http://www.kuka-robotics.com`