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



122,000

135M



Our authors are among the

TOP 1%





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



Kinematic Modeling, Linearization and First-Order Error Analysis

Andreas Pott⁺ and Manfred Hiller[‡]

† Fraunhofer Institute for Manufacturing Engineering and Automation, Stuttgart ‡ Chair of Mechatronics, University of Duisburg-Essen, Germany

1. Introduction

The kinematic analysis of parallel kinematic machines (PKM) is a challenging field, since PKM are complex multi-body systems involving a couple of closed kinematic loops. It is well-known that the forward kinematic function has in general no closed-form solution, and that up to 40 different real solutions may exist for general geometry (Husty, 1996; Dietmaier, 1998). Therefore, an efficient and handy method is needed in practise, e.g. for design, simulation, control, and calibration.

The analysis of manufacturing and assembly errors of manipulators is a topic that is highly relevant for practical applications because the magnitude of these errors is directly coupled to the total cost of production of the manipulator. In this setting, there exist intensive studies on how to estimate the error of certain moving points, e.g. the tool center point, in terms of the errors in the components of the mechanism (Brisan et al., 2002; Jelenkovic & Budin, 2002; Kim & Choi, 2000; Song et al., 1999; Zhao et al., 2002), as well as how to allocate cost-optimal tolerances to a mechanism (Chase et al., 1990; Ji et al., 2000). In this paper, an approach to estimate the first-order influence of geometric errors on target quantities is suggested in which linearization is performed by considering the force transmission of the manipulator. This enables one to obtain a comprehensive model of linearized geometric sensitivities at a low computational cost.

Error analysis for serial manipulators is relatively easy because one can establish an analytical expression for the forward kinematics which maps the generalized joint and link coordinates to the spatial displacements of the end-effector. There are numerous methods to parameterize the forward kinematics, where the approach of Denavit and Hartenberg (1955) is the most popular one. Once one has a closed-form expression for the forward kinematics, one can take derivatives of it (with respect to the geometric parameters one is interested in) and use these as sensitivity coefficients. In general, one introduces the sensitivity parameters in such a way that they vanish at the nominal configuration. This is always possible by introducing corresponding constant offsets where necessary.

For example, consider a robot involving a universal joint, and assume that the sensitivity to errors in the fulfilment of the intersection property of the axes is to be analyzed. This can be done by adding a parameter for the normal distance between the joint axes which is zero in the nominal design, and with respect to which the partial derivative will yield the sought sensitivity. However, such a method for sensitivity analysis results in a model with a Source: Parallel Manipulators, Towards New Applications, Book edited by: Huapeng Wu, ISBN 978-3-902613-40-0, pp. 506, April 2008, I-Tech Education and Publishing, Vienna, Austria

significant overhead. Examples of such models for joints are presented (Brisan et al., 2002; Song et al., 1999). Some force-based methods for clearance analysis were introduced, which are similar to the approach in this paper (Innocenti, 1999; Innocenti, 2002; Parenti-Castelli & Venanzi, 2002; Parenti-Castelli & Venanzi, 2005).

A linearization method for complex mechanisms using the kinetostatic dualism and the concept of *kinematical differentials* to efficiently set up the equations of motion of multi-body systems has been proposed (Kecskeméthy & Hiller, 1994). Using this method, all required partial derivatives can be described solely by using the kinematic transmission functions for position and velocity, as well as the force transmission function of the system. Based on these transmission functions, an algorithm is formulated for generating the Jacobian matrix and the equations of motion through multiple evaluations of the kinematic transmission functions for certain pseudo input velocities and accelerations. The corresponding algorithms are denoted as kinematical differentials for the case of the pure kinematic transmission function (Hiller & Kecskeméthy, 1989) and kinetostatic approach for the case of use of force transmission (Kecskeméthy, 1994). Later, Lenord et al. (2003) showed that kinematical differentials may be applied also to more general interdisciplinary systems which also involve hydraulic components by using an exact linearization through the kinematical differentials for the determination of the velocity linearization and numerical differentiation for the calculation of the stiffness matrix of the hybrid system. Other authors studied the determination of the stiffness matrix for complex multi-body systems using explicit symbolic derivatives (El-Khasawneh & Ferreira, 1998; Rebeck & Zhang, 1999), taking into account the stiffness of the actuators and the stiffness of special components. These approaches however require numerous computational steps when many sensitivity parameters are involved.

2. Kinematic modeling of parallel kinematic machines

2.1 Kinematic delimitation and geometry

In order to study a wide range of machine types, a generic approach for the modeling of PKM is proposed (Pott, 2007). Since PKMs tend to be symmetric and different types of PKM have similar components a modular design is used. In a first step the machine is divided into three types of components: frames, platforms and legs (Fig. 1), which form the modules of the kinetostatic code.



Fig. 1. Platform, legs, and machine frame modules of a generic six-degree-of-freedom parallel kinematic machine

The machine frame defines the position and orientation of six pivot points A_i . The mobile platform introduces the position of six pivot point B_i . Furthermore, the platform defines the parameterization of the six-degrees-of-freedom (dof) of spatial motion at the tool center point (TCP). Finally, different types of legs are introduced which mainly determine the kinematic behaviour. The most common legs for PKMs are PUS, UPS and RUS structures each consisting of an actuated prismatic (P) or revolute (R) joint as well as a pair of a universal (U) and a spherical (S) joints. Each of these structures can be described by one scalar constraint, as it is shown hereafter.



Fig. 2. Generic model of a spatial six-degree-of-freedom parallel kinematic machine

Each legs considered in this paper possess a pair of joints formed by a universal joint and a spherical joint. For the analysis of the closed-kinematic chains, these joints are known as characteristic pair of joints (Woernle, 1988). One can remove both of these joints and replace this partial chain by one nonlinear scalar constraint. This constraint describes the geometrical distance between the center of the universal joint and the center of the spherical joint for the i-th leg as

$$\mathbf{a}_{i} + \mathbf{l}_{i} = \mathbf{r} + \mathbf{R}\mathbf{b}_{i} , \qquad (1)$$

where \mathbf{a}_i denotes the position vector of the pivot point on the base and \mathbf{b}_i is the relative position of the pivot point with respect to the coordinate system fixed to the platform. The Cartesian position and orientation of the platform frame K_{TCP} is given by the vector \mathbf{r} and the orthogonal matrix \mathbf{R} , respectively. The vector \mathbf{l}_i denotes the length of the leg. Solving Eq. (1) for the magnitude \mathbf{l}_i^2 of the vector \mathbf{l}_i yields the system of six nonlinear constraints

$$(\mathbf{a}_i - \mathbf{r} - \mathbf{R}\mathbf{b}_i)^2 - \mathbf{l}_i = 0$$
 $i=1,...,6.$ (2)

The world coordinates **y** consist of a parameterization of the position **r** and the orientation matrix **R**. The geometry of the machine is expressed by the vectors \mathbf{a}_i , \mathbf{b}_i and \mathbf{l}_i . In the following sections the definition of these vectors is introduced depending on the generalized coordinate q_i of the six actuators and the kinematic structure of the basic types of legs for parallel kinematic machines.

The UPS legs are used in the Stewart-Gough-platforms which are often applied for motion simulators of cars and aircrafts. The prismatic joint is actuated as linear actuator, e.g. by a

linear direct drive, ball bearing screw, hydraulic/pneumatic cylinder. For these mechanisms the pivot points A_i on the base

$$\mathbf{a}_{i} = \mathbf{c}_{i} \tag{3}$$

are determined by the machine frame and fixed to a given position c_i . The length of the strut can be controlled by the drive through

$$l_i = q_i + q_o \tag{4}$$

The PUS leg results from changing the order of the joints within the UPS leg. The universalspherical pair encloses a leg of constant length while the proximal pivot point is actuated along a line. PUS legs are the basic leg components for Hexaglide, Linaglide and Linapod PKM. They are described by the position vector \mathbf{c}_i and the direction \mathbf{u}_i . Thus, the position of the proximal pivot point A_i is defined as

$$\mathbf{a}_{i} = \mathbf{c}_{i} + q_{i} \mathbf{u}_{i} , \qquad (5)$$

where q_i is the generalized coordinate of the drive. The length l_i =const of the strut is given by design.

Finally, the kinematics of the RUS leg is considered which is the basis of the Delta-robot (Clavel, 1988). In contrast to the PUS leg, the proximal pivot points A_i of RUS legs move a circle defined by its center c_i , an axis of rotation u_i , and a lever v_i which is given in its initial position. Thus, it holds for the point A_i

$$\mathbf{a}_{i} = \mathbf{c}_{i} + \mathbf{T}(\mathbf{u}_{i}, q_{i}) \mathbf{v}_{i}$$
(6)

where $\mathbf{T}(\mathbf{u}, q)$ is the rotation matrix for the axis \mathbf{u} and the angle q as it can be calculated by Rodrigues formula. Again, for RUS legs the length of the strut l_i =const is given by design.



Fig. 3. Geometric parameters of the leg-types under consideration

www.intechopen.com

where q_0 is a constant offset.

2.2 Review of the kinetostatic method

In the sequel, a short introduction to the kinetostatic method (Kecskeméthy, 1993; Kecskeméthy, 1994) is given as a basis for the herewith presented linearization procedure. Below, the basic equations of the kinetostatic transmission element are reviewed for better reference in this paper. Details can be found in the cited papers. In the kinetostatic formalism, mechanical components are modeled as transmission elements (Fig. 4) that map the kinematic state \mathbf{q} , $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$ at the input to the kinematic state \mathbf{q}' , $\dot{\mathbf{q}}'$, $\ddot{\mathbf{q}}'$ at the output and the associated generalized forces \mathbf{Q}' at the output to generalized forces \mathbf{Q} at the input. The kinetostatic state is composed of position, velocity, acceleration, and force. A mechanism is divided into joints and links which transmit the state from one set of coordinate frames K_i and scalar variables β_i to another set { K_i', β_i }. This concept allows one to model serial manipulators as chains of transmission elements.



Fig. 4. General kinetostatic transmission element

Interestingly, the motion of closed kinematic loops can also be represented with the help of transmission functions. For simple kinematic loops with one dof, this may be done in an explicit form. In general, however, one has to employ iterative methods to solve the loops. Nevertheless, in both cases one is able to compute the transmission function for position, velocity, acceleration, and force.

Assuming that the position transmission function is given as $\mathbf{y}=\boldsymbol{\varphi}(\mathbf{q})$, where $\mathbf{q}=[q_1, ..., q_n]^T$ is a set of independent joint variables of the mechanism, the velocity transmission takes the form

$$\dot{\mathbf{y}} = \frac{\partial \boldsymbol{\varphi}(\mathbf{q})}{\varphi \mathbf{q}} \dot{\mathbf{q}} = \mathbf{J}_{q}(\mathbf{q}) \dot{\mathbf{q}}$$
(7)

For a given set of joint coordinates \mathbf{q} , the twist $\dot{\mathbf{y}}$ of the end-effector frame (EEF) is a linear combination of the joint rates $\dot{\mathbf{q}}$ with the columns of the Jacobian \mathbf{J}_q acting as coefficient vectors. Assuming ideal transmission behaviour within the transmission element, power is neither generated nor consumed. Thus, virtual work at the input and the output can be set equal, and one obtains

$$\delta \mathbf{q}^{\mathrm{T}} \mathbf{Q} = \delta \mathbf{q}^{\mathrm{T}} \mathbf{Q}^{\mathrm{T}} \tag{8}$$

where the virtual displacements fulfil $\delta \mathbf{q}' = \mathbf{J}_q \delta \mathbf{q}$. This yields $\delta \mathbf{q}^T \mathbf{Q} = \delta \mathbf{q}^T \mathbf{J}_q^T \mathbf{Q}'$ and since this holds for any $\delta \mathbf{q}$, the force transmission takes the form

$$\mathbf{Q} = \mathbf{J}_{a}^{\mathrm{T}} \mathbf{Q}' \,. \tag{9}$$

The relations (7) and (9) for velocity transmission and force transmission is called *kinetostatic duality*. The basic idea of the *kinematical differentials* is to evaluate the velocity transmission function $\dot{\boldsymbol{\phi}}$ for pseudo unit velocities $\hat{\boldsymbol{q}}_i = [0,...,1,...,0]^T$ with zeros everywhere up to the i-th component in order to identify the i-th column of the Jacobian \mathbf{J}_q . Thus, the Jacobian \mathbf{J}_q can be determined with n passes of velocity transmission. This method is called *velocity-based Jacobian algorithm*. By exploiting kinetostatic duality, this algorithm can be analogously applied to force transmission yielding the *force-based Jacobian algorithm*. Here the Jacobian \mathbf{J}_q is computed row-wise by setting unit pseudo forces $\hat{\mathbf{Q}}_i = [0,...,1,...,0]^T$ with zeros everywhere besides the i-th component, performing the force mapping (9), and storing the resulting vector of generalized forces $\hat{\mathbf{Q}}_i$ at the input as the i-th row of \mathbf{J}_q . Thus, the Jacobian can be computed row-wise by six force transmissions for a general manipulator independently of the number of input parameters. This Jacobian evaluation procedure shall be further exploited here.

The complete kinetostatic formalism is implemented in the object-oriented programming library Mobile that uses the C++ programming language (Kecskeméthy, 1994), and a differential geometric interpretation of the kinetostatics has been given in (Kecskeméthy, 1993).

2.3 Modular modeling of parallel kinematic machines

As already mentioned before, the PKM is subdivided into the modules platform, frame, and legs. These components are the foundation of a modular kinetostatic model, which automatically assembles and solves the system of nonlinear constraints. The expressions introduced for the different legs in section 2.1 can be used to calculate the relative kinematics of the different types of PKM. By means of the kinetostatic method, C++ classes for the elementary components of multi-body systems, i.e. prismatic joints, revolute joints, and rigid bodies as well as the constraint solvers for kinematic loops are used for the modeling. These elements defined the required transmission functions for position, velocity, acceleration, and force. Thus, if the machine can be automatically assembled from these classes, one receives a comprehensive tool for the kinematic analysis of parallel robots.

A class called generic machine assembles a kinetostatic model from the components introduced above. Firstly, the legs is attached to the platform and to the frame. For the forward kinematics the legs provide constraints that characterize which motions can be transmitted. The generic machine assembly module collects the constraints from the legs and the generalized coordinates from the platform in order to combine them to a nonlinear system of equations. Then a numerical procedure like a Newton-Raphson-algorithm is applied to solve the forward kinematics. Once the position of the platform is determined one can use local methods from the legs to calculate the complementary variables of the passive joints in each kinematic loop.

The module frame defines the geometry of the base of the PKM by providing the position and orientation of the coordinate frame K_{Ci} , i=1,...,6. These coordinate frames are connected to the world coordinate system by rigid links. On the other hand, the coordinate frames K_{Ci} are the interface for the legs to be attached to it. The module platform firstly defines the endeffectors frame K_{P} by means of the world coordinates $\mathbf{y}=[x,y,z,\psi,\theta,\xi]^{T}$ with respect to the world coordinate frame K_0 , where x,y,z define the Cartesian position of the end-effector and ψ,θ,ξ are a parameterization of the special orthogonal group SO(3), e.g. by use of Bryant angles. The position of the platform pivot points K_{Bi} , i=1,...,6 can be defined with respect to the frame K_{P} giving the geometry of the moveable platform. This presents a kinetostatic transmission function μ (**y**) mapping the world coordinates **y** to the pivot points K_{Bi} . The modules for the legs present the governing properties for the kinematic transmission of the PKM, i.e. by presenting the kinetostatic transmission elements for the joints and the rigid links. The frames K_{Bi} and K_{Ci} act as interfaces to attach the legs to the platform and the base. To solve the forward kinematics each type of leg presents a specific constraint $v_i(K_{\text{Bi}}, K_{\text{Ci}}, q_i)$ which will be used by a central solver for forward kinematics. The constraint v_i for the different types of legs basically implements equations (3) – (6). Finally, the legs implement functions to solve for the angles in the passive joints, i.e. computes the orientation of the universal and spherical joints. This can be done in an explicit way by projection techniques that are well-known from solving four-link bar mechanisms.

For the forward kinematic problem one has to determine the platform world coordinates \mathbf{y} from given generalized coordinates \mathbf{q} . Based on the aforementioned modules the following algorithm can be used for all parallel robots treated in this work:

- 1. The module frame calculates the pivot points K_{Ci}.
- 2. A central constraint solver collects the constraint $v_i(K_{Bi}, K_{Ci}, q_i)$ from each leg module. Furthermore, the constraint solver receives the function $\mu(\mathbf{y})$ from the module platform. The constraint solver uses these equations to set up the nonlinear system $\Gamma(\mathbf{q}, \mathbf{y})=0$.
- 3. The constraint solver calculates the solution \mathbf{y}^* for the system $\Gamma(\mathbf{q}, \mathbf{y})=\mathbf{0}$ with a Newton-Raphson-algorithm.
- 4. The platform updatest the K_{Bi} with $\mu(\mathbf{y}^*)$.
- 5. Each leg determines the dependent angles of the passive joints from the known values of (K_{Bi},K_{Ci},q_i).

Thus, a comprehensive algorithm for forward kinematics for the Stewart-Gough-platform, the Delta-robot, and Linapod like machines is presented. Note, that by using the kinetostatic methods one also receives these relations in terms of velocity, acceleration, and force. The resulting kinetostatic model can be used for a wide range of functions for kinematic analysis e.g. forward kinematics, calculation of the Jacobian matrix, and dexterity indexes, and equations of motion. The discussion of all these algorithms is out of the scope of the paper. In the following section, the determination of a geometrical linearization will be highlighted.

2.4 Linearization and sensitivity analysis

In this study, the function of the forward kinematics of a multi-body system is denoted by φ (**q**,**g**), where **q** are the generalized independent joint coordinates, and **g** collects all geometric parameters of the manipulator. The evaluation of the forward kinematics yields the world coordinates **y** of a particular point of the end-effector of a manipulator together with the orientation of the end-effector, which shall be denoted here as end-effector frame (EEF). For most of the non-serial mechanisms, the function of the forward kinematics φ is not unique, since there may be multiple positions for the EEF that correspond to a given set of generalized joint coordinates **q** due to different assembly modes. Here, it is assumed that it is possible to choose the solution that corresponds to the actual assembly mode, e.g. by

giving appropriate initial conditions. The linearization of the mechanism is formally achieved by taking the derivative with respect to the variables **q** and **g**, respectively, i.e. as

$$\delta \mathbf{y} = \frac{\partial \varphi(\mathbf{q}, \mathbf{g})}{\partial (\mathbf{q}, \mathbf{g})} \delta(\mathbf{q}, \mathbf{g}) = \frac{\partial \varphi(\mathbf{q}, \mathbf{g})}{\partial \mathbf{q}} \delta \mathbf{q} + \frac{\partial \varphi(\mathbf{q}, \mathbf{g})}{\partial \mathbf{g}} \delta \mathbf{g} = \mathbf{J}_{q} \delta \mathbf{q} + \mathbf{J}_{g} \delta \mathbf{g} .$$
(10)

Here, quantities $\delta \mathbf{y}$, $\delta \mathbf{q}$, $\delta \mathbf{g}$ denote infinitesimal variations of the aforementioned coordinates, while J_q denotes the well-known Jacobian matrix that is related to the kinematic dexterity of the manipulator. The matrix J_g is also a Jacobian matrix which characterizes the sensitivity of the position y of the EEF with respect to small changes, e.g. errors, in geometric parameters and which is used for sensitivity analysis.

For serial manipulators with n dof, the function ϕ can be written analytically in terms of the Denavit-Hartenberg-parameters $(\alpha, \theta, d, a)_i$ (Denavit & Hartenberg, 1955) and the vector of the geometric parameters becomes

$$\mathbf{g} = [\alpha_1, \theta_1, d_1, a_1, \dots, \alpha_n, \theta_n, d_n, a_n]^{\mathrm{T}}.$$
(11)

Thus, the Jacobian matrices J_q and J_g can be calculated symbolically for serial manipulators. For nontrivial robots, however, the expressions are usually so extensive that they only can be handled by means of computer algebra.

Complex manipulator systems are characterized by the occurrence of closed kinematic loops. Such mechanisms have more joints than degrees-of-freedom, and the joint coordinates are coupled through closure conditions. This implies that the expressions for ϕ are either complicated, or that $\boldsymbol{\varphi}$ can only be computed point-wise by the iterative solution of an implicit system of nonlinear constraints the latter being the general case which occurs especially for parallel kinematic mechanisms that involve multiple coupled loops. Closed kinematic loops also occur in transmission mechanisms that can be found for instance in hydraulically driven manipulators like excavators or large scale manipulators, since they support the force transmission.

To overcome the lack of an analytical forward kinematic function for complex manipulators, the loop closure conditions f(y,g)=0 can be utilized for sensitivity analysis; by applying implicit differentiation (see e.g. Wittwer et al., 2004), one obtains

$$\frac{\partial \mathbf{f}(\mathbf{y},\mathbf{g})}{\partial \mathbf{y}} \delta \mathbf{y} + \frac{\partial \mathbf{f}(\mathbf{y},\mathbf{g})}{\partial \mathbf{g}} \delta \mathbf{g} = \mathbf{A} \delta \mathbf{y} + \mathbf{B} \delta \mathbf{g} = \mathbf{0} .$$
(12)

where $\mathbf{y} = [\mathbf{x}^T, \boldsymbol{\theta}^T]^T$ are the world coordinates of the end-effector frame, e.g. in form of a position vector \mathbf{x} in \mathbb{R}^3 and the orientation $\boldsymbol{\theta}$ holding e.g. Bryant angles, and \mathbf{g} are the geometric parameters. Then, one immediately obtains for the variation of the EEF world coordinates $\delta y = A^{-1}B\delta g$, where the matrix $A^{-1}B$ maps the errors δg in the components to the displacements δy of the EEF (Wittwer et al., 2004).

There are certain drawbacks to this approach: First, for mechanisms with more than three dof, an analytical form of matrix A-1 can hardly be handled due to the length of the corresponding expressions. Second, if sensitivity analysis is established on the closure condition, one cannot access geometric parameters that are canceled ad-hoc through the formulation of the closure conditions. For example, the normal distance between the joint axes in universal joints is often eliminated because the number of closure constraints can be significantly reduced by assuming it to be exactly zero.

162

2.5 Linearization of manipulator systems



Fig. 5. Velocity and force transmission in a chain of kinetostatic transmission elements

Applying the kinetostatic formalism provides a procedure to calculate position, velocity, acceleration, and force transmission for an arbitrary manipulator. In general, this results in a chain of transmission elements as depicted in Fig. 5, where the individual mapping can also correspond to closed kinematic loops since such loops are also represented by transmission elements. In this figure, a twist $\mathbf{t}_i^T = \begin{bmatrix} \boldsymbol{\omega}_i^T, \mathbf{v}_i^T \end{bmatrix}$ denotes the combination of an angular velocity $\boldsymbol{\omega}_i$ of a frame K_i and its corresponding velocity \mathbf{v}_i of its origin, both decomposed in some frame. A wrench $\mathbf{w}_i^T = \begin{bmatrix} \mathbf{m}_i^T, \mathbf{f}_i^T \end{bmatrix}$ is composed of an applied moment \mathbf{m}_i at the frame K_i and an applied force \mathbf{f}_i at its origin, again decomposed in some frame. Given a certain set of joint coordinates \mathbf{q} , one can introduce a virtual twist displacement $\delta \mathbf{t}_i^T = \begin{bmatrix} \delta \boldsymbol{\varphi}_i^T, \delta \mathbf{r}_i^T \end{bmatrix}$ at the frame K_i , where $\delta \mathbf{r}_i$ is a virtual translational displacement and $\delta \boldsymbol{\varphi}_i$ is an infinitesimal rotational increment in the space of rigid-body rotations, and study the corresponding virtual twist displacement $\delta \mathbf{t}_N$ at the EEF K_N . This linear relation is given by

$$\delta \mathbf{t}_{\mathrm{N}} = (\mathbf{J}_{i+1}\mathbf{J}_{i+2}...\mathbf{J}_{\mathrm{N}})\delta \mathbf{t}_{i} = \mathbf{J}_{i+1}\delta \mathbf{t}_{i} , \qquad (13)$$

where \mathbf{J}_i denotes the Jacobian of the velocity transmission from frame K_{i-1} to the frame K_i . Using kinematical differentials (Sec. 2.2) one calculates the Jacobian $\hat{\mathbf{J}}_{i+1}$, which contains the sensitivity of the frame K_N with respect to displacements in K_i , and then concatenates the matrices $\hat{\mathbf{J}}_i^T$ for the sought matrix $\mathbf{J}_g^T = \begin{bmatrix} \hat{\mathbf{J}}_1^T, \hat{\mathbf{J}}_2^T, ..., \hat{\mathbf{J}}_N^T \end{bmatrix}$. Thus, for a comprehensive linearization, one needs six passes of the velocity transmission function for each $\hat{\mathbf{J}}_i^T$ and hence 6N passes for the whole manipulator. In contrast, one can evaluate the force transmission function, relating the wrench \mathbf{w}_N at the

EEF to the internal wrenches $\mathbf{w}_{i}^{T} = [\mathbf{m}_{i}^{T}, \mathbf{f}_{i}^{T}]$ at the intermediate frames K_{i} , where \mathbf{m}_{i} represents the moment being applied to the frame K_{i} from the base-distal subchain to the base-proximal subchain, and \mathbf{f}_{i} is the corresponding force with respect to the origin. Due to kinetostatic duality, one obtains

$$\mathbf{w}_{i} = \left(\mathbf{J}_{i+1}\mathbf{J}_{i+2}...\mathbf{J}_{N}\right)^{\mathrm{T}}\mathbf{w}_{N} = \hat{\mathbf{J}}_{i+1}^{\mathrm{T}}\mathbf{w}_{N} .$$
(14)

The force transmission presents the major advantage that one can use \mathbf{w}_{i+1} to determine \mathbf{J}_i . Therefore, only 6 passes of the force transmission are needed to calculate the complete Jacobian J_g . This leads to the following simple algorithm to determine the sensitivity Jacobian J_g :

- 1. Solve the forward kinematics for the desired set of joint coordinates \mathbf{q} of the manipulator
- 2. Choose unit forces \mathbf{F}_{x} , \mathbf{F}_{y} , \mathbf{F}_{z} and unit torques \mathbf{M}_{x} , \mathbf{M}_{y} , \mathbf{M}_{z} with $|\mathbf{F}_{x}| = |\mathbf{F}_{y}| = ... = |\mathbf{M}_{z}| = 1$ along the axes x, y, z of the EEF, respectively.
- 3. For each of the unit loads described above, perform the following steps:
 - a) Apply the load to the EEF and compute the internal forces/torques \mathbf{w}_i for each internal frame K_i one is interested in.
 - b) Store the internal forces/torques \mathbf{w}_i in the respective row of the Jacobian \mathbf{J}_g .



Fig. 6. Planar four-bar mechanism a) with nominal geometry b) with virtual error joint for changes δl_2 in length l_2 of the coupler

2.6 Virtual error joints

The relations in the previous section can be intuitively illustrated by *virtual error joints* which were introduced by Woernle (1988) and extended by Pott and Hiller (2004). The basic idea is to insert additional independent joints which allow motion in the direction of the expected errors. Fig. 6a presents a mechanism that involves a kinematic loop. For this planar four-bar mechanism, one might wish to investigate the influence of changes in length of the coupler. One introduces an additional prismatic joint in which the variation of the length of the coupler is embodied (Fig. 6b) and the influence of changes in length of the coupler can be calculated by using the velocity transmission. The algorithm in Sec. 2.5 can also be derived using this model (Pott & Hiller, 2004), where the virtual error joints are used to measure the back-propagated forces.

Based on the linearization algorithm described above, a number of applications can be investigated, as described next.

3. Applications

3.1 Manufacturing error analysis

The Jacobian matrix J_g permits to study the influence of geometric errors on the accuracy of the EEF. These errors may arise from the manufacturing and assembly process of the manipulator. They cannot be avoided, but may be controlled through more precise, but at the same time more expensive manufacturing techniques. Consequently, a comprehensive analysis of the influence of changes in parameters is useful for optimal system design. This analysis is basically done evaluating the Jacobian J_g mapping parameter variations δg to the

displacement $\delta \mathbf{y}$ of the EEF. The magnitude of geometric errors $\delta \mathbf{g}_i$ is normally small compared to the nominal parameter $\delta \mathbf{g}_i$. Therefore, the error estimated by the linear model is very close to the error calculated with the generally nonlinear model (Sec. 4.2).

For different manipulators of the same type, the actual kinematic parameters may vary within the tolerance intervals defined by the manufacturing and assembly process. Here, it is assumed that the actual errors are Gaussian variables with a standard deviation proportional to the tolerance. The changes in parameters are assumed to be small and independent. Thus, the square of the total error is equal to the sum of squares of the single errors obtained by propagation of the manufacturing, clearance, and assembly errors.

For the addition of two Gaussian variables, the standard variation of the sum becomes $\sigma = \sqrt{\sigma_1^2 + \sigma_2^2}$, where σ_1 , σ_2 denote the standard deviations of each summand. In the case of a three-dimensional vector, the total error becomes $\Delta e^2 = e_x^2 + e_y^2 + e_z^2$, where $e_{x,r}e_{y,r}e_z$ are the errors in the three components. For the columns of the Jacobian, one has to mix rotational and translational components. This requires the introduction of metric coefficients that relate rotations to translations and vice versa. Such metric coefficients can be regarded as virtual levers that map rotations at one end to translations at the other and thus generate sensitivities such as "long and slender" (orientations over-emphasized) or "short and thick" (rotations under-emphasized). Assuming that standard deviations σ_i are known for all geometric parameters, the error of the EEF becomes

$$\left|\Delta \mathbf{e}\right| = \sqrt{\sum_{i} \sum_{k} \left(\rho_{k} [\mathbf{J}_{g}]_{ik} \sigma_{i}\right)^{2}} .$$
(15)

where $[\mathbf{J}_g]_{ik}$ denote the entries of the Jacobian and ρ_k are the aforementioned metric coefficients. For the case of an identical standard deviation σ for all components σ_i one obtains

$$\left|\Delta \mathbf{e}\right| = \sigma \sqrt{\sum_{i} \sum_{k} [\mathbf{J}_{g}]_{ik}^{2}} = \sigma \hat{\sigma} .$$
(16)

Here, $\hat{\sigma}$ is referred to as the *overall error amplification index* since it estimates the sensitivity of the whole manipulator at a given configuration with respect to geometric errors. This index is similar to the statistical approach to error analysis of Wittwer et al. (2004).

If a certain accuracy is required for a specific task, the maximal error Δe_{max} of the EEF is known and one can estimate the average standard deviation $\sigma = \Delta e_{max} / \hat{\sigma}$ that is needed for the geometric parameters. This is illustrated in example Sec. 4.2.

3.2 Stiffness analysis

The linearization of a manipulator with respect to its geometric parameters provides a linear mapping between infinitesimal changes in the geometry and infinitesimal variations of the EEF. Assume J_g to be the Jacobian transmitting infinitesimal twists δt_i at each of the frames K_i to infinitesimal twists δt_{EEF} at the end-effector frame K_{EEF} . Moreover, denote by δw_{EEF} a small wrench being applied to the end-effector frame K_{EEF} and by δw_i the corresponding wrenches at the frames K_i ensuring static equilibrium. The Jacobian J_g can be used to set up the stiffness matrix of the mechanism as follows. As pointed out in section 2.2, by equivalence of virtual work it holds

$$\delta \mathbf{w}_{g}^{T} \delta \mathbf{t}_{g} = \delta \mathbf{w}_{EEF}^{T} \delta \mathbf{t}_{EEF} .$$
(17)

where $\delta \mathbf{t}_{g} = [\delta t_{1,1}, \delta t_{1,2}, ..., \delta t_{N,6}]^{T}$ collects all virtual variations of the geometric parameters and $\delta \mathbf{w}_{g} = [\delta \mathbf{w}_{1,1}, \delta \mathbf{w}_{1,2}, ..., \delta \mathbf{w}_{N,6}]^{T}$ are the respective internal forces. Here, each $\delta \mathbf{t}_{i}$ is decomposed in its six elementary components $\delta \mathbf{t}_{i,1}, ..., \delta \mathbf{t}_{i,6}$, where $\delta \mathbf{t}_{i,1}, \delta \mathbf{t}_{i,2}, \delta \mathbf{t}_{i,3}$ are elementary infinitesimal rotations and $\delta \mathbf{t}_{i,4}, \delta \mathbf{t}_{i,5}, \delta \mathbf{t}_{i,6}$ are elementary translations with respect to the coordinate frame axis (Fig. 7). Similarly, the wrench $\delta \mathbf{w}_{i} = [\delta \mathbf{w}_{i,1}, ..., \delta \mathbf{w}_{i,6}]^{T}$ is set up. Assuming that each elementary infinitesimal twist component $\delta \mathbf{t}_{i,j}$ produces a corresponding infinitesimal wrench component $\delta \mathbf{w}_{i,1}$ by means of an associated linear spring with stiffness coefficient $\mathbf{k}_{i,j}$, one obtains

$$\mathbf{K}_{g}^{-1}\delta\mathbf{w}_{g}^{T} = \delta\mathbf{t}_{g} \text{ with } \mathbf{K}_{g}^{-1} = \text{diag}\left\{\frac{1}{k_{1,1}}, \frac{1}{k_{1,2}}, \dots, \frac{1}{k_{N,6}}\right\}.$$

$$(18)$$



Fig. 7. Decomposition of the unit twist δt_i at frame K_i

Note that this assumption is a simplification of the structural properties of a general mechanical component that applies to many technical applications (slender bars, joints, etc.). The generalization to a full generic model is accomplished by a stiffness matrix in which all coefficients may be non-zero and which may be obtained from finite element analysis. This generalization is not further pursued here as is bears no new insight and is not required for the examples treated in this paper. A generalization is conceivable as a later step. On the other hand, it holds

$$\mathbf{K}_{\text{EEF}}^{-1} \delta \mathbf{w}_{\text{EEF}} = \delta \mathbf{t}_{\text{EEF}} \,. \tag{19}$$

where \mathbf{K}_{EEF} is the sought stiffness matrix at the EEF. Substituting Eq. (18) and Eq. (19) into Eq. (17) and using the global force transmission $\delta \mathbf{w}_{g} = \mathbf{J}_{g}^{T} = \delta \mathbf{w}_{\text{EEF}}$ gives

$$\delta \mathbf{w}_{\text{EEF}}^{\mathrm{T}} \mathbf{J}_{g} \mathbf{K}_{g}^{-1} \mathbf{J}_{g}^{\mathrm{T}} \delta \mathbf{w}_{\text{EEF}} = \delta \mathbf{w}_{\text{EEF}}^{\mathrm{T}} \mathbf{K}_{\text{EEF}}^{-1} \delta \mathbf{w}_{\text{EEF}} .$$
(20)

Since this equation holds for any $\delta \mathbf{w}_{\text{EEF}}$, it follows

$$\mathbf{K}_{\text{EEF}}^{-1} = \mathbf{J}_{g} \mathbf{K}_{g}^{-1} \mathbf{J}_{g}^{T} \,. \tag{21}$$

Thus, the Jacobian J_g can be used to transform the stiffness coefficients $k_{i,j}$ of the geometric parameters contained in the stiffness matrix K_g to the global stiffness matrix K_{EEF} .

4. Examples

In this section, the proposed linearization technique is applied to analyze a six-dof parallel kinematic machine where no closed-form solution for the forward kinematics is possible.

4.1 Error analysis for a parallel kinematic manipulator

This example considers the six-dof parallel kinematic machine tool Linapod (Pritschow et al. 2004; Wurst, 1998) installed at the Institute for Control Engineering of Machine Tools and Manufacturing Units at the University Stuttgart (Germany), see Fig. 8. Six rigid links connect the mobile platform to the fixed frame with spherical/universal joints. The pivot points on the frame are actuated by linear drives moving parallel to the z-axis. The nominal position lies in the center of the workspace. Errors in the length of every leg are assumed to be small. Applying the algorithm from section 2.5, the sensitivity matrix J_g for errors in the length of the bar can be established. To this end, the forward kinematic problem is solved to obtain the position \mathbf{r}_i and the direction \mathbf{u}_i of each of the six legs with respect to the EEF (Fig. 3b). The calculation of the internal forces in the bars from force equilibrium conditions is carried out by applying unit forces and torques (Fig. 8) to the EEF resulting in the matrix equation

$$\begin{bmatrix} \mathbf{u}_1 & \dots & \mathbf{u}_6 \\ \underline{\chi}_1 & \dots & \underline{\chi}_6 \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{f}_1 & \dots & \mathbf{f}_6 \end{bmatrix}}_{\mathbf{H}} = \underbrace{\begin{bmatrix} \mathbf{F}_x & \mathbf{F}_y & \mathbf{F}_z & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{M}_x & \mathbf{M}_y & \mathbf{M}_z \end{bmatrix}}_{\mathbf{I}_6}.$$
 (22)

where $\mathbf{f}_i = \mathbf{f}_i \mathbf{u}_i$ are the leg forces, respectively, and $\boldsymbol{\chi}_i = \mathbf{u}_i \times \mathbf{r}_i$. The resulting Jacobian becomes $\mathbf{J}_g^T = \mathbf{A}^{-1}$. With the geometric parameters of Linapod (see Tab.1) the Jacobian \mathbf{J}_g becomes



Fig. 8. Six-dof parallel kinematic machine Linapod with fixed length legs. Unit forces and torques are applied to the platform.

	$\mathbf{J}_{g} = \begin{pmatrix} -0.058 & 0.61 \\ -0.678 & 0.28 \\ -0.154 & -0.13 \\ 0.905 & -2.13 \\ 1.930 & -0.13 \\ -2.230 & -2.23 \end{pmatrix}$	$\begin{array}{rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	567 316 230 410 510 020	16	(23)
leg i		bi	ui	li	$\overline{} q_i$	
1	[0.250, 0.886, 0.0]	[-0.126, 0.180, 0.2]	[0,0,1]	1.25	1.221	
2	[-0.780, -0.421, 0.0]	[-0.093, -0.199, 0.2]	[0,0,1]	1.25	1.221	
3	[0.755, -0.465, 0.0]	[0.219, 0.019, 0.2]	[0,0,1]	1.25	1.221	
4	[-0.250, 0.886, 0.0]	[0.115, 0.164, 0.4]	[0,0,1]	1.70	1.933	
5	[-0.755, -0.465, 0.0]	[-0.199, 0.017, 0.4]	[0,0,1]	1.70	1.933	
6	[0.780, -0.421, 0.0]	[0.085, -0.181, 0.4]	[0,0,1]	1.70	1.933	

Table 1. Geometrical Parameters for the PKM Linapod at its home position.



Fig. 9. Difference between discrete error calculation (exact) and linearization.

Assuming that the length error for all bars is $\Delta \mathbf{e} = \epsilon [1,1,1,1,1,1]^T$ with $\epsilon = 10 \mu m$, the total position error is $|\Delta \mathbf{e}_{\text{EEF}}| = 11.528 \mu m$. This matches the exact solution using the nonlinear forward kinematics up to nine digits. In Fig. 9, the effect of variations of the scaling factor on

the difference $\Delta \mathbf{e}$ between linearized and exact model is illustrated. As it can be verified, the approximation is accurate up to a geometric error of about $\varepsilon = 1$ mm. Still, for $\varepsilon = 10$ mm the relative error is only about 1%, which is still enough for most applications. This shows that the linearization procedure described in this paper is sufficient for most practical applications.

4.2 Accuracy of the Linapod

In this section, the geometric accuracy of the PKM Linapod is analyzed with the force-based method. Assuming errors in every component of the mechanism, the sensitivity matrix J_g contains 126 columns corresponding to the individual geometric parameters. Orientation errors are ignored as these errors are negligible with respect to the translational errors. In Fig. 10 the overall error amplification index according to Eq. (16) is plotted over the workspace. It is recognized from the diagram that the error amplification has its minimum in the center of the workspace, and that the error distribution is roughly circular. It is interesting to observe that changes in the overall error amplification are relatively small from about $\hat{\sigma}$ =4.485 in the center to $\hat{\sigma}$ =5 on the border.



Fig. 10. Overall error-amplification $\hat{\sigma}$ for equally distributed errors in all components. The lines mark the used workspace for Linapod.

Presuming a required accuracy of $\Delta e_{max}=10\mu m$ which is typical for machine tools, this results in an average standard deviation of $\overline{\sigma}=2\mu m$ which is essential to reach the given accuracy. One can conclude that it is not possible to manufacture and assemble the machine

with state-of-the-art techniques and reasonable effort at this tolerance level. Therefore, additional steps like calibration are required to ensure the fulfilment of manufacturing requirements.

4.3 Calculation of the stiffness matrix of the parallel robot Linapod

As shown is Sec. 3.2, the Jacobian J_g can be used for the calculation of the geometric error stiffness matrix. The stiffness coefficients related to elementary geometric variations of a frame are set as $k_1 = 8.8e7$ Nm⁻¹ for the lower and k_u =6.0e7 Nm⁻¹ for the upper leg. Furthermore, elasticity in the linear drives is taken into account with a spring constant k_d =8.13e8 Nm⁻¹. For the calculations, only the translational part of the stiffness matrix is taken into account in order to avoid mixing translational and rotational parts. The resulting stiffness behavior of the Linapod is depicted in Fig. 11 by plotting the minimal eigenvalue of the stiffness matrix over the workspace. As it can be seen, the maximum stiffness property is achieved at the home configuration, with softer values farer away of the home configuration.



Fig. 11. Minimal eigenvalue λ_{min} [10⁷ Nm⁻¹] of the stiffness matrix of the Linapod.

www.intechopen.com

170

Algorithm	All para	ameters	Optimized for Linapod		
	Time (ms)	Relative Time	Time (ms)	Relative Time	
numerical differentiation	163.92	68.87	12.31	5.17	
velocity-based Jacobian	52.13	21.90	4.33	1.82	
force-based Jacobian	2.38	1.00	2.38	1.00	

Table 2. Performance of different algorithms implemented in Mobile on an AMD Athlon 1GHz for the error analysis of all 252 parameters for *Linapod*. Relative times compared to force-based Jacobian.

4.4 Computational considerations

In this section, the computational effort of different algorithms to calculate the sensitivity matrix J_g is compared. The total cost of an algorithm for the error analysis depends on the number of kinematic evaluations, while the administrative overhead e.g. copying and storing the results can be neglected. For the numerical differentiation approach, one needs one evaluation to solve the nominal forward kinematics and one evaluation of the position forward kinematics for each geometric parameter that is considered. The total numerical effort depends on the number of targeted geometric parameters. The velocity-based method (Pott et al., 2007) needs one evaluation of the velocity forward kinematics for each parameter. The force-based approach needs six evaluations of the force transmission. In Tab.2 the computational times of Mobile (Kecskeméthy, 1994) are listed. It can be seen that the numerical differentiation approach needs more time than the velocity-based method, although both need the same number of forward kinematic evaluations. The force-based method needs even less time than the velocity-based method.

5. Conclusions

The contribution describes a general method for kinematic modeling of many wide-spread parallel kinematic machines, i.e. for the Stewart-Gough-platform, the Delta-robot, and Linaglide machines. The kinetostatic method is applied for a comprehensive kinematic analysis of these machines. Based on that model, a general method is proposed to compute the linearization of the transmission behaviour from geometric parameters to the endeffector motion of these machines. By applying the force transmission method, one can perform a linearization with respect to all geometric parameters, for parallel kinematic machines. Especially in cases where no closed-form solution for the forward kinematics is available, the force-based approach provides an efficient procedure for obtaining the linear equations. The method can be directly applied to the presented kinetostatic models of the manipulator and permits also to study parameters that are canceled in the closure conditions. The linear model is used for error analysis and calculation of the stiffness matrix. The algorithm provides a good numerical performance and can be applied to practical examples.

6. Acknowledgment

This work was partly funded by the German Research Foundation (Deutsche Forschungsgemeinschaft) under HI370/19-2 and HI370/19-3 as part of the priority program SPP1099 Parallel Kinematic Machine Tools.

7. References

- Brisan, C.; Franitza, D.; Hiller, M. (2002). Modeling and Analysis of Errors for Parallel Robots, In: *Proceedings of the Kolloquium of SFB 562*, 83-96, Braunschweig, Germany.
- Chase, K. W.; Greenwood, W. H.; Loosli, B. G.; Hauglund, L. F. (1990). Least Cost Tolerance Allocation for Mechanical Assemblies with Automated Process Selection. *Manufactoring Review*, Vol. 3, No. 1, 49-59
- Clavel, R. (1988). Delta, a Fast Robot with Parallel Geometry, In: *18th Int. Symp. on Industrial Robot*, 91-100
- Denavit, J.; Hartenberg, R. (1955). A Kinematic Notation for Lower Pair Mechanisms Based on Matrices. *Mechanisms Based on Matrices*, Vol. 77
- Dietmaier, P. (1998). The Stewart-Gough Platform of General Geometry can have 40 Real Postures, In: *Advances in Robot Kinematics*, 7-16, Kluwer Academic Publishers, Dordrecht
- El-Khasawneh, B. S.; Ferreira, P. M. (1994). Computation of Stiffness and Stiffness bounds for Parallel Link Manipulators. *International Journal of Machine Tools and Manufacture*, Vol. 39, 321-342
- Hiller, M.; Kecskeméthy, A. (1989). Equations of Motion of Complex Multibody Systems Using Kinematical Differentials, *Transactions of the Canadian Society of Mechanical Engineering*, Vol. 13, No. 4, 113-121
- Husty, M. L. (1996). An Algorithm for Solving the Direct Kinematic of Stewart-Gough-type Platforms. *Mechanism and Machine Theory*, Vol. 31, No. 4, 365-380
- Innocenti, C. (1999). A Static-Based Method to Evaluate the Effect of Joint Clearances on the Positioning Errors of Planar Mechanisms, In: *Tenth World Congress on the Theory of Machines and Mechanisms*, Oulu, Finland
- Innocenti, C. (2002). Kinematic Clearance Sensitivity Analysis of Spatial Structures with Revolute Joints. *Transactions of the ASME*, Vol. 124, 52-57
- Jelenkovic, L. & Budin, L. (2002). Error Analysis of a Stewart Platform based Manipulators, In: 6th International Conference on Intelligent Engineering Systems, 83-96, Opatija
- Ji, S.; Li, X.; Ma, Y.; Cai, H. (2000). Optimal Tolerance Allocation Based on Fuzzy Comprehensive Evaluation and Genetic Algorithm. *International Journal of Advanced Manufacturing Technology*, Vol. 16, 461-468
- Kecskeméthy, A. (1993). Objektorientierte Modellierung der Dynamik von Mehrkörpersystemen mit Hilfe von Übertragungselementen, Fortschritt-Berichte VDI, Reihe 20, Nr. 88, VDI Verlag, Düsseldorf
- Kecskeméthy, A. (1994). Mobile User's Guide and Reference Manual, Fachgebiet Mechatronik, University Duisburg-Essen

- Kecskeméthy, A.; Hiller, M. (1994). An Object-Oriented Approach For An Effective Formulation of Multibody Dynamics. *Computer Methods in Applied Mechanics and Engineering*, Vol. 115, 287-314
- Kim, H. S. & Choi Y. J. (2000). The Kinematic Error Bound Analysis of the Stewart Platform. *Journal of Robotic Systems,* Vol. 17, No. 1, 63-73
- Lenord, O.; Fang, S.; Franitza, D.; Hiller, M. (2003). Numerical Linearisation Method to Efficiently Optimize the Oscillation Damping of an Interdisciplinary System Model. *Multibody System Dynamics*, Vol. 10, 201-217
- Parenti-Castelli, V.; Venanzi, S. (2002). A New Deterministic Method for Clearance Influence Analysis in Spatial Mechanisms, In: *Proceedings of ASME International Mechanical Engineering Congress*, New Orleans, Louisiana
- Parenti-Castelli, V.; Venanzi, S. (2005). Clearance Influence Analysis on Mechanisms. *Mechanism and Machine Theory*, Vol. 40, No. 12, 1316-1329
- Pott, A.; Hiller, M. (2004). A Force Based Approach to Error Analysis of Parallel Kinematic Mechanisms, In: *Advances in Robot Kinematics*, 293-302, Kluwer Academic Publishers, Dordrecht
- Pott, A. (2007). Analyse und Synthese von Werkzeugmaschinen mit paralleler Kinematik, Fortschritt-Berichte VDI, Reihe 20, Nr. 409, VDI Verlag, Düsseldorf
- Pott, A.; Kecskeméthy, A.; Hiller, M. (2007). A Simplified Force-Based Method for the Linearization and Sensitivity Analysis of Complex Manipulation Systems. *Mechanism and Machine Theory*, Vol. 42, No. 11, 1445-1461
- Pritschow, G.; Boye, T.; Franitza, T. (2004). Potentials and Limitations of the Linapod's Basic Kinematic Model, *Proceedings of the 4th Chemnitz Parallel Kinematics Seminar*, 331-345, Verlag Wissenschaftliche Scripten, Chemnitz
- Rebeck, E. & Zhang, G. (1999). A method for evaluating the stiffness of a hexapod machine tool support structure. *International Journal of Flexible Automation and Integrated Manufacturing*, Vol. 7, 149-165
- Song, J.; Mou, J.-I.; King, C. (1999). Error Modeling and Compensation for Parallel Kinematic Machines, In: *Parallel Kinematic Maschines*, 172-187, Springer-Verlag, London
- Wittwer, J. W.; Chase, K. W.; Howell, L. L. (2004). The Direct Linearization Method Applied to Position Error in Kinematic Linkages. *Mechanism and Machine Theory*, Vol. 39, No. 7, 681-693
- Woernle, C. (1988). Ein systematisches Verfahren zur Aufstellung der geometrischen Schließbedingungen in kinematischen Schleifen mit Anwendung bei der Rückwärtstransformation für Industrieroboter, Fortschritt-Berichte VDI, Reihe 18, Nr. 18, VDI Verlag, Düsseldorf
- Wurst, K.-H. (1998). Linapod Machine Tools as Parallel Link System in a Modular Design, Proceedings of the 1st European-American Forum on Parallel Kinematic Machines, Milan, Italy

Zhao, J.-W.; Fan, K.-C.; Chang, T.-H.; Li, Z. (2002). Error Analysis of a Serial-Parallel Type Machine Tool. International Journal of Advanced Manufacturing Technology, Vol. 19, 174-179







Parallel Manipulators, towards New Applications Edited by Huapeng Wu

ISBN 978-3-902613-40-0 Hard cover, 506 pages **Publisher** I-Tech Education and Publishing **Published online** 01, April, 2008 **Published in print edition** April, 2008

In recent years, parallel kinematics mechanisms have attracted a lot of attention from the academic and industrial communities due to potential applications not only as robot manipulators but also as machine tools. Generally, the criteria used to compare the performance of traditional serial robots and parallel robots are the workspace, the ratio between the payload and the robot mass, accuracy, and dynamic behaviour. In addition to the reduced coupling effect between joints, parallel robots bring the benefits of much higher payload-robot mass ratios, superior accuracy and greater stiffness; qualities which lead to better dynamic performance. The main drawback with parallel robots is the relatively small workspace. A great deal of research on parallel robots has been carried out worldwide, and a large number of parallel mechanism systems have been built for various applications, such as remote handling, machine tools, medical robots, simulators, micro-robots, and humanoid robots. This book opens a window to exceptional research and development work on parallel mechanisms contributed by authors from around the world. Through this window the reader can get a good view of current parallel robot research and applications.

How to reference

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

Andreas Pott and Manfred Hiller (2008). Kinematic Modeling, Linearization and First-Order Error Analysis, Parallel Manipulators, towards New Applications, Huapeng Wu (Ed.), ISBN: 978-3-902613-40-0, InTech, Available from:

http://www.intechopen.com/books/parallel_manipulators_towards_new_applications/kinematic_modeling__line arization_and_first-order_error_analysis

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 © 2008 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the <u>Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License</u>, 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.



