

Trajectory Control of Serial and Parallel Flexible Manipulators Using Model Inversion

Robert Seifried, Markus Burkhardt, and Alexander Held

Abstract Traditional manipulator designs are based on maximized stiffness to suppress undesired elastic vibrations. This results in high accuracy in end-effector trajectory tracking, while it usually includes a drastic mass increase, a poor weight-to-payload ratio and high energy consumption. In contrast, modern light weight designs result in low energy consumption and allow often high working speeds. However, due to the light weight design the bodies have a significant flexibility which yields undesired vibrations. Therefore, in the control design these flexibilities must be taken into account. In this chapter feedforward control designs based on inverse models are presented and applied to serial and parallel flexible manipulators. Thereby, for a given system output the inverse model provides the control input for exact reproduction of the desired output trajectory and the trajectories of the generalized coordinates.

1 Introduction

In order to achieve in modern machines low energy consumption and allowing high working speeds light weight designs are increasingly often used. However, due to the light weight design the bodies of the manipulators have a significant flexibility which yields undesired vibrations. Therefore, the manipulators must be modeled as flexible multibody system and in the control design these flexibilities must be taken into account. Flexible manipulators are typical examples of underactuated multibody systems, since they generally possess less control inputs than degrees of freedom. In order to obtain a good performance in end-effector trajectory tracking

R. Seifried (✉) · M. Burkhardt · A. Held
Institute of Engineering and Computational Mechanics, University of Stuttgart,
Pfaffenwaldring 9, 70569 Stuttgart, Germany
e-mail: robert.seifried@itm.uni-stuttgart.de

M. Burkhardt
e-mail: markus.burkhardt@itm.uni-stuttgart.de

A. Held
e-mail: alexander.held@itm.uni-stuttgart.de

an accurate and efficient feedforward control is necessary. This is then supplemented by additional feedback control to account for small disturbances and uncertainties. In this chapter the feedforward control design based on inverse models is presented and applied to serial and parallel flexible manipulators. Thereby for a given system output the inverse model provides the control input for exact reproduction of the desired output trajectory. In addition the inverse model provides the trajectories for all generalized coordinates, which can be used in additional feedback control.

In this chapter, firstly, an exact inverse model using concepts from differential geometric control theory [8, 12] is presented and applied to serial and parallel flexible manipulators. The starting point is the explicit symbolic transformation of the equations of motion into the nonlinear input-output normal-form. From this the inverse model is derived, consisting of a chain of differentiators, the driven internal dynamics and an algebraic part. The stability properties of the internal dynamics determine the complexity of the feedforward control design. If the internal dynamics are stable they can be solved by forward time integration. Otherwise, bounded solutions for the internal dynamics must be found by the solution of a two-sided boundary value problem [4, 17]. In order to avoid this, optimization based output relocation is proposed to obtain a system with stable internal dynamics, while keeping the tracking error of the flexible manipulator small.

In addition an alternative approach for feedforward control based on servo-constraints is presented. This yields at first a set of differential-algebraic equations. By using numerical projection into the unconstrained subspace the description of the internal dynamics is obtained, while its differentiation index is reduced. Using the methods available for the first approach, the internal dynamics are then solved in a similar way. This feedforward control concept is applied to a parallel flexible manipulator, where the loop closing constraints and servo-constraints are treated concurrently.

2 Feedforward Control Design by Symbolic Coordinate Transformation

For modeling flexible light weight manipulators the method of flexible multibody systems is often most suitable to represent large nonlinear working motions coupled with elastic vibrations. Since for manipulators the elastic deformations are comparatively small, the floating frame of reference approach can be used [16]. The symbolic multibody system research software Neweul-M² [9] is used to derive the equations of motion in minimal coordinates based on the Newton-Euler equations and D'Alembert's principle. The availability of the symbolic equations of motion is very convenient for nonlinear controller design.

With the vector of generalized coordinates $\mathbf{q} \in \mathbb{R}^f$ the equations of motion in minimal coordinates are obtained as

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{k}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{B}(\mathbf{q})\mathbf{u}. \quad (1)$$

Thereby \mathbf{M} is the mass matrix, \mathbf{k} the vector of generalized Coriolis-, centrifugal- and gyroscopic-forces and \mathbf{g} the vector of applied forces and inner forces due to the body elasticity. The input matrix \mathbf{B} distributes the control inputs $\mathbf{u} \in \mathbb{R}^m$ onto the directions of the f generalized coordinates. The vector of generalized coordinates consists of the rigid coordinates $\mathbf{q}_r \in \mathbb{R}^{f_r}$ representing the rigid body motion and the elastic coordinates $\mathbf{q}_e \in \mathbb{R}^{f_e}$. Then, the equations of motion can be partitioned into

$$\begin{bmatrix} \mathbf{M}_{rr}(\mathbf{q}) & \mathbf{M}_{re}(\mathbf{q}) \\ \mathbf{M}_{re}^T(\mathbf{q}) & \mathbf{M}_{ee}(\mathbf{q}) \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_r \\ \ddot{\mathbf{q}}_e \end{bmatrix} + \begin{bmatrix} \mathbf{k}_r(\mathbf{q}, \dot{\mathbf{q}}) \\ \mathbf{k}_e(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} = \begin{bmatrix} \mathbf{g}_r(\mathbf{q}, \dot{\mathbf{q}}) \\ \mathbf{g}_e(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_r \\ \mathbf{B}_e \end{bmatrix} \mathbf{u}. \quad (2)$$

For serial flexible manipulators the actuation occurs only at the joints of the systems. Then, using a tangent frame for the elastic bodies, the control inputs \mathbf{u} do not directly effect the elastic coordinates and it is $\mathbf{B}_e = \mathbf{0}$. In addition, using relative coordinates the inputs act directly on the rigid coordinates and it is $\mathbf{B}_r = \mathbf{I}$. Since serial manipulators are considered first, this special choice is used in the remainder of this section.

Flexible multibody systems are typical underactuated multibody systems, since they have less control inputs $\mathbf{u} \in \mathbb{R}^m$ than generalized coordinates $\mathbf{q} \in \mathbb{R}^f$ with $m < f$. The control goal is tracking of a system output $\mathbf{y} \in \mathbb{R}^m$, e.g. the end-effector point. In this section an exact inverse model for feedforward control design is derived using concepts from differential geometric control theory [8, 12]. Thereby the starting point is the transformation of the flexible multibody system into the nonlinear input-output normal-form. Then, from this the inverse model is derived.

2.1 Coordinate Transformation into Input-Output Normal-Form

For control design it is often helpful to transform the nonlinear system into the so-called nonlinear input-output normal-form by a diffeomorphic coordinate transformation $\mathbf{z} = \Phi(\mathbf{x})$. Thereby $\mathbf{x} = [\mathbf{q}^T, \dot{\mathbf{q}}^T]^T$ are the original coordinates and \mathbf{z} are the new coordinates of the input-output normal-form, which are derived partially from the system output. The first two derivatives of the system output are

$$\dot{\mathbf{y}} = \mathbf{H}(\mathbf{q})\dot{\mathbf{q}}, \quad \ddot{\mathbf{y}} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}''(\mathbf{q}, \dot{\mathbf{q}}), \quad (3)$$

where \mathbf{H} is the Jacobian matrix of the system output and $\mathbf{h}'' = \dot{\mathbf{H}}\dot{\mathbf{q}}$ is the local acceleration. In (3) the second derivative of the generalized coordinates $\ddot{\mathbf{q}}$ can be replaced by the equations of motion (1), yielding

$$\begin{aligned} \ddot{\mathbf{y}} &= \mathbf{H}\ddot{\mathbf{q}} + \mathbf{h}'' = \mathbf{H}\mathbf{M}^{-1}[\mathbf{g} - \mathbf{k} + \mathbf{B}\mathbf{u}] + \mathbf{h}'' \\ &= \mathbf{H}\mathbf{M}^{-1}\mathbf{B}\mathbf{u} + \mathbf{H}\mathbf{M}^{-1}[\mathbf{g} - \mathbf{k}] + \mathbf{h}''. \end{aligned} \quad (4)$$

If the matrix $\mathbf{H}\mathbf{M}^{-1}\mathbf{B}$ is nonsingular (4) can be solved for the control inputs \mathbf{u} . In this case the matrix $\mathbf{H}\mathbf{M}^{-1}\mathbf{B}$ is called decoupling matrix and the system is said to have vector relative degree $\mathbf{r} = \{r_1, \dots, r_m\} = \{2, \dots, 2\}$. Following [8] the relative degree is defined as the minimal number of derivatives of each system output $h_i(\mathbf{q})$,

$i = 1, \dots, m$ until the inputs \mathbf{u} can be computed. Then, no further derivatives are necessary and the first part of the coordinate transformation is found, which is typical for many flexible manipulators. Then, the nonlinear coordinate transformation is given by

$$\mathbf{z} = \Phi(\mathbf{x}) = \Phi(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \quad \begin{array}{l} z_1 = \mathbf{y} = \mathbf{h}(\mathbf{q}) \in \mathbb{R}^m, \\ \text{with } z_2 = \dot{\mathbf{y}} = \mathbf{H}(\mathbf{q})\dot{\mathbf{q}} \in \mathbb{R}^m, \\ z_3 = \Phi_3(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{2(f-m)}. \end{array} \quad (5)$$

Thereby the coordinates z_3 are determined such that (5) forms at least a local diffeomorphic coordinate transformation, which requires that the Jacobian matrix $\mathbf{J} = \partial\Phi(\mathbf{x})/\partial\mathbf{x}$ is nonsingular.

Applying the coordinate transformation (5) to the equations of motion (1) yields the nonlinear input-output normal-form. However, the complete symbolic transformation is often quite difficult. In the following it is shown, that the input-output normal-form can be established efficiently using a linearly combined system output and the partitioned equations of motion (2). For flexible manipulators a linearly combined output $\mathbf{y} = \mathbf{q}_r + \mathbf{\Gamma}\mathbf{q}_e$ of rigid coordinates \mathbf{q}_r and elastic coordinates \mathbf{q}_e is often a suitable choice [11, 15]. With such a system output the end-effector position of serial flexible manipulators may be approximated such that $\mathbf{r}^{ef}(\mathbf{q}_r, \mathbf{q}_e) \approx \mathbf{r}(\mathbf{y})$. The determination of the weighting matrix $\mathbf{\Gamma}$ is discussed in Sect. 3. For the special case of $\mathbf{\Gamma} = \mathbf{0}$ the output reduces to $\mathbf{y} = \mathbf{q}_r$, which is the so-called collocated output. Thus, for flexible manipulators with linearly combined output a suitable coordinate transformation is given by

$$\mathbf{z}_1 = \mathbf{y} = \mathbf{q}_r + \mathbf{\Gamma}\mathbf{q}_e, \quad \mathbf{z}_2 = \dot{\mathbf{y}} = \dot{\mathbf{q}}_r + \mathbf{\Gamma}\dot{\mathbf{q}}_e, \quad \mathbf{z}_3 = [\mathbf{q}_e^T, \dot{\mathbf{q}}_e^T]^T. \quad (6)$$

In order to derive the input-output normal-form the rigid coordinates \mathbf{q}_r are expressed in terms of the output \mathbf{y} and the elastic coordinates \mathbf{q}_e . Then, after symbolic manipulations the nonlinear input-output normal-form is obtained

$$\tilde{\mathbf{M}}\ddot{\mathbf{y}} = \tilde{\mathbf{g}} - \tilde{\mathbf{k}} + \mathbf{u} \quad (7a)$$

$$(\mathbf{M}_{ee} - \mathbf{M}_{re}^T\mathbf{\Gamma})\ddot{\mathbf{q}}_e = \mathbf{g}_e - \mathbf{k}_e - \mathbf{M}_{re}^T\tilde{\mathbf{M}}^{-1}(\tilde{\mathbf{g}} - \tilde{\mathbf{k}} + \mathbf{u}), \quad (7b)$$

with the terms summarized according to the convention $\tilde{\mathbf{M}} = \mathbf{M}_{rr} - (\mathbf{M}_{re} - \mathbf{M}_{rr}\mathbf{\Gamma})(\mathbf{M}_{ee} - \mathbf{M}_{re}^T\mathbf{\Gamma})^{-1}\mathbf{M}_{re}^T$ and $\tilde{\mathbf{g}} = \mathbf{g}_r - (\mathbf{M}_{re} - \mathbf{M}_{rr}\mathbf{\Gamma})(\mathbf{M}_{ee} - \mathbf{M}_{re}^T\mathbf{\Gamma})^{-1}\mathbf{g}_e$ and $\tilde{\mathbf{k}} = \mathbf{k}_r - (\mathbf{M}_{re} - \mathbf{M}_{rr}\mathbf{\Gamma})(\mathbf{M}_{ee} - \mathbf{M}_{re}^T\mathbf{\Gamma})^{-1}\mathbf{k}_e$. For details on the performed symbolic computations see e.g. [15]. Equation (7a) has dimension m and describes the relationship between the inputs \mathbf{u} and outputs \mathbf{y} . Equation (7b) has in this case dimension $f - m$ and is called internal dynamics. Its behavior is crucial for control design and has to be analyzed carefully. For this task the concept of zero-dynamics is often very useful. The zero-dynamics of a nonlinear system are the internal dynamics of the system under the constraint that the output is kept exactly constant, e.g. $\mathbf{y} = \mathbf{0}, \forall t$. A nonlinear system is called asymptotically minimum phase if the equilibrium point of the zero-dynamics is asymptotically stable. Otherwise the system is called non-minimum phase [8, 12].

2.2 Inverse Model

The inverse model follows directly from the input-output normal-form (7a)–(7b). The inverse model provides the inputs \mathbf{u}_d which are required for exact reproduction of a desired system output trajectory $\mathbf{y} = \mathbf{y}_d$. In addition the corresponding trajectories of all generalized coordinates \mathbf{q}_d are obtained. The inputs \mathbf{u}_d follow from Eq. (7a) as

$$\mathbf{u}_d = \tilde{\mathbf{M}}(\mathbf{y}_d, \mathbf{q}_e) \ddot{\mathbf{y}}_d - \tilde{\mathbf{g}}(\mathbf{y}_d, \mathbf{q}_e, \dot{\mathbf{y}}_d, \dot{\mathbf{q}}_e) + \tilde{\mathbf{k}}(\mathbf{y}_d, \mathbf{q}_e, \dot{\mathbf{y}}_d, \dot{\mathbf{q}}_e). \quad (8)$$

The computation of the inputs \mathbf{u}_d depends on the desired outputs $\mathbf{y}_d, \dot{\mathbf{y}}_d$ and the elastic states $\mathbf{q}_e, \dot{\mathbf{q}}_e$. The latter are the solution of the internal dynamics given by Eq. (7b) which is driven by $\mathbf{y}_d, \dot{\mathbf{y}}_d$ and \mathbf{u}_d . Replacing \mathbf{u}_d in the internal dynamics (7b) by Eq. (8) yields for the values of the elastic states $\mathbf{q}_e, \dot{\mathbf{q}}_e$ the differential equation

$$\begin{aligned} & [\mathbf{M}_{ee}(\mathbf{y}_d, \mathbf{q}_e) - \mathbf{M}_{re}^T(\mathbf{y}_d, \mathbf{q}_e) \mathbf{\Gamma}] \ddot{\mathbf{q}}_e \\ & = \mathbf{g}_e(\mathbf{y}_d, \mathbf{q}_e, \dot{\mathbf{y}}_d, \dot{\mathbf{q}}_e) - \mathbf{k}_e(\mathbf{y}_d, \mathbf{q}_e, \dot{\mathbf{y}}_d, \dot{\mathbf{q}}_e) - \mathbf{M}_{re}^T(\mathbf{y}_d, \mathbf{q}_e) \ddot{\mathbf{y}}_d. \end{aligned} \quad (9)$$

Several methods for model inversion exist which differ in the solution of the internal dynamics (9):

Classical Inversion In classical inversion [6] the $\mathbf{q}_e, \dot{\mathbf{q}}_e$ variables can be found through forward integration of the internal dynamics (9) from the starting time point t_0 to the final time point t_f , using the initial values $\mathbf{q}_e(t_0) = \mathbf{q}_{e0}, \dot{\mathbf{q}}_e(t_0) = \dot{\mathbf{q}}_{e0}$. However, depending on the stability of the internal dynamics forward integration might yield unbounded $\mathbf{q}_e, \dot{\mathbf{q}}_e$ values and thus unbounded inputs \mathbf{u}_d , which cannot be used as feedforward control. Therefore, this approach can only be used for feedforward control design if the internal dynamics (9) remain bounded, which implies that the system is minimum phase. An example is a flexible manipulator with collocated output $\mathbf{y} = \mathbf{q}_r$, i.e. $\mathbf{\Gamma} = \mathbf{0}$.

Stable Inversion Flexible manipulators with the end-effector point as system output turn out to be often non-minimum phase and classical inversion cannot be used. However, using stable inversion [4] the inversion problem can be solved, such that the trajectories $\mathbf{q}_e, \dot{\mathbf{q}}_e$ of the internal dynamics (9) and the control inputs \mathbf{u}_d remain bounded. However, the solution might be non-causal. The solution of the stable inversion is formulated as a two-sided boundary value problem, where the boundary conditions are described by the unstable and stable eigenspaces $\mathbf{E}_0^u, \mathbf{E}_f^s$ at the corresponding equilibrium points of the internal dynamics. These are local approximations of the unstable manifold \mathbf{W}_0^u and stable manifold \mathbf{W}_f^s at the starting and ending equilibrium point, respectively, see [12]. This yields for the internal dynamics bounded trajectories $\mathbf{q}_e, \dot{\mathbf{q}}_e$ which start at time t_0 on the unstable manifold \mathbf{W}_0^u and reach the stable manifold \mathbf{W}_f^s at time t_f . Thus the initial conditions $\mathbf{q}_{u0}, \dot{\mathbf{q}}_{u0}$ at time t_0 cannot exactly be pre-designated. A pre-actuation phase $[t_{pr}, t_0]$ is

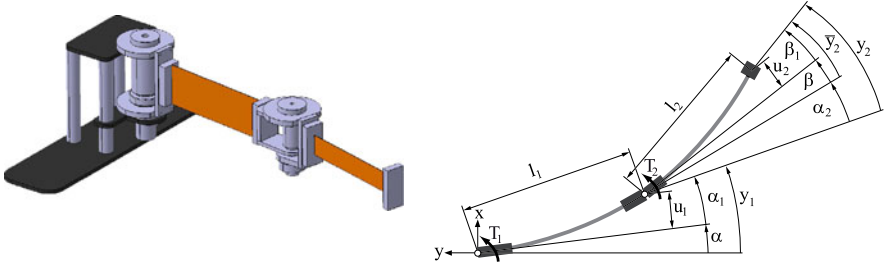


Fig. 1 Serial flexible manipulator

necessary which drives the system along the unstable manifold to a particular initial condition $\mathbf{q}_e(t_0)$, $\dot{\mathbf{q}}_e(t_0)$, while maintaining the constant output $\mathbf{y}_d = \mathbf{y}_d(t_0)$. Also a post-actuation phase $[t_f, t_{po}]$ is necessary to drive the internal dynamics along the stable manifold close to its resting position. The two-sided boundary value problem has to be solved numerically. This can be done by a finite difference method [17], e.g. using the Matlab solver *bvp5c*.

3 Determination of the Linearly Combined Output

The presented derivation of the inverse model depends on a linearly combined system output. In this section the derivation of the linearly combined output is demonstrated exemplarily for a serial flexible manipulator which moves in the horizontal plane. The manipulator consists of two elastic arms with rigid elements attached at their ends for mounting of the motors and end-effector mass, see Fig. 1. The total length of the first arm is denoted as l_1 and the rigid end parts have length l_{r11} and l_{r12} . The second arm has length l_2 and the rigid end parts have length l_{r21} and l_{r22} . The two motors produce the applied torques T_1 and T_2 which act on the joint angles $\mathbf{q}_r = [\alpha, \beta]^T$. The transverse elastic deformations of the two arms are described by the elastic coordinates $\mathbf{q}_e^1 \in \mathbb{R}^r$ and $\mathbf{q}_e^2 \in \mathbb{R}^s$. In the following the matrix $\mathbf{\Gamma}$ is derived from geometrical considerations and by optimization such that the outputs $\mathbf{y} = [y_1, y_2]^T$ yield a very good approximation of the end-effector position,

$$\mathbf{r}^{ef}(\mathbf{q}_r, \mathbf{q}_e) \approx \bar{\mathbf{r}}^{ef}(\mathbf{y}) = \begin{bmatrix} l_1 \sin(y_1) + l_2 \sin(y_1 + y_2) \\ -l_1 \cos(y_1) - l_2 \cos(y_1 + y_2) \end{bmatrix}. \quad (10)$$

Here y_1 and y_2 can be viewed as auxiliary angles, see Fig. 1. With this approximation the desired trajectories for the system output \mathbf{y}_d can be computed by rigid body inverse kinematics from the desired trajectory \mathbf{r}_d^{ef} of the end-effector point.

3.1 Choice of System Output Using Geometrical Considerations

Due to the body elasticity the tips of the arms are subjected to the displacements u_1 and u_2 . These are perpendicular to the undeformed axes of the arm. Restricting to small elastic rotations, these elastic deformations are given by

$$\begin{aligned} u_1 &= \sum_{i=1}^r \Phi_i^1 q_{ei}^1 + l_{r12} \sum_{i=1}^r \Psi_i^1 q_{ei}^1, \\ u_2 &= \sum_{i=1}^s \Phi_i^2 q_{ei}^2 + l_{r21} \sum_{i=1}^s \Psi_i^2 q_{ei}^2. \end{aligned} \quad (11)$$

The i th elastic coordinate of the first arm is denoted by q_{ei}^1 and of the second arm by q_{ei}^2 . The value of the i th displacement shape function at the end of the elastic parts of the first and second arm are denoted by Φ_i^1 and Φ_i^2 , respectively. The values of the i th shape functions for the elastic rotation evaluated at the end of the elastic parts of arm one and two are denoted by Ψ_i^1 and Ψ_i^2 , respectively. Thus, the first terms of Eq. (11) represent the transverse elastic deformation of the elastic parts of both bodies. Due to the elastic rotation of these elastic parts, the rigid end parts of the arms undergo an rotation with respect to their reference frame given by

$$\bar{\alpha}_2 = \sum_{i=1}^r \Psi_i^1 q_{ei}^1 \quad \text{and} \quad \bar{\beta}_2 = \sum_{i=1}^s \Psi_i^2 q_{ei}^2. \quad (12)$$

The influence of $\bar{\alpha}_2$ and $\bar{\beta}_2$ is represented by the second terms of Eq. (11). These deformations u_1, u_2 result in the approximate deformation angles α_1, β_1 , see Fig. 1. For small displacements these two deformation angles α_1, β_1 can be determined and expressed as linear combinations of the elastic coordinates as

$$\alpha_1 \approx \frac{u_1}{l_1} = \sum_{i=1}^r \frac{\Phi_i^1 + l_{r12} \Psi_i^1}{l_1} q_{ei}^1 = \sum_{i=1}^r \Gamma_{1i} q_{ei}^1 \quad (13)$$

and

$$\beta_1 \approx \frac{u_2}{l_2} = \sum_{i=1}^s \frac{\Phi_i^2 + l_{r22} \Psi_i^2}{l_2} q_{ei}^2 = \sum_{i=1}^s \Gamma_{2(r+i)} q_{ei}^2. \quad (14)$$

From these two equations the linearly combined system outputs $y_1 = \alpha + \alpha_1$ and $\bar{y}_2 = \beta + \beta_1$ can be determined. For flexible manipulators without any rigid parts similar outputs are used for trajectory tracking by [11] and [18]. However, when using this output to approximate the end-effector point (10) of a multi-link flexible manipulator, the elastic rotation of the coordinate system attached to tip of arm 1 is neglected. Since in this coordinate system the motor angle β is described, the output \bar{y}_2 is not suitable for end-effector tracking, see [15]. Thus the system output y_2 has to be corrected by the additional angle α_2 given by

$$\alpha_2 = \bar{\alpha}_2 - \alpha_1 = \sum_{i=1}^r (\Psi_i^1 - \Gamma_{1i}) q_{ei}^1 = \sum_{i=1}^r \Gamma_{2i} q_{ei}^1. \quad (15)$$

From this follows the system output $y_2 = \bar{y}_2 + \alpha_2 = \beta + \beta_1 + \alpha_2$, which contains contributions of the elastic deformation of arm 1 and arm 2. The linearly combined system output which is suitable to approximate the end-effector point by Eq. (10) is then given by

$$y = q_r + \Gamma q_e = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} + \begin{bmatrix} \Gamma_{11} & \dots & \Gamma_{1r} & 0 & \dots & 0 \\ \Gamma_{21} & \dots & \Gamma_{2r} & \Gamma_{2(r+1)} & \dots & \Gamma_{2(r+s)} \end{bmatrix} \begin{bmatrix} q_{e1}^1 \\ \vdots \\ q_{er}^1 \\ q_{e1}^2 \\ \vdots \\ q_{es}^2 \end{bmatrix}. \quad (16)$$

3.2 Design of System Output by Optimization

Output relocation is a method where a different system output \hat{y} is chosen in order to achieve minimum phase property. In [11] output relocation for a flexible two arm manipulator is investigated. Thereby a linearly combined system output with block diagonal matrix Γ is used. It is shown that for the two outputs the entries of Γ can be scaled with a value between 0 and 1 to obtain minimum phase property. However, the influence of the elastic rotation (15) of the first body on the second system output is neglected and might result in large end-effector errors, see [15].

Therefore, an optimization based design procedure for a new system output \hat{y} is proposed here to obtain a minimum phase design of underactuated multibody systems and also a very good approximation of the end-effector point r^{ef} . Thereby the new output \hat{y} is an artificial output, which does not represent a specific material point of the multibody system. For this task the linearly combined output is very convenient, since it provides an easy way for the parametrization of the design variables,

$$\hat{y} = q_r + \Gamma(p)q_e. \quad (17)$$

Here the design variables p are just the entries of the weighting matrix Γ . Then, it follows with $y_d = \mathbf{0}$, $\forall t$ from (9) that the zero dynamics, which depend only on the elastic coordinates q_e , \dot{q}_e and the design variables p , are given by

$$[M_{ee}(q_e) - M_{re}^T(q_r)\Gamma(p)]\ddot{q}_r = g_r(q_r, \dot{q}_e) - k_e(q_e, \dot{q}_e). \quad (18)$$

The linearization of the zero dynamics yield the system matrix $A(p)$ which also depends on the design variables. The design goal is to achieve a stable zero dynamics by changing the system output, whereby the new system output should still yield a good approximation of the end-effector position r^{ef} . Therefore, a two-step computation of the optimization criterion $f(p)$ is proposed, which should be minimized in the course of the optimization. The two steps of the optimization criterion computation are:

Step 1 Firstly, local asymptotic stability is checked. Thus, all eigenvalues of the linearized zero dynamics must be in the left half-plane,

$$\operatorname{Re}[\lambda(\mathbf{A}(\mathbf{p}))] < 0. \quad (19)$$

If at least one eigenvalue has a non-negative real part, a large default value for the optimization criterion $f(\mathbf{p})$ is returned. Otherwise, the linearized analysis shows asymptotic stability of the zero dynamics and it is proceeded with step 2.

Step 2 If all eigenvalues of the zero dynamics are in the left half-plane, the actual optimization criterion $f(\mathbf{p})$ will be calculated. Therefore a feedforward control for a test trajectory \mathbf{y}_d is computed. From this inverse model the end-effector trajectory $\mathbf{r}^{ef}(\mathbf{q}_r, \mathbf{q}_e) \in \mathbb{R}^m$ and the effective deviation $e(t)$ from the desired trajectory can be determined as

$$e(t) = \sqrt{\sum_{i=1}^m e_i^2(t)} \quad \text{with } \mathbf{e}(t) = \mathbf{r}_d^{ef}(t) - \mathbf{r}^{ef}(\mathbf{q}_r, \mathbf{q}_e). \quad (20)$$

Then, the optimization criterion is chosen as the maximal effective end-effector deviation

$$f(\mathbf{p}) = \max_t e(t). \quad (21)$$

With the solution for the internal dynamics for the desired trajectory \mathbf{y}_d the boundedness of the internal dynamics is verified. If unbounded states for this design occur the time integration fails and also a large default value is returned for the optimization criterion $f(\mathbf{p})$.

The optimization criterion is discontinuous due to the distinction between stable and unstable designs. Therefore, gradient based methods cannot be used and the stochastic particle swarm optimization algorithm is applied, see [13] for details on the used algorithm. The presented optimization based approach for designing a suitable system output yields a good tracking performance for a given desired output trajectory. If the optimized output $\mathbf{\Gamma}(\mathbf{p}^*)$ also yields a good end-effector approximation for a different trajectory has to be checked for each particular case.

4 Application to a Serial Manipulator

The feedforward control design is first demonstrated for a planar serial flexible manipulator as shown in Fig. 1. The first arm has length $l_1 = 351$ mm and consists of a first rigid part of length $l_{r11} = 63.5$ mm, an elastic part of length $l_{e1} = 209$ mm and a second rigid part of length $l_{r12} = 78.5$ mm. The elastic part has thickness 1.27 mm and height 76.2 mm. The second arm has length $l_2 = 287.5$ mm and consists of a first rigid part of length $l_{r21} = 62.5$ mm, an elastic part of length $l_e = 210$ mm and a rigid end-effector mass of length $l_{r22} = 15$ mm. The elastic part of the second arm

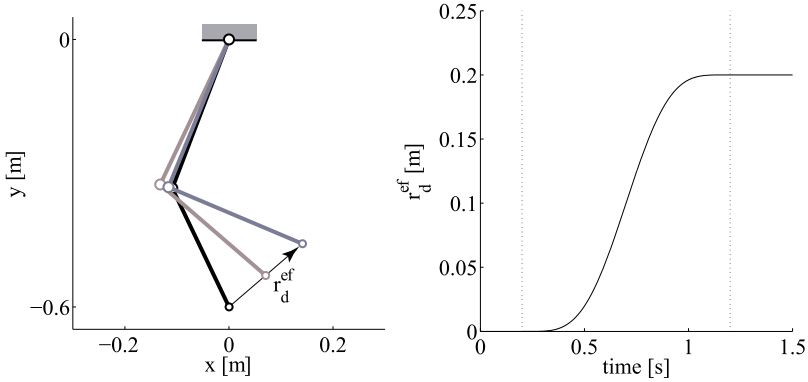


Fig. 2 Desired trajectory of the manipulator

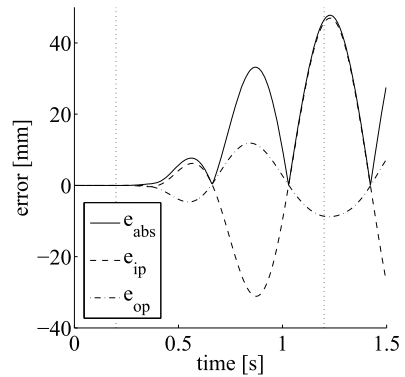
has thickness 0.9 mm and height 38.1 mm. The rigid parts of the arms represent the joints, motor mounting and end-effector mass. All parts of the manipulator are made out of steel. For the description of the elastic deformation of the arms the first two bending eigenmodes are used as shape functions for each arm, i.e. $\mathbf{q}_e \in \mathbb{R}^4$. A tangent floating frame of reference is used. The end-effector of the manipulator should follow a straight test trajectory, whose path is shown in Fig. 2. Along the path, the trajectory is described by a polynomial of ninth order, such that the velocities and accelerations are zero at both the start and the end of the trajectory. The path length is 0.2 m and the time for following the trajectory is 1 s. The trajectory starts from rest at time 0.2 s and ends in rest at time 1.2 s.

In the following simulation results for this flexible manipulator are presented, see also [15] for more details. Thereby the computed feedforward control is tested by simulation, whereby it is combined with PID-feedback control for the joint trajectories \mathbf{q}_j . However, since the inverse model is exact, the PID-control has in these tests only to compensate numerical errors. In reality there are larger uncertainties and disturbances which the controller has to account for. The presented results should demonstrate the capacities of the feedforward control, and the maximal achievable accuracy in the ideal case. In the following the trajectory error in path direction e_{ip} , orthogonal to the path e_{op} and the absolute error $e_{abs} = \sqrt{e_{ip}^2 + e_{op}^2}$ are presented.

Firstly, a feedforward control based on a rigid system output is used, i.e. the elasticities are neglected yielding $\mathbf{\Gamma} = \mathbf{0}$. This is the so-called collocated output which yields a minimum phase system and classical inversion can be used. From the tracking error for this strategy, which is presented in Fig. 3, the strong influence of body flexibility is seen, yielding an unacceptable behavior and errors of several centimeters.

The inversion based feedforward control with linearly combined output is considered next. Using the elastic data of the manipulator and Eqs. (10)–(16) the matrix $\mathbf{\Gamma}$ is given by

Fig. 3 Error of the end-effector trajectory using exact inversion with collocated output



$$\mathbf{\Gamma} = \begin{bmatrix} 25.779 & 47.552 & 0 & 0 \\ 13.500 & 88.897 & 32.316 & 39.470 \end{bmatrix}. \quad (22)$$

An analysis of the zero-dynamics of the flexible manipulator with this output shows, that the system is non-minimum phase. Thus, stable inversion is necessary and the Matlab boundary-value solver *bvp5c* is used. However, it turns out, that for the $\mathbf{\Gamma}$ values given by (22) no solution can be computed numerically using the *bvp5p*. But by slight variations of $\Gamma_{21} = 13.375$ and $\Gamma_{22} = 92.5$, a numerical solution of the boundary value problem is found. In Fig. 4 the end-effector trajectory error is shown for this feedforward control. A very high accuracy is achieved. The trajectory errors are in the magnitude of less than 0.1 mm. After reaching the final position at time 1.2 s only minor deviations of the end-effector point remain.

The optimization based system output design is applied next. For the optimization based design the non-zero entries of the weighting matrix $\mathbf{\Gamma}$ are used as design variables. Based on the values for $\mathbf{\Gamma}$ derived from geometrical considerations (22), the bounds for the design variables are defined. These are set such that a variation of the design variables of $\pm 20\%$ around the geometrical case (22) is allowed. For the optimization 100 particles are used and yields the values

$$\mathbf{\Gamma} = \begin{bmatrix} 25.025 & 45.909 & 0 & 0 \\ 15.292 & 90.345 & 29.296 & 32.297 \end{bmatrix}. \quad (23)$$

The new system output yields a minimum phase system and feedforward control design by classical inversion can be applied. The simulation results are presented in Fig. 4 and show that with this approach high accuracy for the end-effector trajectory can be obtained. The maximal trajectory error is about 0.28 mm. Compared to the previous results the achievable accuracy for this minimum phase system is slightly worse than the output using geometrical considerations yielding a non-minimum phase system, see Fig. 4, but much better than using the collocated output, see Fig. 3. Comparing to results for this manipulator presented in [15], the optimization based output design yields also better results than the so-called quasi-static deformation compensation, which is an alternative approach for feedforward control design of flexible manipulators.

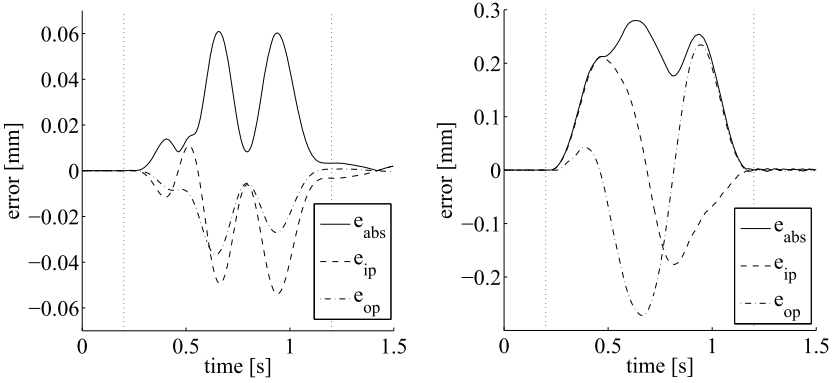


Fig. 4 Error of the end-effector trajectory using stable inversion with Γ from geometrical considerations (*left*) and optimization (*right*)

5 Extension to Parallel Manipulators

For multibody systems with kinematic loops and f_c degrees of freedom the description of the kinematics using a minimal set of generalized coordinates is in general not directly possible. Therefore, the kinematic loop is cut at a suitable joint, yielding a multibody system in tree structure with $f > f_c$ degrees of freedom. The equations of motion of the obtained open loop system are derived in analogy to systems with tree structure according to Eq. (1). By introducing n_c implicit algebraic loop closing constraint equations

$$\mathbf{c}(\mathbf{q}, t) = \mathbf{0}, \quad (24)$$

the equations of motion of the closed loop system can be formulated as a set of differential-algebraic equations (DAE)

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{k}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{B}(\mathbf{q})\mathbf{u} + \mathbf{C}^T(\mathbf{q})\boldsymbol{\lambda} \quad (25a)$$

$$\mathbf{c}(\mathbf{q}, t) = \mathbf{0} \quad (25b)$$

where the matrix \mathbf{C} is the Jacobian matrix of the constraint equations \mathbf{c} with respect to the generalized coordinates $\mathbf{q} \in \mathbb{R}^f$. The vector $\boldsymbol{\lambda} \in \mathbb{R}^{n_c}$ represents the generalized reaction forces in the cut joints. In order to derive the input-output normal-form of the system (25a)–(25b) according to Sect. 2, the set of differential-algebraic equations has to be transformed into a set of $f_c = f - n_c$ ordinary differential equations.

There are several ways of rephrasing differential-algebraic equations as purely differential equations. One way is partitioning the generalized coordinates $\mathbf{q} \in \mathbb{R}^f$ into a set of independent coordinates $\mathbf{q}_i \in \mathbb{R}^{f_c}$ and dependent coordinates $\mathbf{q}_d \in \mathbb{R}^{n_c}$ according to

$$\mathbf{q}^T = [\mathbf{q}_i^T \quad \mathbf{q}_d^T] \quad (26)$$

and expressing the system dynamics in these independent coordinates, see e.g. [10]. The first step is to differentiate the constraint equations \mathbf{c} twice with respect to the time t leading to the constraint equations on acceleration level

$$\ddot{\mathbf{c}}(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}, t) = \mathbf{C}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{c}''(\dot{\mathbf{q}}, \mathbf{q}, t) = \mathbf{0}, \quad (27)$$

with the vector \mathbf{c}'' representing the local accelerations due to the constraints. For linear independent constraint equations and a reasonable choice of dependent coordinates, the Jacobian matrix \mathbf{C} can be split up in an independent and a dependent part leading to

$$\ddot{\mathbf{c}}(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}, t) = \mathbf{C}_d(\mathbf{q})\ddot{\mathbf{q}}_d + \mathbf{C}_i(\mathbf{q})\ddot{\mathbf{q}}_i + \mathbf{c}''(\dot{\mathbf{q}}, \mathbf{q}, t) = \mathbf{0}, \quad (28)$$

in which $\mathbf{C}_d \in \mathbb{R}^{n_c \times n_c}$ must be a regular matrix. Solving Eq. (28) for the dependent accelerations $\ddot{\mathbf{q}}_d$ and dropping the dependencies for better readability, the dependent accelerations $\ddot{\mathbf{q}}_d$ can be expressed as

$$\ddot{\mathbf{q}}_d = -\mathbf{C}_d^{-1}(\mathbf{C}_i\ddot{\mathbf{q}}_i + \mathbf{c}''). \quad (29)$$

Based on this relationship, the generalized accelerations can be written as

$$\ddot{\mathbf{q}} = \begin{bmatrix} \mathbf{I} \\ -\mathbf{C}_d^{-1}\mathbf{C}_i \end{bmatrix} \ddot{\mathbf{q}}_i + \begin{bmatrix} \mathbf{0} \\ -\mathbf{C}_d^{-1}\mathbf{c}'' \end{bmatrix} = \mathbf{J}_c\ddot{\mathbf{q}}_i + \mathbf{b}''. \quad (30)$$

By inserting Eq. (30) into Eq. (25a) and by left-side multiplication with the transposed Jacobian matrix \mathbf{J}_c the equations of motion in minimal coordinates of the multibody body system with kinematic loop are obtained,

$$\mathbf{J}_c^T \mathbf{M}(\mathbf{q})(\mathbf{J}_c\ddot{\mathbf{q}}_i + \mathbf{b}'') + \mathbf{J}_c^T \mathbf{k}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{J}_c^T \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{J}_c^T \mathbf{B}(\mathbf{q})\mathbf{u} + \mathbf{J}_c^T \mathbf{C}^T(\mathbf{q})\boldsymbol{\lambda}. \quad (31)$$

Since the relation

$$\mathbf{J}_c^T \mathbf{C}^T = [\mathbf{I} \quad -\mathbf{C}_i^T \mathbf{C}_d^{-T}] \begin{bmatrix} \mathbf{C}_i^T \\ \mathbf{C}_d^T \end{bmatrix} = \mathbf{C}_i^T - \mathbf{C}_i^T \mathbf{C}_d^{-T} \mathbf{C}_d^T = \mathbf{0} \quad (32)$$

holds, the Lagrange multipliers $\boldsymbol{\lambda}$ vanish in Eq. (31) and the equations of motion can be displayed as

$$\overline{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}}_i + \overline{\mathbf{k}}(\mathbf{q}, \dot{\mathbf{q}}) = \overline{\mathbf{g}}(\mathbf{q}, \dot{\mathbf{q}}) + \overline{\mathbf{B}}(\mathbf{q})\mathbf{u}. \quad (33)$$

These equations of motion in minimal form allow the partitioning of the independent accelerations $\ddot{\mathbf{q}}_i$ into the rigid accelerations $\ddot{\mathbf{q}}_r$ and the elastic accelerations $\ddot{\mathbf{q}}_e$ according to Eq. (2).

Based on this representation the equations of motion (33) can be transformed into the input-output normal-form, which permits the analysis of the internal dynamics and the application of the exact model inversion procedures discussed in the previous sections. This approach is applied to the flexible parallel manipulator shown in Fig. 5. Using for the coordinate partitioning the independent coordinates $\mathbf{q}_i = [\mathbf{q}_r, \mathbf{q}_e]$ with $\mathbf{q}_r = [s_1, \alpha]$, again a linearly combined system output can be used to approximate the end-effector position

$$\mathbf{r}^{ef}(\mathbf{q}_r, \mathbf{q}_e) \approx \mathbf{r}^{ef}(\mathbf{y}) \quad \text{with } y_1 = s_1, \quad y_2 = \alpha + \boldsymbol{\Gamma}\mathbf{q}_e. \quad (34)$$

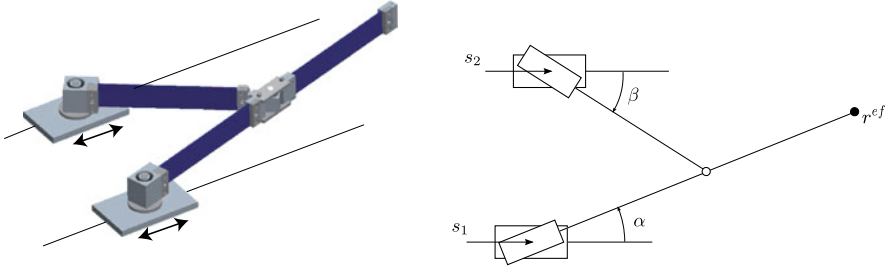


Fig. 5 Parallel flexible manipulator

The weighting matrix Γ can be derived similarly to Sect. 3. The transformation into input-output normal-form shows, that it is again a system of vector relative degree $\mathbf{r} = \{2, \dots, 2\}$ and the elastic coordinates \mathbf{q}_e describe the internal dynamics. Also in this case the weighting matrix Γ derived from geometric considerations yields a non-minimum phase system and requires a stable inversion. A simulation result is presented in Sect. 7.

6 Model Inversion Using Servo-Constraints and Projections

In this section an alternative approach for the solution of the exact model inversion problem is introduced, which is especially appealing for parallel flexible manipulators. Formulating the complete problem as a set of differential-algebraic equations according to

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{k}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{B}(\mathbf{q})\mathbf{u} + \mathbf{C}^T(\mathbf{q})\boldsymbol{\lambda} \quad (35a)$$

$$\mathbf{h}(\mathbf{q}, t) = \mathbf{y}(\mathbf{q}) - \mathbf{y}_d(t) = \mathbf{0} \quad (35b)$$

$$\mathbf{c}(\mathbf{q}, t) = \mathbf{0}, \quad (35c)$$

with the vector $\mathbf{c} \in \mathbb{R}^{n_c}$ representing the constraints due to the kinematic loops and the vector $\mathbf{h} \in \mathbb{R}^m$ describing the so-called servo-constraints [1, 14]. This unifies the handling of both types of constraints. Due to the fact that the constraint equations are on position level, it is not guaranteed that a numerical solution can be found easily. In order to ensure that a solution can be obtained, a formalism to reduce the index of the system without imposing a numerical drift is presented.

6.1 Differentiation Index

A very important characteristic of differential-algebraic equations is the differentiation index. It describes how a perturbation in the variables effects the solution of the DAE. The differentiation index is defined as the minimal number of differentiations

of the constraint equations necessary to obtain ordinary differential equations for all variables. To ensure a better readability, the constraints in Eq. (35b) and Eq. (35c) are combined leading to the representation

$$\mathbf{M}(q)\ddot{q} + \mathbf{k}(q, \dot{q}) = \mathbf{g}(q, \dot{q}) + \Theta(q) \begin{bmatrix} u \\ \lambda \end{bmatrix} \quad (36a)$$

$$\xi(q, t) = \mathbf{0} \quad (36b)$$

with the matrix $\Theta = [\mathbf{B} \ \mathbf{C}^T] \in \mathbb{R}^{f \times (m+n_c)}$ and the vector $\xi^T = [\mathbf{h}^T \ \mathbf{c}^T]^T \in \mathbb{R}^{(m+n_c)}$ of the constraints. Firstly, the constraint equations are differentiated with respect to the time

$$\begin{aligned} \frac{d\xi}{dt} &= \frac{\partial \xi}{\partial q} \dot{q} + \frac{\partial \xi}{\partial \lambda} \dot{\lambda} + \frac{\partial \xi}{\partial u} \dot{u} + \frac{\partial \xi}{\partial t} = \mathbf{0} \\ \dot{\xi} &= \begin{bmatrix} \mathbf{H} \\ \mathbf{C} \end{bmatrix} \dot{q} + \xi' = \Xi \dot{q} + \xi' = \mathbf{0}, \end{aligned} \quad (37)$$

in which the matrix $\Xi = [\mathbf{H}^T, \mathbf{C}^T]^T \in \mathbb{R}^{(m+n_c) \times f}$ summarized the Jacobian matrices of the servo-constraints \mathbf{H} and geometric constraints \mathbf{C} . As the constraint equations do neither depend on the reaction forces λ nor on the system inputs u , the Jacobian matrix of the unified constraints ξ with respect to λ and u vanish. Since Eq. (37) also does not depend on λ or u , the second partial differentiation can be displayed as

$$\ddot{\xi} = \Xi \ddot{q} + \xi'' = \mathbf{0}. \quad (38)$$

By solving Eq. (36a) for the generalized accelerations \ddot{q} and inserting them into Eq. (38), the constraint equations on velocity level can be presented as

$$\ddot{\xi} = \Xi(q)\mathbf{M}(q)^{-1} \left(\mathbf{g}(q, \dot{q}) - \mathbf{k}(q, \dot{q}) + \Theta(q) \begin{bmatrix} u \\ \lambda \end{bmatrix} \right) + \xi'' = \mathbf{0}. \quad (39)$$

The final differentiation yields to the formulation

$$\begin{aligned} \ddot{\xi} &= \frac{\partial \ddot{\xi}}{\partial q} \dot{q} + \frac{\partial \ddot{\xi}}{\partial \dot{q}} \ddot{q} + \frac{\partial \ddot{\xi}}{\partial u} \dot{u} + \frac{\partial \ddot{\xi}}{\partial \lambda} \dot{\lambda} + \frac{\partial \ddot{\xi}}{\partial t} \\ &= \frac{\partial \ddot{\xi}}{\partial q} \dot{q} + \frac{\partial \ddot{\xi}}{\partial \dot{q}} \ddot{q} + \Xi(q)\mathbf{M}(q)^{-1} \Theta(q) \begin{bmatrix} \dot{u} \\ \dot{\lambda} \end{bmatrix} + \frac{\partial \ddot{\xi}}{\partial t} = \mathbf{0} \end{aligned} \quad (40)$$

providing a set of differential equations for the generalized reaction forces λ and the system inputs u , if the matrix $\mathbf{R} = \Xi \mathbf{M}^{-1} \Theta$ is regular. In this case, the system has a differentiation index of 3. If the matrix \mathbf{R} is not invertible, the differentiation index is accordingly higher. For the special case, that the system has only independent geometric constraints, \mathbf{R} reduces to $\mathbf{R} = \mathbf{C} \mathbf{M}^{-1} \mathbf{C}^T$ which is positive definite and therewith invertible. If only servo-constraints occur it is $\mathbf{R} = \mathbf{H} \mathbf{M}^{-1} \mathbf{B}$. This is exactly the decoupling matrix occurring in Eq. (4) of the symbolic coordinate transformation into input-output normal-form. Thus, if \mathbf{R} is invertible the system has differentiation index 3 and vector relative degree $\mathbf{r} = \{2, \dots, 2\}$. In [2] it is discussed that the differentiation index is one higher than the relative degree, if the

internal dynamics are not affected by a constraint. In the following it is assumed, that servo-constraints and geometric constraints occur and that the matrix \mathbf{R} has full rank, providing differentiation index 3.

6.2 Projections

In order to reduce the index of the system, the equations of motion (36a) and the constraint equations on acceleration level (39) are taken into account. The first step is eliminating the generalized reaction forces λ and the system inputs \mathbf{u} with a projection. For classical DAEs this method is sometimes referred as the null space method, see [7]. For servo-constraint problems, the corresponding projection matrices are obtained with two QR decompositions according to

$$\mathbf{\Xi}^T = \mathbf{Q}_r \mathbf{R}_r = [\mathbf{Q}_{r,1} \quad \mathbf{J}_r] \begin{bmatrix} \mathbf{R}_{r,1} \\ \mathbf{0} \end{bmatrix} = \mathbf{Q}_{r,1} \mathbf{R}_{r,1}, \quad (41a)$$

$$\mathbf{\Theta} = \mathbf{Q}_l \mathbf{R}_l = [\mathbf{Q}_{l,1} \quad \mathbf{J}_l] \begin{bmatrix} \mathbf{R}_{l,1} \\ \mathbf{0} \end{bmatrix} = \mathbf{Q}_{l,1} \mathbf{R}_{l,1}. \quad (41b)$$

The matrix $\mathbf{\Xi}^T$ is split into an orthogonal $\mathbf{Q}_r \in \mathbb{R}^{f \times f}$ and a triangular matrix $\mathbf{R}_r \in \mathbb{R}^{f \times (m+n_c)}$, whose $f - (m+n_c)$ lower rows are zero. Therefore, the matrix $\mathbf{\Xi}^T$ can be displayed as the product of the first $(m+n_c)$ columns of \mathbf{Q}_r , called $\mathbf{Q}_{r,1}$, and the first $(m+n_c)$ rows of \mathbf{R}_r , called $\mathbf{R}_{r,1}$. With this relationship, the matrix $\mathbf{\Xi}$ can be written as

$$\mathbf{\Xi} = \mathbf{R}_{r,1}^T \mathbf{Q}_{r,1}^T. \quad (42)$$

Due to the fact that the matrix \mathbf{Q}_r is orthogonal, the matrix product

$$\mathbf{\Xi} \mathbf{J}_r = \mathbf{R}_{r,1}^T \underbrace{\mathbf{Q}_{r,1}^T \mathbf{J}_r}_{=\mathbf{0}} \quad (43)$$

vanishes. Therefore, the columns of the matrix \mathbf{J}_r span the null space of the matrix $\mathbf{\Xi}$ and the columns of the matrix $\mathbf{Q}_{r,1}$ span the row space of the matrix $\mathbf{\Xi}$, respectively. Since the dimensions of $\mathbf{\Xi}^T$ and $\mathbf{\Theta}$ match, an analog procedure shows that the columns of the matrix \mathbf{J}_l span the left null space of the matrix $\mathbf{\Theta}$ and the columns of the matrix $\mathbf{Q}_{l,1}$ span the column space of the matrix $\mathbf{\Theta}$.

The properties of the matrix \mathbf{Q}_r are used to introduce a new set of generalized accelerations $\ddot{\mathbf{z}}$ according to

$$\ddot{\mathbf{q}} = \mathbf{Q}_r \ddot{\mathbf{z}} = \mathbf{J}_r \ddot{\mathbf{z}}_i + \mathbf{Q}_{r,1} \ddot{\mathbf{z}}_d, \quad (44)$$

with the independent accelerations $\ddot{\mathbf{z}}_i \in \mathbb{R}^{f-(m+n_c)}$ and the dependent accelerations $\ddot{\mathbf{z}}_d \in \mathbb{R}^{(m+n_c)}$. Substituting the generalized accelerations in Eq. (38) with Eq. (44) results in

$$\begin{aligned}\ddot{\xi} &= \Xi(J_r \ddot{z}_i + Q_{r,1} \ddot{z}_d) + \xi'' \\ &= R_{r,1}^T \underbrace{Q_{r,1}^T J_r \ddot{z}_i}_{=0} + R_{r,1}^T \underbrace{Q_{r,1}^T Q_{r,1} \ddot{z}_d}_{=I} + \xi'' = \mathbf{0}.\end{aligned}\quad (45)$$

Solving Eq. (45) for the dependent accelerations \ddot{z}_d leads to

$$\ddot{z}_d = -R_{r,1}^{-T} \xi''. \quad (46)$$

Therefore, using Eq. (44), the generalized coordinates can be expressed as

$$\ddot{q} = J_r \ddot{z}_i - Q_{r,1} R_{r,1}^{-T} \xi''. \quad (47)$$

The next step is to eliminate the reaction forces λ and system inputs u and to express the equations of motion in terms of the independent accelerations \ddot{z}_i . By substituting the generalized accelerations \ddot{q} in Eq. (36a) with Eq. (47) and multiplying the arising equations with the transposed Jacobian matrix J_l from the left yields to

$$J_l^T M(q)(J_r \ddot{z}_i - Q_{r,1} R_{r,1}^{-T} \xi'') + J_l^T k(q, \dot{q}) = J_l^T g(q, \dot{q}) + \underbrace{J_l^T \Theta(q)}_{=0} \begin{bmatrix} u \\ \lambda \end{bmatrix}. \quad (48)$$

This leads to the equations of motion in the new, independent coordinates z_i . These equations describe the internal dynamics of the servo-constraint problem given in Eqs. (35a)–(35c). This corresponds to the internal dynamics given by Eq. (9) using symbolic coordinate transformation. In order to solve the initial value problem, these equations have to be transformed back to the original set of coordinates q . Solving Eq. (48) for independent accelerations \ddot{z}_i according to

$$\ddot{z}_i = (J_l^T M(q) J_r)^{-1} J_l^T (g(q, \dot{q}) - k(q, \dot{q}) + M(q) Q_{r,1} R_{r,1}^{-T} \xi'') \quad (49)$$

and inserting the independent accelerations \ddot{z}_i into Eq. (47) leads to

$$\ddot{q} = J_r (J_l^T M(q) J_r)^{-1} J_l^T (g - k + M(q) Q_{r,1} R_{r,1}^{-T} \xi'') - Q_{r,1} R_{r,1}^{-T} \xi''. \quad (50)$$

In order to derive a state space representation, the same procedure is done for the velocities. Similar to Eq. (47), the independent velocities \dot{z}_i can be expressed as

$$\dot{q} = J_r \dot{z}_i - Q_{r,1} R_{r,1}^{-T} \xi'. \quad (51)$$

Multiplying Eq. (51) with the transposed Jacobian matrix J_l from the left and solving for \dot{z}_i results in

$$\dot{z}_i = (J_l^T J_r)^{-1} J_l^T (\dot{q} + Q_{r,1} R_{r,1}^{-T} \xi'). \quad (52)$$

Inserting this equation into Eq. (51) results in the representation necessary for state space representation

$$\dot{q} = J_r (J_l^T J_r)^{-1} J_l^T (\dot{q} + Q_{r,1} R_{r,1}^{-T} \xi') - Q_{r,1} R_{r,1}^{-T} \xi', \quad (53)$$

in which the velocities do not violate the constraint equations. One of the benefits of this representation is, that the equations of motion no longer dependent on the

reaction forces nor on the system inputs and at the same time comply with the constraints. But the back transformation also introduced $2(m + f_c)$ zero eigenvalues leading to a numerical drift in the solution [3]. The same effect occurs, when the Lagrange equations of the first kind are applied to constraint mechanical systems.

A suitable way of avoiding the drift is to reformulate the system given by Eq. (50) and Eq. (53) as an index 1-DAE by only considering $2(f - (m + n_c))$ differential equations, which describe the internal dynamics, and taking the remaining $2(m + f_c)$ coordinates as algebraic equations into account. The generalized coordinates are split according to $\mathbf{q}^T = [\mathbf{q}_i^T \ \mathbf{q}_d^T]$ with \mathbf{q}_i being the coordinates describing the internal dynamics and \mathbf{q}_d being the coordinates describing the driven dynamics. From the analysis in Sect. 5 it is known that the internal dynamics can be described by the elastic coordinates and thus it is $\mathbf{q}_d = \mathbf{q}_e$. Then, the equations of motion of the servo-constraint problem in state space representation can be expressed as

$$\begin{aligned} & \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_i \\ \ddot{\mathbf{q}}_i \\ \mathbf{q}_d \\ \dot{\mathbf{q}}_d \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{P}(\mathbf{J}_r(\mathbf{J}_l^T \mathbf{J}_r)^{-1} \mathbf{J}_l^T (\dot{\mathbf{q}} + \mathbf{Q}_{r,1} \mathbf{R}_{r,1}^{-T} \xi') - \mathbf{Q}_{r,1} \mathbf{R}_{r,1}^{-T} \xi') \\ \mathbf{P}(\hat{\mathbf{M}}(\mathbf{g} - \mathbf{k} + \mathbf{M}(\mathbf{q}) \mathbf{Q}_{r,1} \mathbf{R}_{r,1}^{-T} \xi'') - \mathbf{Q}_{r,1} \mathbf{R}_{r,1}^{-T} \xi'') \\ \xi(\mathbf{q}, t) \\ \dot{\xi}(\dot{\mathbf{q}}, \mathbf{q}, t) \end{bmatrix}, \quad (54) \end{aligned}$$

in which the matrix \mathbf{P} represents the Jacobian matrix of the vector \mathbf{q}_i with respect to the generalized coordinates \mathbf{q} and the matrix $\hat{\mathbf{M}}$ equals $\mathbf{J}_r(\mathbf{J}_l^T \mathbf{M}(\mathbf{q}) \mathbf{J}_r)^{-1} \mathbf{J}_l^T$. The numerical integration of these equations as an initial value problem does not provoke a drift in the solution.

In analogy to Sect. 2.2 the desired system inputs \mathbf{u}_d together with the reaction forces λ , which both are eliminated in Eq. (54), can be derived by solving Eq. (36a) for the generalized accelerations and substituting them with Eq. (47) according to

$$\mathbf{J}_r \ddot{\mathbf{z}}_i - \mathbf{Q}_{r,1} \mathbf{R}_{r,1}^{-T} \xi'' = \mathbf{M}(\mathbf{q})^{-1} (\mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{k}(\mathbf{q}, \dot{\mathbf{q}})) + \mathbf{M}(\mathbf{q})^{-1} \Theta(\mathbf{q}) \begin{bmatrix} \mathbf{u} \\ \lambda \end{bmatrix}. \quad (55)$$

Multiplying this equation from the left with the transposed matrix $\mathbf{Q}_{r,1}$ yields, in case of an index 3 problem, a regular matrix $\mathbf{Q}_{r,1}^T \mathbf{M}^{-1} \Theta$. Thus solving for the desired reaction forces and system inputs leads to

$$\begin{bmatrix} \mathbf{u}_d \\ \lambda \end{bmatrix} = -(\mathbf{Q}_{r,1}^T \mathbf{M}^{-1} \Theta)^{-1} (\mathbf{R}_{r,1}^{-T} \xi'' + \mathbf{Q}_{r,1}^T \mathbf{M}^{-1} (\mathbf{g} - \mathbf{k}) - \underbrace{\mathbf{Q}_{r,1}^T \mathbf{J}_r \ddot{\mathbf{z}}_i}_{=0}). \quad (56)$$

6.3 Solution of the Two-Point Boundary Value Problem

In order to solve the two-point boundary value problem for this approach, some additional considerations have to be taken into account. Firstly, the question arises

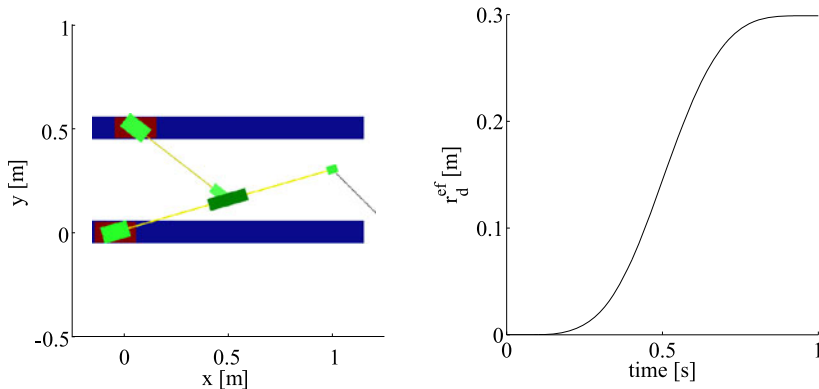


Fig. 6 Desired trajectory of the parallel manipulator

which representation is supposed to be used to solve the boundary value problem. There are three qualified candidates providing different advantages and drawbacks. The first candidate is the representation describing solely the internal dynamics of the model described by the first two rows of Eq. (54). By using this representation the formulation of the boundary conditions follows the description in Sect. 2, but it is necessary to compute the coordinates of the driven dynamics with a root search at each time step.

If a corresponding solver is available, the differential-algebraic representation according to Eq. (54) can be used to obtain a bounded solution. The third possibility is using the projected dynamics according to Eqs. (50) and (53). In this case, the numerical overhead is considerably reduced. The last two candidates have in common that the formulation of the boundary conditions has to be modified. In addition to the already mentioned boundary conditions based on the eigenspaces of the internal dynamics, it is necessary to use a combination of the constraint equations on position and velocity level at the time t_0 or t_f as well. A reasonable choice is ensuring that the solution fulfills the constraint equations on position level at t_0 and t_f .

7 Application to Parallel Manipulators

The model inversion formalisms using coordinate transformation presented in Sects. 2 and 5 and servo-constraints presented in Sect. 6 are applied to the parallel flexible manipulator shown in Fig. 5. The trajectory presented in Fig. 6 is used. The system consists of a long and a short arm each mounted on a car. The long arm is composed of three rigid parts connected with two elastic links and the short arm is composed of two rigid parts connected with an elastic link. The three identical elastic links, which are made out of steel, have length $l_e = 400$ mm, height $h_e = 80$ mm and depth $d_e = 2$ mm. The overall length of the long arm is $l_l = 1081$ mm, whereas the length of the short arm is $l_s = 560$ mm. The end of the short arm is connected

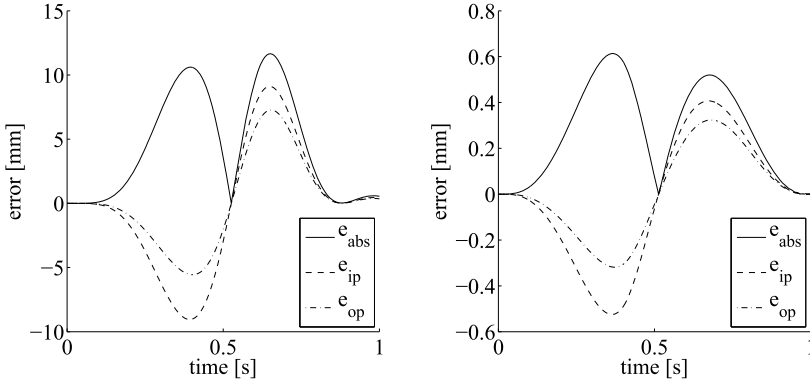


Fig. 7 Error of the end-effector trajectory with rigid output (*left*) and using optimized weights and classical inversion (*right*)

to the middle of the long arm by a revolute joint. The long arm is modeled as one elastic body consisting of two beam elements which are connected with rigid bodies. A model order reduction of the long arm based on proper orthogonal decomposition, see [5], results in a reduced elastic body with twelve shape functions to describe the elastic deformations. A similar procedure leads to a reduced model of the short arm with six shape functions. Due to the revolute joint the elastic deformations of the long arm are described in a secant floating frame of reference, whereas the deformations of the short arm are described in a tangent floating frame of reference. Therefore, the kinematics of the system with cut kinematic loop can be described with four rigid coordinates $\mathbf{q}_r = [s_1 \ s_2 \ \alpha \ \beta]^T$ and 18 elastic coordinates \mathbf{q}_e . In analogy to the serial manipulator, the end-effector of the long arm is supposed to follow a straight test trajectory, see Fig. 6. The end-effector point is the system output and can be approximated using the system output

$$\mathbf{y} = \begin{bmatrix} s_1 \\ 0 \end{bmatrix} + \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \left(\begin{bmatrix} l_{long} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \sum_{i=1}^{f_e} w_i \Phi_i q_{ei} \end{bmatrix} \right), \quad (57)$$

where Φ_i is the i th shape function evaluated at the end-effector point. For $w_i = 1$ the exact end-effector position is obtained. In this case the system is non-minimum phase, and an output relocation can be performed for obtaining a minimum phase system. In this case the weights w_i are used as the design parameters for the optimization as presented in Sect. 3.2. For the servo-constraint approach the system output (57) is used, while for the coordinate transformation approach the linearly combined system output (34) is used.

Four different cases are studied. First of all, the servo-constraint approach is used and the weights w_i are all set to zero leading to an output omitting the elastic deformations, corresponding to a rigid output. In this case, the internal dynamics are stable and the forward integration of Eq. (54) can be used. The resulting error of the end-effector trajectory tracking is presented in Fig. 7, which shows a very large deviation of approximately 12 mm.

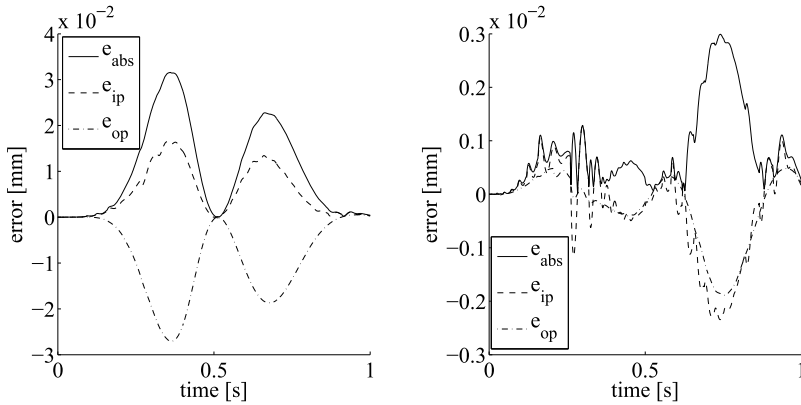


Fig. 8 Error of the end-effector trajectory using coordinate transformation (*left*) and servo constraints (*right*)

In order to improve the achievable accuracy of the trajectory tracking problem, an optimization of the weights is performed. Therefore, the algorithm presented in Sect. 3.2 is applied to the system dynamics described in Eq. (54). In this case, the design parameters \mathbf{p} are the weights \mathbf{w} , which are varied from -1 to 1 . The optimized output yields to a minimum phase system with a minimal trajectory tracking error. Then, for feedforward control design the servo-constraint approach is used in combination with forward integration. Figure 8 shows, that the error obtained is about 0.6 mm and thus less than a tenth of the error obtained with the rigid output.

Next, the stable inversion is applied to the system with end-effector point as system output, which has an unbounded internal dynamics. First, the coordinate transformation approach is used with linearly combined output (34), to approximate the end-effector point. The boundary value problem is solved with the Matlab solver *bvp5c*. Figure 8 shows the obtained error of the end-effector trajectory, which is around 0.03 mm. While the linearly combined system output is reproduced nearly exactly, this small tracking error of the end-effector point originates from its approximation by the linearly combined output.

Finally, the servo-constraint approach is used with exact output (57). Unlike the previous computations using the servo-constraint approach, only the differential part of Eq. (54) is considered, because the boundary value problem solver *bvp5c* does not support differential-algebraic equations. The error of the solution obtained by the boundary value problem is presented in Fig. 8. The maximal discrepancy is about 0.003 mm. Unlike the other cases, this error represents solely the solver tolerance. In summary, both stable inversion based approaches yield nearly exact reproduction of the end-effector trajectory.

8 Summary

The derivation of feedforward control designs for serial and flexible manipulators were presented. Firstly, exact inverse model based on concepts from differential geometric control theory were used and applied to serial and parallel flexible manipulators. It was shown, that the stability properties of the internal dynamics determines the complexity of the feedforward control design. By output optimization stable internal dynamics can be obtained, while keeping the end-effector tracking error small. In addition an alternative approach for feedforward control based on servo-constraints was presented and applied to a parallel flexible manipulator. By using numerical projection into the unconstrained subspace the description of the internal dynamics is obtained, while its differentiation index is reduced. Then, for the solution the same concepts as in the first feedforward control approach can be used. Both approaches provide powerful tools to design accurate feedforward control for flexible manipulators

Acknowledgements The authors would like to thank the German Research Foundation (DFG) for financial support of this work within the Cluster of Excellence in Simulation Technology (EXC 310/1) at the University of Stuttgart.

References

1. Blajer, W., Kolodziejczyk, K.: A geometric approach to solving problems of control constraints: theory and a DAE framework. *Multibody Syst. Dyn.* **11**, 343–364 (2004)
2. Campbell, S.: High-index differential algebraic equations. *Mech. Based Des. Struct. Mach.* **23**(2), 199–222 (1995)
3. Campbell, S., Leimkuhler, B.: Differentiation of constraints in differential-algebraic equations. *Mech. Based Des. Struct. Mach.* **19**(1), 19–39 (1991)
4. Devasia, S., Chen, D., Paden, B.: Nonlinear inversion-based output tracking. *IEEE Trans. Autom. Control* **41**(7), 930–942 (1996)
5. Fehr, J., Lehner, M., Eberhard, P.: On the automated generation of reduced order models in flexible multibody dynamics. In: *Proceedings in Applied Mathematics and Mechanics* (2007)
6. Hirschorn, R.: Invertibility of multivariable nonlinear control systems. *IEEE Trans. Autom. Control* **24**, 855–865 (1979)
7. Ider, S., Amirouche, F.: Coordinate reduction in the dynamics of constrained multibody systems—a new approach. *J. Appl. Mech.* **55**(4), 899–904 (1988)
8. Isidori, A.: *Nonlinear Control Systems*. Springer, London (1995)
9. Kurz, T., Eberhard, P., Henninger, C., Schiehlen, W.: From Neweul to Neweul-M²: symbolical equations of motion for multibody system analysis and synthesis. *Multibody Syst. Dyn.* **24**, 25–41 (2010)
10. Leister, G., Bestle, D.: Symbolic-numerical solution of multibody systems with closed loops. *Veh. Syst. Dyn.* **21**(1), 129–142 (1992)
11. Moallem, M., Patel, R., Khorasani, K.: *Flexible-Link Robot Manipulators: Control Techniques and Structural Design*. Springer, London (2000)
12. Sastry, S.: *Nonlinear Systems: Analysis, Stability and Control*. Springer, New York (1999)
13. Sedlaczek, K., Eberhard, P.: Using augmented Lagrangian particle swarm optimization for constrained problems in engineering. *Struct. Multidiscip. Optim.* **32**, 277–286 (2006)
14. Seifried, R.: Two approaches for feedforward control and optimal design of underactuated multibody systems. *Multibody Syst. Dyn.* **27**(1), 75–93 (2012)

15. Seifried, R., Held, A., Dietmann, F.: Analysis of feed-forward control design approaches for flexible multibody systems. *J. Syst. Des. Dyn. (Special Issue ACMD 2010)* **5**(3), 429–440 (2011)
16. Shabana, A.: *Dynamics of Multibody Systems*. Cambridge University Press, Cambridge (2005)
17. Taylor, D., Li, S.: Stable inversion of continuous-time nonlinear systems by finite-difference methods. *IEEE Trans. Autom. Control* **47**(3), 537–542 (2002)
18. Zhao, H., Chen, D.: Tip trajectory tracking for multilink flexible manipulators using stable inversion. *J. Guid. Control Dyn.* **21**(2), 314–320 (1998)