

# Inverse Dynamics of Flexible Manipulators

Stig Moberg, Sven Hanssen

Division of Automatic Control

E-mail: [stig@isy.liu.se](mailto:stig@isy.liu.se), [soha@kth.se](mailto:soha@kth.se)

8th March 2010

Report no.: LiTH-ISY-R-2939

Accepted for publication in Multibody Dynamics 2009, 29th June -  
2nd July 2009, Warsaw, Poland

Address:

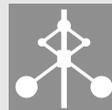
Department of Electrical Engineering

Linköpings universitet

SE-581 83 Linköping, Sweden

WWW: <http://www.control.isy.liu.se>

AUTOMATIC CONTROL  
REGLERTEKNIK  
LINKÖPINGS UNIVERSITET



## Abstract

High performance robot manipulators, in terms of cycle time and accuracy, require well designed control methods, based on accurate dynamic models. Robot manipulators are traditionally described by the flexible joint model or the flexible link model. These models only consider elasticity in the rotational direction. When these models are used for control or simulation, the accuracy can be limited due to the model simplifications, since a real manipulator has a distributed flexibility in all directions. This work investigates different methods for the inverse dynamics of a more general manipulator model, called the extended flexible joint model. The inverse dynamics solution is needed for feed-forward control, which is often used for high-precision robot manipulator control.

The inverse dynamics of the extended flexible joint model can be computed as the solution of a high-index differential algebraic equation (DAE). One method is to solve the discretized DAE using a constant-stepsize constant-order backwards differentiation formula (BDF). This work shows that there is only a small difference between solving the original high-index DAE and the index-reduced DAE. It is also concluded that scaling of the algebraic equations and their derivatives is important.

The inverse dynamics can be solved as an initial-value problem if the zero dynamics of the system is stable, i.e., minimum phase. For unstable zero dynamics, an optimization approach based on the discretized DAE is suggested. An optimization method, using a continuous DAE formulation, is also suggested and evaluated. The solvers are illustrated by simulation, using a manipulator with two actuators and five degrees-of-freedom.

**Keywords:** Manipulator, Control, Differential Algebraic Equation, Flexible Multibody Dynamics, Non-minimum Phase, Inverse Dynamics.

# INVERSE DYNAMICS OF FLEXIBLE MANIPULATORS

Stig Moberg<sup>\*,†</sup> and Sven Hanssen<sup>\*,‡</sup>

\*ABB AB – Robotics  
SE-721 68 Västerås, Sweden

†Division of Automatic Control, Department of Electrical Engineering  
Linköping University, SE-581 83 Linköping, Sweden  
e-mail: stig@isy.liu.se

‡Department of Solid Mechanics  
Royal Institute of Technology, SE-100 44 Stockholm, Sweden  
e-mail: soha@kth.se

**Keywords:** Manipulator, Control, Differential Algebraic Equation, Flexible Multibody Dynamics, Non-minimum Phase, Inverse Dynamics.

**Abstract.** *High performance robot manipulators, in terms of cycle time and accuracy, require well designed control methods, based on accurate dynamic models. Robot manipulators are traditionally described by the flexible joint model or the flexible link model. These models only consider elasticity in the rotational direction. When these models are used for control or simulation, the accuracy can be limited due to the model simplifications, since a real manipulator has a distributed flexibility in all directions. This work investigates different methods for the inverse dynamics of a more general manipulator model, called the extended flexible joint model. The inverse dynamics solution is needed for feedforward control, which is often used for high-precision robot manipulator control.*

*The inverse dynamics of the extended flexible joint model can be computed as the solution of a high-index differential algebraic equation (DAE). One method is to solve the discretized DAE using a constant-stepsize constant-order backwards differentiation formula (BDF). This work shows that there is only a small difference between solving the original high-index DAE and the index-reduced DAE. It is also concluded that scaling of the algebraic equations and their derivatives is important.*

*The inverse dynamics can be solved as an initial-value problem if the zero dynamics of the system is stable, i.e., minimum phase. For unstable zero dynamics, an optimization approach based on the discretized DAE is suggested. An optimization method, using a continuous DAE formulation, is also suggested and evaluated. The solvers are illustrated by simulation, using a manipulator with two actuators and five degrees-of-freedom.*

## 1 INTRODUCTION

Current industrial robot development is focused on increasing the robot performance, reducing the robot cost, improving safety, and introducing new functionalities as described in [6]. The need for cost reduction results in the use of weight- and cost optimized robot components with increased elasticity. This results in more complicated vibration modes inside the control bandwidth. To maintain or improve the robot performance, the motion control must be improved for this new generation of robots. This can, e.g., be obtained by improving the model-based control as described in [2]. An example of a modern industrial robot manipulator is shown in Fig. (1).



Figure 1: The IRB4600 robot from ABB.

Most publications concerning robot manipulators only consider elasticity in the rotational direction. If gear elasticity is considered we get the *flexible joint model*, and if link deformation restricted to a plane perpendicular to the preceding joint is included we get the *flexible link model*. These models are described in, e.g., [7] and are useful for many purposes and simplify the control design. However, these models cannot accurately describe a modern industrial robot. An extended model is needed to increase the accuracy in, e.g., simulation and control.

The inverse dynamics problem for a number of mechanical systems can be formulated and solved as a DAE, see, e.g., [3] or [4]. For many systems, the problem can be reformulated as the solution of algebraic equations or ordinary differential equations. One example of such a system is the *flexible joint model*. In [22], a general serial-link elastic manipulator model called the *extended flexible joint model* was presented, and the inverse dynamics of this model was formulated and solved as a differential-algebraic equation (DAE). A comparison of feedforward control based on the *flexible joint model* and the *extended flexible joint model* was also performed in [22], and it was shown that the extended model yields a dramatic improvement.

## 2 THE EXTENDED FLEXIBLE JOINT MODEL

The model consists of a serial kinematic chain of rigid bodies. The rigid bodies are connected by multidimensional spring-damper pairs, representing the rotational, and possibly also the translational, deflection. If two rigid bodies are joined by a motor and a gearbox, we get an actuated joint represented by one spring-damper pair, modeling the rotational deflection of the gearbox. The other spring-damper pairs represent non-actuated joints, modeling, e.g., elasticity of bearings, foundation, tool, and load as well as bending and torsion of the links. One example of this model is illustrated in Fig. (2). This model consists of three motors, three links, four rigid bodies, and five spring-damper pairs. Thus, the model has three actuated joints, two

non-actuated joints, and eight<sup>1</sup> degrees-of-freedom (DOF). Generally, if the number of added non-actuated joints is  $\Upsilon_{na}$  and the number of actuated joints is  $\Upsilon_a$ , the system has  $2\Upsilon_a + \Upsilon_{na}$  degrees-of-freedom. The model equations are<sup>2</sup>

$$M(q_a)\ddot{q}_a + c(q_a, \dot{q}_a) + g(q_a) = \begin{bmatrix} \tau_g \\ \tau_e \end{bmatrix}, \quad (1a)$$

$$\tau_g = K_g(q_m - q_g) + D_g(\dot{q}_m - \dot{q}_g), \quad (1b)$$

$$\tau_e = -K_e q_e - D_e \dot{q}_e, \quad (1c)$$

$$\tau_m - \tau_g = M_m \ddot{q}_m + f(\dot{q}_m), \quad (1d)$$

where  $q_a = [q_g \ q_e]^T$ ,  $q_g \in \mathbb{R}^{\Upsilon_a}$  is the actuated joint angular position,  $q_e \in \mathbb{R}^{\Upsilon_{na}}$  is the non-actuated joint angular position, and  $q_m \in \mathbb{R}^{\Upsilon_a}$  is the motor angular position.  $M_m \in \mathbb{R}^{\Upsilon_a \times \Upsilon_a}$  is the inertia matrix of the motors and  $M(q_a) \in \mathbb{R}^{(\Upsilon_a + \Upsilon_{na}) \times (\Upsilon_a + \Upsilon_{na})}$  is the inertia matrix for the joints. The Coriolis and centrifugal torques are described by  $c(q_a, \dot{q}_a) \in \mathbb{R}^{\Upsilon_a + \Upsilon_{na}}$ ,  $g(q_a) \in \mathbb{R}^{\Upsilon_a + \Upsilon_{na}}$  is the gravity torque, and the nonlinear friction in motor bearings and gearboxes is  $f(\dot{q}_m) \in \mathbb{R}^{\Upsilon_a}$ . The actuator torque is  $\tau_m \in \mathbb{R}^{\Upsilon_a}$ ,  $\tau_g \in \mathbb{R}^{\Upsilon_a}$  is the actuated joint torque, and  $\tau_e \in \mathbb{R}^{\Upsilon_{na}}$  is the non-actuated joint torque, i.e., the constraint torque.  $K_g$ ,  $K_e$ ,  $D_g$ , and  $D_e$  are the stiffness- and damping matrices for the actuated and non-actuated directions, with obvious dimensions. Note that the spring-damper torque also can be defined as nonlinear, see, e.g., [25]. Also note that the inertial couplings between the motors and the rigid bodies are neglected under the assumption of high gear ratio, see, e.g., [30]. For a complete model including the position and orientation of the tool,  $\theta$ , the forward kinematic model of the robot must be added. The kinematic model is a mapping of  $q_a \in \mathbb{R}^{\Upsilon_a + \Upsilon_{na}}$  to  $\theta \in \mathbb{R}^{\Upsilon_a}$ . The complete model of the robot is then described by (1) and

$$\theta = \Gamma(q_a). \quad (2)$$

Note that no unique inverse kinematics exists. This is a fact that makes the inverse dynamics problem difficult to solve.

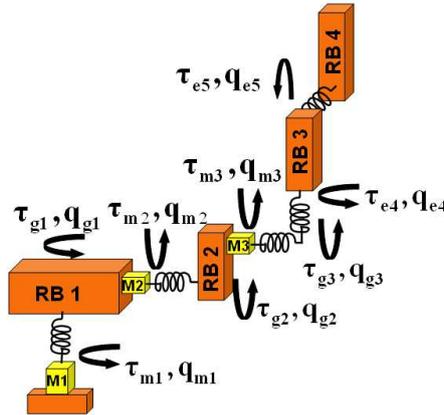


Figure 2: An extended flexible joint dynamic model with 8 degrees-of-freedom.

<sup>1</sup>The number of DOF is the number of independent coordinates necessary to specify the (internal) configuration of a system. This manipulator is specified by 3 motor angular positions, 3 actuated joint angular positions, 2 non-actuated joint angular positions, and has thus 8 DOF. The manipulator tool is specified by 6 coordinates (position and orientation) but only 3 coordinates can be controlled using this manipulator, e.g., the position. In this paper, the number of DOF refers to the manipulator system, and not to the tool.

<sup>2</sup>All positions, inertias, and torques are expressed on the joint side of the gearbox, hence, the gear ratio does not appear in the equations.

The equations of motion (1) can be derived by computing the linear and angular momentum. By using Kane's method [15] the projected equations of motion can be derived to yield a system of ordinary differential equations (ODE) with minimum number of DOF. One alternative way of deriving the equations of motion is to compute the potential and kinetic energy and to use Lagrange's equation, see, e.g., [29]. If some parameters of the *extended flexible joint model* are unknown, these parameters can be obtained by identification, see, e.g., [32]. From [32], it is also clear that the *extended flexible joint model* is needed to accurately describe a modern industrial robot.

### 3 INVERSE DYNAMICS FOR THE SIMPLIFIED FLEXIBLE JOINT MODEL

The *flexible joint model* can be derived as a special case of the *extended flexible joint model* if all non-actuated joints are removed, i.e.,  $\Upsilon_{na} = 0$ . If friction and damping are neglected we get the *simplified flexible joint model*. The equations describing this model are

$$M(q_a)\ddot{q}_a + c(q_a, \dot{q}_a) + g(q_a) - K_g(q_m - q_a) = 0, \quad (3a)$$

$$M_m\ddot{q}_m + K_g(q_m - q_a) = \tau_m. \quad (3b)$$

The inverse dynamics of this model is described, e.g., in [14]. By solving (3a) for  $q_m$  and differentiating twice, with respect to time, we get<sup>3</sup>

$$q_m = q_a + K_g^{-1}(M(q_a)\ddot{q}_a + c(q_a, \dot{q}_a) + g(q_a)), \quad (4a)$$

$$\begin{aligned} \ddot{q}_m = \ddot{q}_a + K_g^{-1}(\ddot{M}(q_a, \dot{q}_a, \ddot{q}_a)\ddot{q}_a + 2\dot{M}(q_a, \dot{q}_a)q_a^{[3]} + M(q_a)q_a^{[4]} + \\ \ddot{c}(q_a, \dot{q}_a, \ddot{q}_a, q_a^{[3]}) + \ddot{g}(q_a, \dot{q}_a, \ddot{q}_a)), \end{aligned} \quad (4b)$$

and by solving (3a) for the spring torque  $K_g(q_m - q_a)$  and inserting this expression together with (4b) in (3b) we get an expression for the control signal, i.e., the motor torque, as

$$\begin{aligned} \tau_m = (M(q_a) + M_m)\ddot{q}_a + c(q_a, \dot{q}_a) + g(q_a) + M_m K_g^{-1}[\ddot{M}(q_a, \dot{q}_a, \ddot{q}_a)\ddot{q}_a + 2\dot{M}(q_a, \dot{q}_a)q_a^{[3]} + \\ M(q_a)q_a^{[4]} + \ddot{c}(q_a, \dot{q}_a, \ddot{q}_a, q_a^{[3]}) + \ddot{g}(q_a, \dot{q}_a, \ddot{q}_a)]. \end{aligned} \quad (5)$$

The *simplified flexible joint model* is an example of a so-called *flat system* (see, e.g., [28]) which can be defined as a system where all state variables and control inputs can be expressed as an algebraic function of the desired trajectory and its derivatives up to a certain order. The desired joint trajectory  $q_a(t)$  can be computed from the desired Cartesian trajectory  $\theta(t)$ , including derivatives, by use of the inverse kinematics. The inverse dynamics can be used for *feedforward control* or *feedback linearization* as described in, e.g., [23]. The *simplified flexible joint model* has no zero dynamics, see, e.g., [7]. This means that it is always possible to find a bounded causal inverse dynamics solution. The inverse dynamics is more complicated if the model is extended with damping, friction, nonlinear spring torque, or inertial coupling between motors and rigid bodies. Approaches for handling these cases are, e.g., described in [7] and [31].

### 4 INVERSE DYNAMICS FOR THE EXTENDED FLEXIBLE JOINT MODEL

With the following states

$$X = [q_g \quad q_e \quad q_m \quad \dot{q}_g \quad \dot{q}_e \quad \dot{q}_m] ^T = [X_1 \quad X_2 \quad X_3 \quad X_4 \quad X_5 \quad X_6] ^T,$$

<sup>3</sup>The  $n^{th}$  time derivative of a variable  $x$  is denoted  $x^{[n]}$ .

(1) and (2) can be written as

$$\begin{aligned} M_{11}(X_1, X_2)\dot{X}_4 + M_{12}(X_1, X_2)\dot{X}_5 + \gamma_1(X_1, X_2, X_3, X_4, X_5, X_6) &= 0, \\ M_{21}(X_1, X_2)\dot{X}_4 + M_{22}(X_1, X_2)\dot{X}_5 + \gamma_2(X_1, X_2, X_4, X_5) &= 0, \\ M_m\dot{X}_6 + \gamma_3(X_1, X_3, X_4, X_6) - u &= 0, \\ \dot{X}_1 - X_4 = 0, \dot{X}_2 - X_5 = 0, \dot{X}_3 - X_6 = 0, \Gamma(X_1, X_2) - \theta(t) &= 0, \end{aligned}$$

or

$$F(t, X, \dot{X}, u) = F(t, X_1, X_2, X_3, X_4, X_5, X_6, \dot{X}_1, \dot{X}_2, \dot{X}_3, \dot{X}_4, \dot{X}_5, \dot{X}_6, u) = 0. \quad (7)$$

The system has  $\Upsilon_a$  input variables<sup>4</sup>  $u$ , i.e., the motor torque  $\tau_m$ , and  $\Upsilon_a$  controlled output variables with a desired reference  $\theta(t)$ , i.e., the position and orientation of the tool<sup>5</sup>. The gravity torque, speed dependent inertia torque, and the torque from the spring-damper pairs are contained in  $\gamma_i$ .

The inverse dynamics solution for the model is obtained if  $\theta(t)$  is regarded as the input signal and  $u$  and  $X$  are solved for. This means that the DAE (7) must be solved, and in [22] this is performed using a constant stepsize 1-step backward differentiation formula (BDF). It was observed that a short stepsize was required to obtain an accurate solution, but that numerical solvability was lost if the stepsize was too small. Some commercially available software packages, e.g., *Dymola*, were also tried on the same systems, but none was able to solve these equations if the number of DOF exceeded 5. Numerical problems could also be seen in cases where a solution was found. The 1-step BDF solver could solve systems with up to 12 DOF, although with some problems. Clearly, the DAE (7) must be further analyzed in order to improve the solver, which is one of the objectives of this work.

In this work, the inverse dynamics solution is intended to be used primarily for feedforward control, and the desired trajectory  $\theta(t)$  must be followed perfectly in the case of no model errors. If the system has a stable zero dynamics (minimum phase), see, e.g., [13], it is possible to obtain a bounded causal solution. However, if the system has an unstable zero dynamics, perfect control requires a non-causal solution, i.e., the torque must be applied before the start of the trajectory execution. Solvers for both cases will be suggested and discussed in the following sections.

## 5 DAE THEORY

As the name implies, a DAE consists of differential and algebraic equations. A DAE can generally be expressed by

$$F(t, x, \dot{x}) = 0, \quad (8)$$

where  $x \in \mathbb{R}^n$  is the state vector and  $F : \mathbb{R}^{2n+1} \rightarrow \mathbb{R}^m$ . If  $F_{\dot{x}} = \partial F / \partial \dot{x}$  is nonsingular, (8) represents an implicit ODE since  $\dot{x}$ , by the implicit function theorem, can be solved from  $x$  and  $t$ . Otherwise it represents a DAE, which in general is considerably harder to solve than an ODE. The differential index,  $\nu$ , of a DAE provides a measure of the singularity of the DAE. Generally, the higher the index, the harder the DAE is to solve. An ODE has  $\nu = 0$ , and a DAE with  $\nu > 1$  is denoted a high-index DAE. The differential index can somewhat simplified be defined according to [5]:

<sup>4</sup>The motor torque control is assumed to be perfect.

<sup>5</sup>For a kinematically redundant manipulator, e.g., a 7-axis manipulator,  $\theta$  must be extended to specify the trajectory.

**Definition 1** *The (differential) index of the DAE (8) is the minimum number of times that all or part of (8) must be differentiated with respect to  $t$  in order to determine  $\dot{x}$  as a continuous function of  $t$  and  $x$ .*

A semi-explicit DAE is a special case of (8) described as

$$\dot{x} = F(x, y), \quad (9a)$$

$$0 = G(x, y), \quad (9b)$$

where the non-differentiated variables are denoted  $y$ . Differentiation of (9b) w.r.t. time yields

$$0 = G_y(x, y)\dot{y} + G_x(x, y)\dot{x}, \quad (10)$$

and if  $G_y$  is nonsingular it is possible to solve for  $\dot{y}$  and  $\nu = 1$ . Furthermore, by the implicit function theorem, it is also possible (at least numerically) to solve for  $y = \phi(x)$ . A straightforward method for solving this DAE is, e.g., Forward Euler according to

1. Compute  $x_{t+1} = x_t + hF(x_t, y_t)$
2. Solve  $y_{t+1}$  from  $G(x_{t+1}, y_{t+1}) = 0$

where  $h$  is the stepsize. For the implicit DAE

$$F(t, x, \dot{x}, y) = 0, \quad (11)$$

we can solve for  $z = [\dot{x} \ y]^T$  if  $F_z$  is non-singular. In this case,  $\nu = 1$ , and the DAE can be solved with implicit methods.

For the analysis of a nonlinear DAE (8), the derivative array plays a central part. The derivative array  $F_l$  is defined according to [5, 17]

$$F_l(t, x, \dot{x}, \dots, x^{[l+1]}) = \begin{bmatrix} F(t, x, \dot{x}) \\ \frac{d}{dt}F(t, x, \dot{x}) \\ \vdots \\ \left(\frac{d}{dt}\right)^l F(t, x, \dot{x}) \end{bmatrix}. \quad (12)$$

Using the derivative array the differential index can be defined according to [5]:

**Definition 2** *The (differential) index of (8) is the smallest  $\nu$  such that  $F_\nu$  uniquely determines the variable  $\dot{x}$  as a continuous function of  $t$  and  $x$ .*

The differential index is a measure of the distance from the DAE to an ordinary differential equation (ODE). In [17], it is argued that a better measure would be the distance to a decoupled system of ODEs and purely algebraic equations. This measure is the strangeness index  $\mu$ . The strangeness index  $\mu$  is defined by the following hypothesis<sup>6</sup> as described in [17].

**Kunkel and Mehrmann Hypothesis** *There exist integers  $\mu$ ,  $a$ , and  $d$  such that the set*

$$\mathbb{L}_\mu = \{(t, x, \dot{x}, \dots, x^{[\mu+1]}) \in \mathbb{R}^{(\mu+2)n+1} | F_\mu(t, x, \dot{x}, \dots, x^{[\mu+1]}) = 0\} \quad (13)$$

*associated with  $F$  is nonempty and such that for every  $(t, x_0, \dot{x}_0, \dots, x_0^{[\mu+1]}) \in \mathbb{L}_\mu$ , there exists a (sufficiently small) neighborhood in which the following properties hold:*

<sup>6</sup>The case where the number of equations equals the numbers of unknowns, i.e.,  $m = n$ , is considered here.

1. We have  $\text{rank } M_\mu(t, x, \dot{x}, \dots, x^{[\mu+1]}) = (\mu+1)n - a$  on  $\mathbb{L}_\mu$  such that there exists a smooth matrix function  $Z_2$  of size  $(\mu+1)n \times a$  and pointwise maximal rank, satisfying  $Z_2^T M_\mu = 0$  on  $\mathbb{L}_\mu$ .
2. We have  $\text{rank } \hat{A}_2(t, x, \dot{x}, \dots, x^{[\mu+1]}) = a$ , where  $\hat{A}_2 = Z_2^T F_{\mu;x}$  such that there exists a smooth matrix function  $T_2$  of size  $n \times d$ ,  $d = n - a$ , and pointwise maximal rank, satisfying  $\hat{A}_2 T_2 = 0$ .
3. We have  $\text{rank } F_{\dot{x}}(t, x, \dot{x}) T_2(t, x, \dot{x}, \dots, x^{[\mu+1]}) = d$  such that there exists a smooth matrix function  $Z_1$  of size  $n \times d$  and pointwise maximal rank, satisfying  $\text{rank } Z_1^T F_{\dot{x}} T_2 = d$ .

The Jacobian  $M_\mu$  is defined according to<sup>7</sup>

$$M_\mu(t, x, \dot{x}, \dots, x^{[\mu+1]}) = F_{\mu;\dot{x}, \dots, x^{[\mu+1]}}(t, x, \dot{x}, \dots, x^{[\mu+1]}). \quad (14)$$

The strangeness index can then be defined according to

**Definition 3** Given a differential-algebraic equation as in (8), the smallest value of  $\mu$  such that  $F$  satisfies the hypothesis is called the strangeness index of (8). If  $\mu = 0$ , then the differential-algebraic equation is called strangeness-free.

An algebraic equation and an ODE have both  $\mu = 0$ . Moreover, (8) can be expressed as the reduced DAE

$$\hat{F}(t, x, \dot{x}) = \begin{bmatrix} Z_1^T F \\ Z_2^T F_\mu \end{bmatrix} = \begin{bmatrix} \hat{F}_1(t, x, \dot{x}) \\ \hat{F}_2(t, x) \end{bmatrix}. \quad (15)$$

If  $x \in \mathbb{R}^n$  is divided into the differential variables  $x_1 \in \mathbb{R}^d$  and the algebraic variables  $x_2 \in \mathbb{R}^a$ , the reduced DAE can (in principle) be expressed as the decoupled system

$$\dot{x}_1 = \hat{G}_1(t, x_1), \quad x_2 = \hat{G}_2(t, x_1). \quad (16)$$

## 6 A MANIPULATOR MODEL WITH 5 DOF

This manipulator model will be used as an example when analyzing and solving the inverse dynamics DAE (7). The model is illustrated in Fig. (3) and is a planar model with linear elasticity in the rotational direction only, and does not demonstrate the all-direction elasticity of the general model. However, it is an appropriate simple example with the desired mathematical structure. The model has two links, three rigid bodies, two actuated joints and one non-actuated joint, i.e.,  $\Upsilon_a = 2$  and  $\Upsilon_{na} = 1$ . The attributes of each joint are length  $l$ , inertia  $j$ , mass  $m$ , mass center  $\xi$ , stiffness  $k$ , damping  $d$ , and motor inertia  $J_m$  (actuated joints only). The model has 5 DOF and is described by (7) with states

$$\begin{aligned} X &= [X_1 \ X_2 \ X_3 \ X_4 \ X_5 \ X_6]^T \\ &= [[q_{g1} \ q_{g2}] \ [q_{e1}] \ [q_{m1} \ q_{m2}] \ [\dot{q}_{g1} \ \dot{q}_{g2}] \ [\dot{q}_{e1}] \ [\dot{q}_{m1} \ \dot{q}_{m2}]]^T \\ &= [q_1 \ q_2 \ q_3 \ q_4 \ q_5 \ \dot{q}_1 \ \dot{q}_2 \ \dot{q}_3 \ \dot{q}_4 \ \dot{q}_5]^T \\ &= [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8 \ x_9 \ x_{10}]^T. \end{aligned}$$

<sup>7</sup> $Q_{y_1, \dots, y_n}$  denotes the partial derivative  $\partial Q / \partial y$ , where  $y = [y_1, \dots, y_n]^T$ .

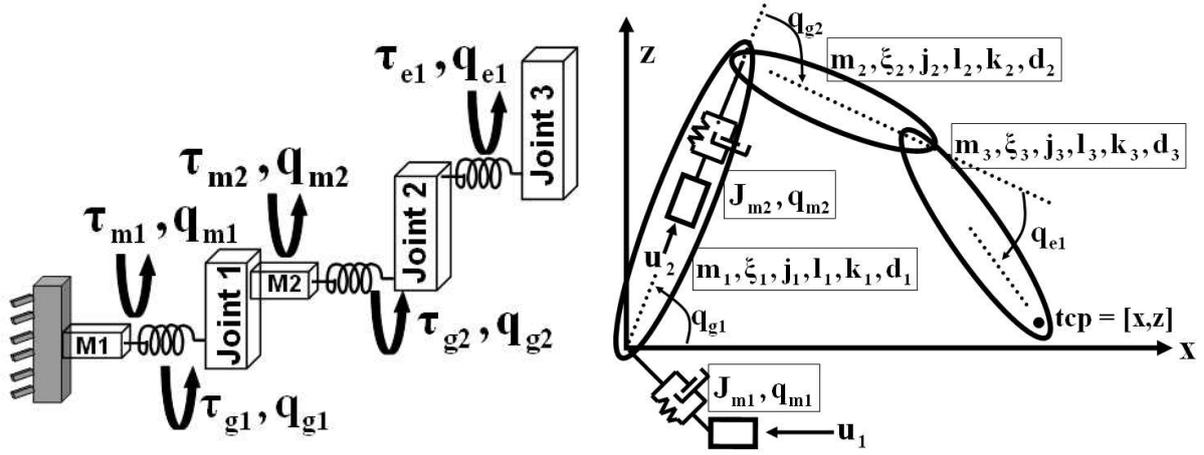


Figure 3: An extended flexible joint dynamic model with 5 DOF.

The inertia matrices  $M$  and  $M_m$  are defined and computed as<sup>8</sup>

$$M_m = \begin{bmatrix} J_{m1} & 0 \\ 0 & J_{m2} \end{bmatrix}, \quad M_{11} = \begin{bmatrix} J_{11} & J_{12} \\ J_{12} & J_{22} \end{bmatrix}, \quad M_{12} = \begin{bmatrix} J_{13} \\ J_{23} \end{bmatrix}, \quad M_{21} = [J_{13} \quad J_{23}], \quad M_{22} = [J_{33}],$$

$$J_{11} = j_1 + j_2 + j_3 + m_1 \xi_1^2 + m_2 l_1^2 s_2^2 + m_2 (c_2 l_1 + \xi_2)^2 + m_3 (s_3 c_2 l_1 + s_3 l_2 + c_3 s_2 l_1)^2$$

$$+ m_3 (c_3 c_2 l_1 + c_3 l_2 - s_3 s_2 l_1 + \xi_3)^2,$$

$$J_{12} = j_2 + j_3 + m_2 \xi_2 (c_2 l_1 + \xi_2) + m_3 l_2 s_3 (s_3 c_2 l_1 + s_3 l_2 + c_3 s_2 l_1) +$$

$$m_3 (c_3 l_2 + \xi_3) (\xi_3 - s_3 s_2 l_1 + c_3 c_2 l_1 + c_3 l_2),$$

$$J_{13} = j_3 + m_3 \xi_3 (\xi_3 - s_3 s_2 l_1 + c_3 c_2 l_1 + c_3 l_2), \quad J_{23} = j_3 + m_3 \xi_3 (c_3 l_2 + \xi_3),$$

$$J_{22} = j_2 + j_3 + m_2 \xi_2^2 + m_3 l_2^2 s_3^2 + m_3 (\xi_3 + c_3 l_2) (c_3 l_2 + \xi_3), \quad J_{33} = j_3 + m_3 \xi_3^2.$$

The kinematics is computed as:

$$\Gamma = \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} l_1 c_1 + l_2 c_{12} + l_3 c_{123} \\ -l_1 s_1 - l_2 s_{12} - l_3 s_{123} \end{bmatrix}.$$

Finally, the position and speed dependent terms are computed as:

$$\gamma_{11} = -s_2 l_1 m_2 (\dot{q}_2^2 \xi_2 + c_2 \dot{q}_1^2 l_1 + 2\dot{q}_1 \dot{q}_2 \xi_2 + \dot{q}_1^2 \xi_2) + m_2 s_2 \dot{q}_1^2 l_1 (c_2 l_1 + \xi_2) - m_3 (s_3 c_2 l_1 + s_3 l_2$$

$$+ c_3 s_2 l_1) (2\dot{q}_1 \dot{q}_2 \xi_3 + 2\dot{q}_3 \dot{q}_1 \xi_3 + \dot{q}_1^2 \xi_3 + \dot{q}_3^2 \xi_3 + 2\dot{q}_3 \dot{q}_2 \xi_3 - s_3 s_2 \dot{q}_1^2 l_1 + 2c_3 \dot{q}_1 \dot{q}_2 l_2 + c_3 \dot{q}_2^2 l_2 + \dot{q}_2^2 \xi_3$$

$$+ c_3 \dot{q}_1^2 l_2 + c_3 c_2 \dot{q}_1^2 l_1) + m_3 (\xi_3 - s_3 s_2 l_1 + c_3 c_2 l_1 + c_3 l_2) (c_3 s_2 \dot{q}_1^2 l_1 + s_3 c_2 \dot{q}_1^2 l_1 + 2s_3 \dot{q}_1 \dot{q}_2 l_2$$

$$+ s_3 \dot{q}_2^2 l_2 + s_3 \dot{q}_1^2 l_2) - m_1 g \xi_1 c_1 - m_2 g (l_1 c_1 + \xi_2 c_{12}) - m_3 g (l_1 c_1 + l_2 c_{12} + \xi_3 c_{123})$$

$$+ k_1 (q_1 - q_4) + d_1 (\dot{q}_1 - \dot{q}_4),$$

$$\gamma_{12} = \xi_2 m_2 s_2 \dot{q}_1^2 l_1 - s_3 l_2 m_3 (2\dot{q}_1 \dot{q}_2 \xi_3 + 2\dot{q}_3 \dot{q}_1 \xi_3 + \dot{q}_1^2 \xi_3 + \dot{q}_3^2 \xi_3 + 2\dot{q}_3 \dot{q}_2 \xi_3 - s_3 s_2 \dot{q}_1^2 l_1 + 2c_3 \dot{q}_1 \dot{q}_2 l_2$$

$$+ c_3 \dot{q}_2^2 l_2 + \dot{q}_2^2 \xi_3 + c_3 \dot{q}_1^2 l_2 + c_3 c_2 \dot{q}_1^2 l_1) + m_3 (\xi_3 + c_3 l_2) (c_3 s_2 \dot{q}_1^2 l_1 + s_3 c_2 \dot{q}_1^2 l_1 + 2s_3 \dot{q}_1 \dot{q}_2 l_2$$

$$+ s_3 \dot{q}_2^2 l_2 + s_3 \dot{q}_1^2 l_2) - m_2 g \xi_2 c_{12} - m_3 g (l_2 c_{12} + \xi_3 c_{123}) + k_2 (q_2 - q_5) + d_2 (\dot{q}_2 - \dot{q}_5),$$

$$\gamma_{21} = \xi_3 m_3 (s_2 s_3 \dot{q}_1^2 l_1 + 2s_3 \dot{q}_1 \dot{q}_2 l_2 + s_3 \dot{q}_2^2 l_2 + s_3 \dot{q}_1^2 l_2) - m_3 g \xi_3 c_{123} + k_3 q_3 + d_3 \dot{q}_3,$$

$$\gamma_{31} = k_1 (q_4 - q_1) + d_1 (\dot{q}_4 - \dot{q}_1), \quad \gamma_{32} = k_2 (q_5 - q_2) + d_2 (\dot{q}_5 - \dot{q}_2).$$

<sup>8</sup>The following shorthand notation is used:  $s_1 = \sin(q_1)$ ,  $c_1 = \cos(q_1)$ ,  $s_{12} = \sin(q_1 + q_2)$ ,  $c_{123} = \cos(q_1 + q_2 + q_3)$  etc.

## 7 ANALYSIS OF THE INVERSE DYNAMICS DAE

The inverse dynamics problem described in Section 4 will now be analyzed using the theory from Section 5 for the model described in Section 6. Note that the problem (7) can be expressed in the implicit DAE form (8) if  $u$  is added to the state vector  $X$  and  $\theta(t)$  is included in  $F(\cdot)$ . Clearly, for (7), the differential index  $\nu \geq 1$  since  $\dot{u}$  is missing in the equation. In [22] it is shown that the differential index is at least 4 and that it can be reduced one step by removing the equation for the motor torque balance. The control signal  $u$  can then be computed from the solution  $X$  of the reduced DAE. The reduced DAE is generally expressed as

$$M_{11}(X_1, X_2)\dot{X}_4 + M_{12}(X_1, X_2)\dot{X}_5 + \gamma_1(X_1, X_2, X_3, X_4, X_5, X_6) = 0, \quad (17a)$$

$$M_{21}(X_1, X_2)\dot{X}_4 + M_{22}(X_1, X_2)\dot{X}_5 + \gamma_2(X_1, X_2, X_4, X_5) = 0, \quad (17b)$$

$$\dot{X}_1 - X_4 = 0, \quad (17c)$$

$$\dot{X}_2 - X_5 = 0, \quad (17d)$$

$$\dot{X}_3 - X_6 = 0, \quad (17e)$$

$$\Gamma(X_1, X_2) - \theta(t) = 0. \quad (17f)$$

The analysis is performed for the 5 DOF model described in Section 6, using three different methods. All symbolical computations are performed using *Maple* and all rank computations are structural, i.e., cannot be guaranteed for all numerical values of states and state derivatives. If this analysis would be a part of a numerical solver, numerical values would have to be used.

### 7.1 Analysis using the Kunkel and Mehrmann hypothesis

The analysis was carried out by applying the hypothesis in Section 5, starting with  $\mu = 0$ , and constructing the matrix functions  $Z_2$ ,  $T_2$  (by computing nullspaces), and  $Z_1$  (by selecting matrix rows). If the hypothesis is not satisfied,  $\mu$  is increased until the hypothesis is fulfilled. The analysis showed that there are 6 algebraic variables, 4 differential variables, and that the strangeness index  $\mu = 2$ . In [24], the hypothesis is applied to an analysis of a 2 DOF manipulator. In that example, due to less complex equations, it was also possible to determine the differential and algebraic variables, and to solve for the decoupled system (16). The existence of a decoupled system is guaranteed by the implicit function theorem, but an analytic decoupling cannot be performed for a general nonlinear system.

### 7.2 Analysis by differentiation

The algebraic constraint was differentiated and the higher order derivatives solved for and substituted. After three differentiations, the Jacobian  $F_{\dot{x}}$  has (structurally) full rank. The differential index,  $\nu$ , is 3 as expected. In general, the strangeness index,  $\mu = \max(\nu - 1, 0)$ .

### 7.3 Analysis by differentiation and introduction of independent variables

This analysis, by introducing independent variables, is based on two methods for index reduction called *Minimal Extension* and *Dummy Derivatives* which are further described and referenced in Section 8.2. If the algebraic constraint (17f) is differentiated and added to the system we can add independent variables, e.g.,  $\hat{X}_1 = \dot{X}_1$  to get a determined system. By inspecting the

system, we can see that the equations involving  $\hat{X}_1$  can be removed. The resulting system is

$$M_{11}(X_1, X_2)\dot{X}_4 + M_{12}(X_1, X_2)\dot{X}_5 + \gamma_1(X_1, X_2, X_3, X_4, X_5, X_6) = 0, \quad (18a)$$

$$M_{21}(X_1, X_2)\dot{X}_4 + M_{22}(X_1, X_2)\dot{X}_5 + \gamma_2(X_1, X_2, X_4, X_5) = 0, \quad (18b)$$

$$\dot{X}_2 - X_5 = 0, \quad (18c)$$

$$\dot{X}_3 - X_6 = 0, \quad (18d)$$

$$\Gamma(X_1, X_2) - \theta(t) = 0, \quad (18e)$$

$$\dot{\Gamma}(X_1, X_2, X_4, X_5) - \dot{\theta}(t) = 0. \quad (18f)$$

By differentiating once more and adding independent variables  $\hat{X}_4 = \dot{X}_4$  we get the determined system

$$M_{11}(X_1, X_2)\hat{X}_4 + M_{12}(X_1, X_2)\dot{X}_5 + \gamma_1(X_1, X_2, X_3, X_4, X_5, X_6) = 0, \quad (19a)$$

$$M_{21}(X_1, X_2)\hat{X}_4 + M_{22}(X_1, X_2)\dot{X}_5 + \gamma_2(X_1, X_2, X_4, X_5) = 0, \quad (19b)$$

$$\dot{X}_2 - X_5 = 0, \quad (19c)$$

$$\dot{X}_3 - X_6 = 0, \quad (19d)$$

$$\Gamma(X_1, X_2) - \theta(t) = 0, \quad (19e)$$

$$\dot{\Gamma}(X_1, X_2, X_4, X_5) - \dot{\theta}(t) = 0, \quad (19f)$$

$$\ddot{\Gamma}(X_1, X_2, X_4, X_5, \hat{X}_4, \dot{X}_5) - \ddot{\theta}(t) = 0. \quad (19g)$$

By computing the Jacobian w.r.t. the highest order derivatives, e.g.,  $[X_1, \dot{X}_2, \dot{X}_3, X_4, \dot{X}_5, X_6, \hat{X}_4]$  for (19), we see that the original system (17) and the one time differentiated system (18) have singular Jacobians, and that the two times differentiated system (19) has (structurally) full rank. This analysis also shows that the (structural) differential index  $\nu$ , is 3, since we reach an index-1 DAE after two differentiations. We also see that  $x_1, x_2, x_6, x_7, x_9, x_{10}$  ( $X_1, X_4, X_6$ ) can be chosen as algebraic variables and that  $x_3, x_4, x_5, x_8$  ( $X_2, X_3, X_5$ ) can be chosen as differential variables. Thus, there are 6 algebraic variables and 4 differential variables, the same result as in the Kunkel and Mehrmann analysis, but in this way, we can also easily determine which variables are algebraic.

## 8 METHODS FOR NUMERICAL SOLUTION OF DAEs

Common numerical method for solving DAEs are Runge-Kutta methods and BDF methods (backwards differentiation formulas). In this section we will concentrate on the k-step BDF methods, where the derivative is approximated according to

$$\dot{x}(t_i) \approx D_h x_i = \frac{1}{h} \sum_{l=0}^k \alpha_l x_{i-l}, \quad (20)$$

and  $h = t_i - t_{i-1}$  is the stepsize. BDF are stable for  $1 \leq k \leq 6$  and have coefficients according to, e.g., [17]. One widely used solver for DAEs is called DASSL, which is a variable-order variable-stepsize method [5]. In this section some common approaches for solving the equations by k-step BDF will be briefly described. In Section 9, these methods will be evaluated on the extended flexible joint inverse dynamics problem.

### 8.1 Solving the original high-index DAE

If the BDF approximation is inserted in the original DAE (8) we get

$$F(t_i, x_i, D_h x_i) = 0, \quad (21)$$

which is a system of nonlinear equations that must be solved w.r.t.  $x_i$ , for each time step, forward in time (initial value problem). The solvability of this system can be analyzed by computing the Jacobian

$$F_{x_i} = \frac{\alpha_0}{h} F_{\dot{x}} + F_x \quad (22)$$

which is ill-conditioned for small stepsizes since  $F_{\dot{x}}$  is singular. This means that the solution may be erroneous or that we might not get any solution at all. However, the k-step BDF method converges for some classes of high-index DAEs, and the equations can be scaled to improve the conditioning of the problem, see [20] and [27].

## 8.2 Index Reduction and Dummy Derivatives

This method is based on index reduction by differentiation. Pantelides's algorithm [26] is often used to determine which equations to differentiate, and a Block Lower Triangular (BLT) form [11] of the differentiated problem can be used to check that the system is non-singular w.r.t. the highest order derivatives. Some problems occur when solving the differentiated problem. If all equations are kept, the resulting system is overdetermined and might be complicated to solve, see, e.g., [17]. If only the differentiated equations are kept, the resulting system with differential index  $\nu = 1$  or  $\nu = 0$  is determined, but when the system is discretized and solved, a problem denoted as *drift off* often occurs due to discretization and roundoff errors. This means that the solution drifts off from the solution manifold since the original algebraic constraints have been removed and replaced by differentiated constraints. Techniques for dealing with this are called constraint-stabilization techniques and are described in, e.g., [5].

In [21] a new method for dealing with these problems is suggested. For each differentiated equation, one differentiated variable is selected as a new independent variable, a so-called *dummy derivative*. In this way, the original constraints are kept, and a determined system is guaranteed. This method can be applied to large systems and is generally used in the numerical solvers of object-oriented simulation languages like *Modelica* [12]. In [16] a method called *minimal extension* is suggested. Minimal extension utilizes the structure of the problem to obtain a minimal system where a new independent variable have been introduced for each differentiated equation, i.e., the method is very similar to the dummy derivative method. The motivation is that the original dummy derivative method cannot handle all problems and that it generally results in an unnecessarily large system. The final numerical solution of the index-reduced system is normally solved with a k-step BDF method as described above.

## 8.3 Numerical solution based on Kunkel and Mehrmanns hypothesis

In [18] a method for numerical solution based on the hypothesis in Section 5 is suggested. The system to solve is

$$F_\mu(t_i, x_i, y_i) = 0, \quad \tilde{Z}_1^T F(t_i, x_i, D_h x_i) = 0, \quad (23)$$

where  $y_i = [\dot{x}, \dots, x^{[\mu+1]}]$  are regarded as independent algebraic variables. This underdetermined system has the same solution as the original high-index DAE. Due to the hypothesis, the dependence of  $y_i$  can be neglected such that (23) only depends on  $x_i$ .  $\tilde{Z}_1$  is an approximation to  $Z_1$  at the desired solution.

## 9 NUMERICAL SOLUTION OF THE INVERSE DYNAMICS

### 9.1 Initial conditions and trajectory generation

An ODE solution is well-defined for any initial conditions. The solution to a DAE is restricted to a space with dimension less than  $n$  by the algebraic constraint, and its derivatives up to order  $\nu - 1$ , i.e., the implicit constraints. Consistent initial conditions w.r.t. all explicit and implicit constraints must be given to avoid initial transients. For the inverse dynamics problem (7), the initial position for all DOF ( $X_{10}$ ,  $X_{20}$ ,  $X_{30}$ ) and the initial control signal ( $u_0$ ) must be chosen such that we get a steady-state solution with initial speed and all state derivatives equal to zero, i.e., we must solve  $F(0, X_{10}, X_{20}, X_{30}, 0, \dots, 0, u_0) = 0$ . If no gravity is present,  $u_0 = 0$ ,  $X_{10} = X_{30}$ , and  $X_{20} = 0$ . Note that the trajectory reference  $\theta(t)$  is implicitly included in  $F$  and that the first  $\nu$  derivatives of the algebraic constraint must be zero for  $t = 0$ . Moreover,  $\theta(t)$  must be sufficiently smooth to obtain a continuous control signal  $u(t)$  as it is (implicitly) differentiated when the DAE is solved. The smoothness requirement is that  $\theta(t)$  is at least  $\nu_0$  times differentiable, where  $\nu_0$  is the differential index of the original (unreduced) inverse dynamics problem (7). This can be accomplished, e.g., by using the trajectory polynomial

$$p(t) = a_6 t^6 + a_7 t^7 + a_8 t^8 + a_9 t^9 + a_{10} t^{10} + a_{11} t^{11},$$

which has  $p^{[i]}(0) = 0$ ,  $i = 0, \dots, 5$ . The coefficients are computed by solving

$$p(t_{end}) = 1, p^{[i]}(t_{end}) = 0, i = 1, \dots, 5.$$

A straight line trajectory in the x-z plane can for example be computed as

$$\begin{bmatrix} \theta_x(t) \\ \theta_z(t) \end{bmatrix} = \begin{bmatrix} \theta_x^{start} \\ \theta_z^{start} \end{bmatrix} + p(t) \left[ \begin{bmatrix} \theta_x^{end} \\ \theta_z^{end} \end{bmatrix} - \begin{bmatrix} \theta_x^{start} \\ \theta_z^{start} \end{bmatrix} \right]. \quad (24)$$

One example of the trajectory polynomial and its first five derivatives is shown in Fig. (4).

### 9.2 Numerical solution of the inverse dynamics for the 5 DOF model

The values of the physical parameters<sup>9</sup> are shown in Table 1. For simplicity, the gravity was set to zero, i.e., the manipulator is working in the horizontal plane.

	$m$	$\xi$	$l$	$j$	$k$	$d$	$J_m$
Joint 1 and 2	100	0.5	1	5	$10^5$	250	100
Joint 3	200	0.1	0.2	50	$10^5$	250	-

Table 1: Parameters of the 5 DOF model (minimum phase dynamics).

The numerical solver is based on a k-step BDF with constant stepsize. The Jacobian  $F_{x_i}$  is ill-conditioned for small values of  $h$ . This can be improved, e.g., by scaling the algebraic and differentiated algebraic equation with  $(1/h)^{3-l}$ , where  $l$  is the number of differentiations. The following systems were solved: Eq. (17) with index 3, Eq. (18) with index 2, and Eq. (19) with index 1. The Jacobians  $F_{x_i}$  were all full-rank, which shows that the discretized system can be solvable, although the continuous system is not. A solver based on the Kunkel and Mehrmann

<sup>9</sup>The motor inertia  $J_m$  is expressed on the joint side, i.e., multiplied by the square of the gear ratio.

Method	L33S	L33	NL33S	NL33	NL61S
Index 3	16 s	97 s	18 s	29 s	73 s
Index 2	15 s	16 s	17 s	18 s	75 s
Index 1	20 s	24 s	22 s	34 s	69 s

Table 2: Execution time: linear dynamics (L), nonlinear dynamics (NL), 3-step BDF and  $h = 3$  ms (33), 6-step BDF and  $h = 1$  ms (61), scaled algebraic equations (S)

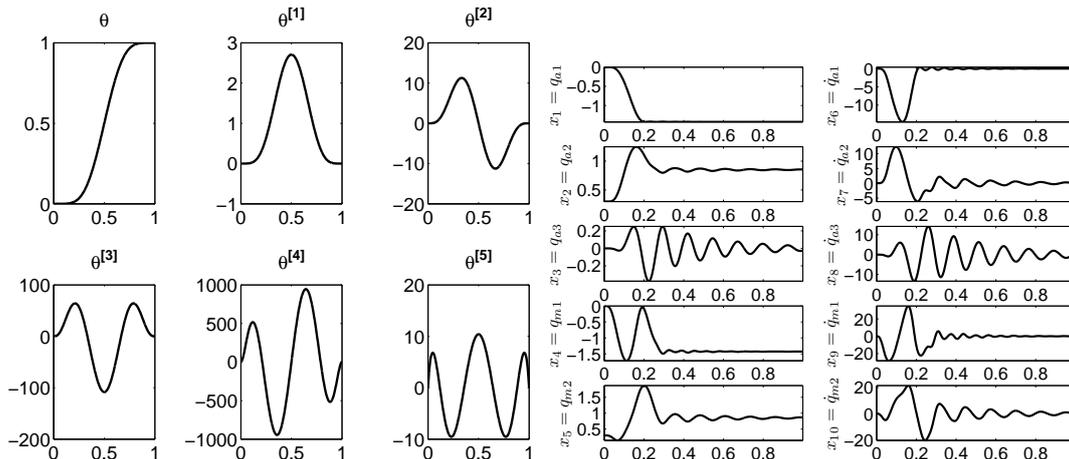


Figure 4: Left: trajectory polynomial  $p(t)$  including first five derivatives. Right: the state solution for 3-step BDF and 3 ms stepsize.

Hypothesis, where the underdetermined system (23) is solved, was also implemented. However, this solver did not give an accurate solution in this implementation. An attempt to solve the index 1 system with the *Matlab* solver *ode15i* also failed due to numerical problems. However, both these solvers were successfully applied to a 2 DOF manipulator as shown in [24].

The reference trajectory is a 2.2 m straight line, performed in 0.3 s, an extremely challenging trajectory for this elastic manipulator with eigenfrequencies down to 2 Hz. The solvers were implemented in *Matlab*, using *fsolve* to solve the system of non-linear equations. No effort to make the solvers computationally efficient was made, but they were still adequate for a comparison of the behavior of the algorithms. The execution times are shown in Table 2. The time for solving the nonlinear dynamics does not increase proportional to the number of operations in the DAE. This indicates that the nonlinear terms in some way improve the solvability of the system. For a given BDF order and stepsize, the solutions for the different systems were almost identical. The solutions for the states and control signals are shown in Fig. (4) and (5). The computed control signal was fed into a simulation model for verification of the inverse dynamics solution. In Fig. (5) the result is shown and it is concluded the errors in the simulated robot tool position are small (less than 0.05 mm) and that there is a small drift due to the lack of a stabilizing feedback controller. For accurate solutions, the stepsize has to be small and an increase of the stepsize will increase the tool vibrations. A 3-step BDF and 3 ms stepsize is adequate in this case. Some snapshots from a simulation are shown in Fig. (6). The continuous system was linearized and analyzed along the solution trajectory. As expected, the system was minimum phase at all points. If the model parameters are changed so that the linearized system is non-minimum phase, the inverse dynamics solution is unbounded as expected. In general, a

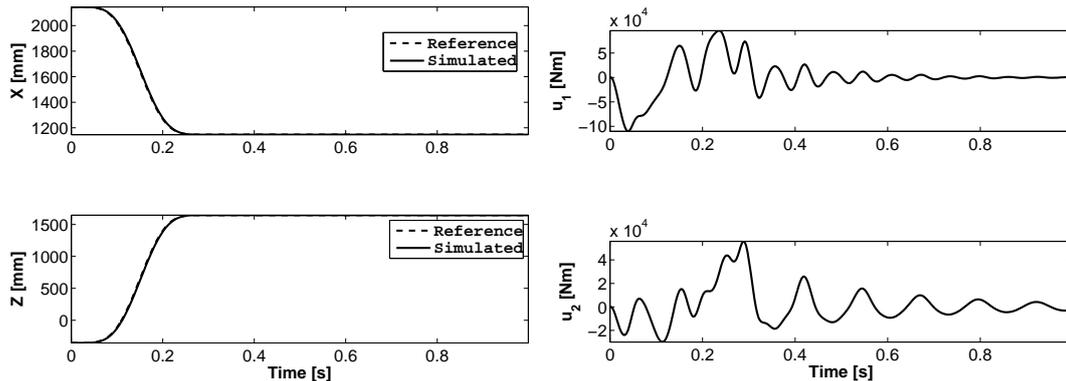


Figure 5: The tool position (left) and control signal (right). 3-step BDF and 3 ms stepsize.

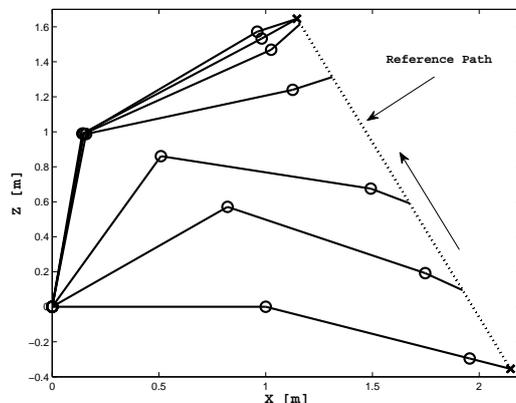


Figure 6: Snapshots from an animation of the movement. The reference path is shown as a dotted line. The direction is indicated by an arrow. Note the deflection of the, outer, non-actuated joint

stable zero dynamics (minimum phase) of the linearized system gives a stable zero dynamics for the nonlinear system, see, e.g., [13]. Note that the discretized system can be non-minimum phase, even though the continuous system is minimum phase, due to zeros introduced by the discretization. Simulations also show that increased damping improves the solvability. This can be understood since the index increases one step if the damping is zero.

## 10 INVERSE DYNAMICS FOR NON-MINIMUM PHASE SYSTEMS

If the system, from input torque to tool position, is non-minimum phase (NMP), the initial-value solvers presented in the previous sections will give an unbounded solution. Several approaches for handling the inverse dynamics of NMP systems have been suggested. For linear systems, a non-causal solution can be obtained if the MP and the NMP dynamics are separated, solved separately (forward and backwards in time, respectively), and then superimposed, see, e.g., [19]. A method for non-causal inverse of nonlinear systems is described in [10], and in [9], a non-causal solution is found for a flexible-link manipulator by applying iterative learning control. Another method is to define an alternative output which gives a stable or non-existing zero dynamics. This approach is used in, e.g., [8]. In [1], the problem is reformulated from tracking to path-following, i.e., the time evolution of the trajectory polynomial  $p(t)$  is allowed to be changed. Another method would be to reduce the model to get a minimum phase system.

	$m$	$\xi$	$l$	$j$	$k$	$d$	$J_m$
Joint 1 and 2	100	0.5	1	5	$5 \cdot 10^5$	500	100
Joint 3	200	0.7	1.4	50	$5 \cdot 10^5$	500	-

Table 3: Parameters of the 5 DOF model (non-minimum phase dynamics).

These methods have been exemplified on, in general, small systems. To be able to solve the problem for a complex multi-axis manipulator, we propose the use of numerical optimization.

### 10.1 A discretized DAE optimization solver

In this approach, the inverse dynamics problem is solved for the complete trajectory in one step. The discretized DAE (k-step BDF) to solve for time step  $i$  is  $F(t_i, x_i, x_{i-1}, \dots, x_{i-k}) = 0$ . Note that  $F(\cdot)$  can be the original high-index or the index-reduced DAE. The DAE to solve for the complete trajectory,  $i = 1, \dots, N$ , is<sup>10</sup>

$$\begin{bmatrix} F(t_1, x_1, x_{1-1}, \dots, x_{1-k}) \\ F(t_2, x_2, x_{2-1}, \dots, x_{2-k}) \\ \dots \\ F(t_N, x_N, x_{N-1}, \dots, x_{N-k}) \end{bmatrix} = G(y) = 0, \quad (25)$$

where

$$y = [x_1 \ x_2 \ \dots \ x_N]. \quad (26)$$

One alternative is to solve the nonlinear least-squares (NLS) problem

$$\min_y \|G(y)\|_2^2,$$

and another approach is to solve the nonlinear constrained optimization (NCO) problem

$$\min_y \|y\|_2^2, \quad \text{subject to } G(y) = 0.$$

The trajectory should start at  $t = \Delta t$  to allow a non-causal solution. The minimization requirement of the states  $y$  is used to get a bounded solution. Other choices, e.g., torque or power are possible. However, in the simulation runs, the obtained solutions were always bounded, even if this requirement was removed. This needs to be further investigated. If needed to obtain a bounded solution, the function  $G(y)$  used in the NLS approach can be extended with, e.g.,  $y$ .

### 10.2 Simulation with 5 DOF model using the discretized DAE optimization solver

The previously described 5 DOF model from Section 6, with parameters as shown in Table 3, was used in the simulation, using the described discretized DAE optimization solver. The model parameters yield a NMP system for the configurations used. To avoid drift in the solution, the system was stabilized with a PID controller according to Fig. (7). The initial values for all time steps in  $y$  are set to the initial values at  $t = 0$ . The example movement is a straight line, 25 mm in both directions (x and z), with an acceleration of  $40 \text{ m/s}^2$ , reaching a speed of 1 m/s. The result is shown in Fig. (8). Note that the torque is applied before the trajectory start (at  $t = 0.1 \text{ s}$ ), in order to create the necessary initial deflection. Also note the control action after

<sup>10</sup>Initial values  $[x_{1-1}, \dots, x_{1-k}]$  are computed as described in Section 9.1

the trajectory end (at  $t = 0.2$  s) in order to release the deflection. A suitable  $\Delta t$  to allow a non-causal solution, seems to be 2–3 times the time constants of the slowest NMP zero. The linearized system has a NMP zero around 27 Hz and the lowest eigenfrequency around 5 Hz. The stepsize is 2 ms which gives 151 time steps in this simulation, and a 6-step BDF is used. If the index-1 system (19) is used, the NCO solver with 1812 variables and 1812 nonlinear equality constraints takes 880 s to solve. The NLS solver takes 630 s and the execution time could be reduced further if the index-2 or index-3 systems were used. The solvers are *Matlab* implementations (using *lsqnonlin* and *fmincon*), numerical gradients and Hessians are used, and additional computational optimization of the algorithms would yield a significant reduction of the computational time, but this has not been the main focus for the present study.

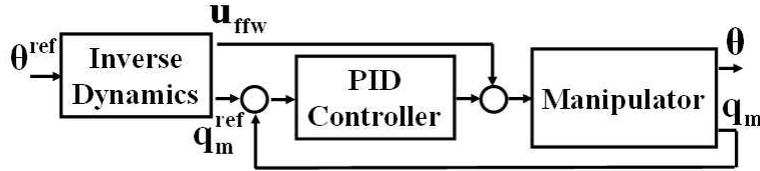


Figure 7: The inverse dynamics computes the desired motor position  $q_m^{ref}$  and the torque feedforward  $u_{ffw}$  from the desired trajectory  $\theta^{ref}$ . The PID controller computes a torque adjustment from the measured motor position  $q_m$  and the desired motor position.

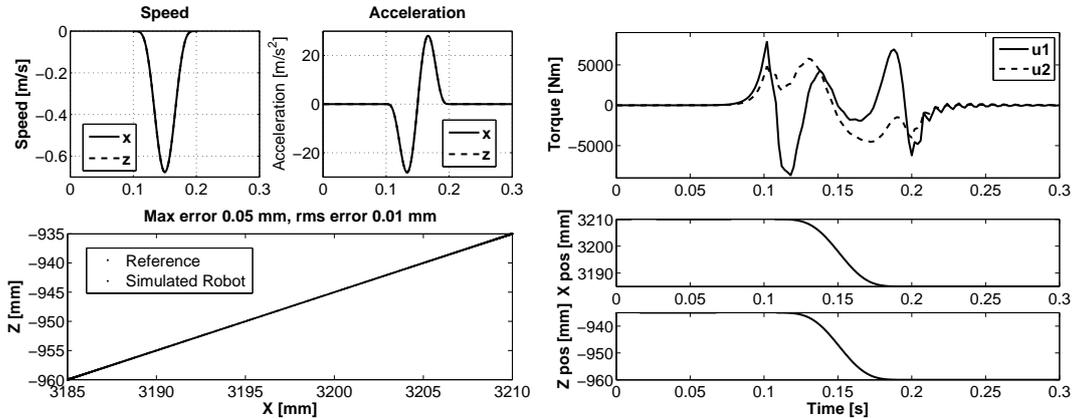


Figure 8: Left: Cartesian speed, acceleration, and the simulated path. Maximum path error is 0.05 mm. Right: The control signals (motor torques) and the tool x- and z-position.

### 10.3 A continuous DAE optimization solver

The algorithms in Section 10.1 are straight forward to apply but result in very large optimization problems. One way of reducing the problem size is to describe the motor and joint angular positions,  $q(t) = [q_a(t) \ q_m(t)]^T$ , using a basis function expansion. In this work, the solution for the complete trajectory is described with one basis function expansion. Depending on the trajectory complexity, it could be advantageous to split the trajectory in several parts and use one expansion for each part. One possible expansion is a combination of a polynomial and

a Fourier series, i.e.,

$$q(t) = \sum_{m=0}^M a_m t^m + \sum_{r=1}^R [b_r \sin(r \frac{2\pi}{Nh} t) + c_r \cos(r \frac{2\pi}{Nh} t)] = q(t, a_1, \dots, a_M, b_1, \dots, b_R, c_1, \dots, c_R). \quad (27)$$

The derivatives  $\dot{q}$  and  $\ddot{q}$  can be computed analytically and the DAE to solve for time step  $i$  is then

$$\Phi(t_i, q_i, \dot{q}_i, \ddot{q}_i) = \Phi(t_i, a_1, \dots, a_M, b_1, \dots, b_R, c_1, \dots, c_R) = 0, \quad (28)$$

where  $\Phi$  is the original *extended flexible joint model* described in (1) and (2), omitting the motor torque balance equation. The DAE to solve for the complete trajectory is then

$$\begin{bmatrix} \Phi(t_1, a_1, \dots, a_M, b_1, \dots, b_R, c_1, \dots, c_R) \\ \Phi(t_2, a_1, \dots, a_M, b_1, \dots, b_R, c_1, \dots, c_R) \\ \dots \\ \Phi(t_N, a_1, \dots, a_M, b_1, \dots, b_R, c_1, \dots, c_R) \end{bmatrix} = \Psi(\mu) = 0, \quad (29)$$

where  $\mu = [a_1 \dots a_M \ b_1 \dots b_R \ c_1 \dots c_R]$ , and each element, e.g.,  $a_1$ , is a vector with the same dimension as  $q$ . The solution can now be obtained by solving the NLS problem

$$\min_{\mu} \|\Psi(\mu)\|_2^2.$$

#### 10.4 Simulation using the continuous DAE optimization solver

The continuous DAE optimization solver was used on the same 5 DOF model and the same trajectory as used in Section 10.2. The series expansion used had  $M = 1$ ,  $R = 20$ , and stepsize  $h = 5$  ms. This resulted in a NLS problem with 210 variables and 305 nonlinear equations which was solved in 22 s. The maximum path error was the same as previously (0.05 mm), although small oscillations could be seen in the motor torque. To reduce these, the value of  $R$  can be increased (number of terms in the Fourier series). The execution time could be reduced further by computing an approximate solution and use as the initial value of  $\mu$ . This could, e.g., be done by computing the series expansion for the rigid solution, i.e., motor and actuated joint angular positions are equal and non-actuated joint positions are zero. Another alternative would be to compute the algebraic (flat) inverse dynamic solution for the flexible joint approximation, omitting the non-actuated joint. The number of necessary terms in the Fourier series can be approximated by Fourier analysis of an approximate solution.

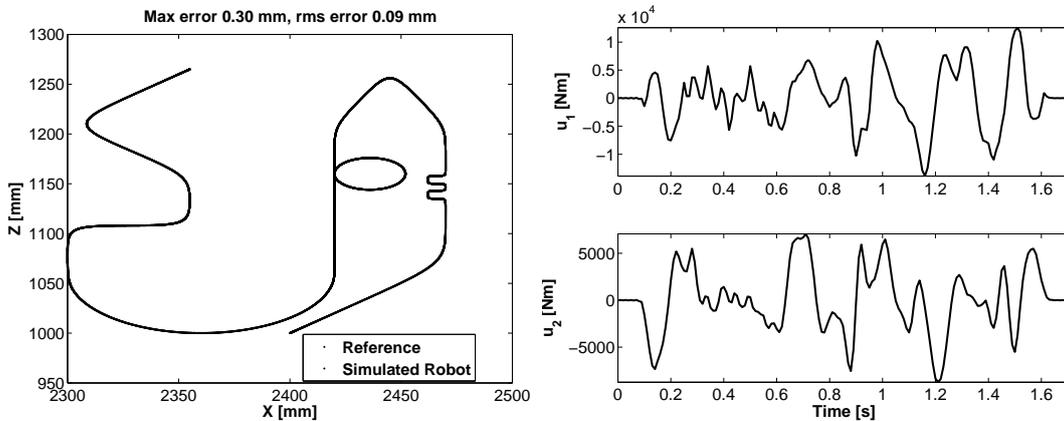


Figure 9: The position (left) and control signal (right) for the complex path.

To show that this approach can be used for more complex trajectories, a complex trajectory containing straight lines, circular segments and corner zones was used. This trajectory is used by robot manufacturers and customers for performance evaluation. The trajectory acceleration was  $50 \text{ m/s}^2$ , giving a speed of up to  $1.5 \text{ m/s}$ . The resulting position and torque are shown in Fig. (9). The parameters were chosen as  $M = 1$ ,  $R = 84$ , and  $h = 10 \text{ ms}$ . The execution time was  $1100 \text{ s}$  and the maximum path error  $0.3 \text{ mm}$ . To solve the inverse dynamics using the continuous DAE optimization method reduces the optimization problem and seems very promising. As a final remark it could be added that trajectories for a manipulator model with 12 DOF have been tested, using the same method. Accurate solutions were found, although the execution time was quite long.

## 11 CONCLUSION, DISCUSSION, AND FUTURE WORK

The inverse dynamics of the extended flexible joint model can be solved with a constant-stepsize constant-order BDF. The difference between solving the original high-index DAE and the index-reduced DAE is small with respect to solvability and execution time. Solution of the original high-index system is attractive since differentiations of the constraint equations are avoided. Scaling of the algebraic equations and their derivatives is important and this should be further investigated, including scaling of variables. The use of more efficient equation solvers and optimizers, utilizing analytic gradients and Hessians, is another area for future research. Evaluation of the algorithms, using a more complex manipulator model, e.g., a six-axes model with transmission nonlinearities, is another important future work. Another logical step is to evaluate the algorithms on a real robot manipulator.

Generally, a practical manipulator controller must be computed on-line as the future trajectory can be, at least partly, unknown, e.g., if sensor input affects the trajectory. This makes the optimization approach unfeasible and will set hard real-time requirements on the initial-value solver. However, for some applications, an off-line feedforward computation could be acceptable. Of course, there are a number of problems to be solved in order to make this approach feasible, e.g., smooth trajectory generation with constrained control signals and handling of manipulator singularities and alternative configurations. The main motivation for this work is, however, that a method for perfect feedforward control is valuable for evaluating the limit of control performance, and for developing approximate control algorithms for on-line computation.

## 12 ACKNOWLEDGEMENTS

This work was supported by ABB AB-Robotics, Vinnova's Industry Excellence Center LINK-SIC at Linköping University, and the Swedish Research Council (VR), which is gratefully acknowledged.

## References

- [1] A.P. Aguiar, J.P. Hespanha, and P.V. Kokotovic. Path-following for nonminimum phase systems removes performance limitations. *IEEE Transactions on Automatic Control*, 50(2):234–239, 2005.
- [2] M. Björkman, T. Brogårdh, S. Hanssen, S.-E. Lindström, S. Moberg, and M. Norrlöf. A new concept for motion control of industrial robots. In *Proceedings of 17th IFAC World Congress, 2008*, Seoul, Korea, July 2008.

- [3] W. Blajer. Dynamics and control of mechanical systems in partly specified motion. *Journal of the Franklin Institute*, 334B(3):407–426, May 1997.
- [4] W. Blajer and K. Kolodziejczyk. A geometric approach to solving problems of control constraints: Theory and a dae framework. *Multibody System Dynamics*, 11(4):341–350, May 2004.
- [5] K. E. Brenan, S. L. Campbell, and L.R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1996. ISBN 0-89871-353-6.
- [6] T. Brogårdh. Present and future robot control development—an industrial perspective. *Annual Reviews in Control*, 31(1):69–79, 2007.
- [7] A. De Luca. Feedforward/feedback laws for the control of flexible robots. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, pages 233–240, San Francisco, CA, April 2000.
- [8] A. De Luca and G. Di Giovanni. Rest-to-rest motion of a one-link flexible arm. In *2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics Proceedings*, pages 923–928, Como, Italy, 2001.
- [9] A. De Luca, S. Panzieri, and G. Ulivi. Stable inversion control for flexible link manipulators. In *Proc. 1998 IEEE International Conference on Robotics and Automation*, pages 799–804, Leuven, Belgium, May 1998.
- [10] A. Devasia, D. Chen, and B. Paden. Nonlinear inversion-based output tracking. *IEEE Transactions on Automatic Control*, 41(7):930–942, 1996.
- [11] I. S. Duff, A. M. Erisman, and J. K. Reid. *Direct Methods for Sparse Matrices*. Clarendon Press, Oxford, 1986.
- [12] P. Fritzon. *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. John Wiley & Sons, New York, 2004.
- [13] A. Isidori. *Nonlinear Control Systems*. Springer-Verlag, London, Great Britain, 1995.
- [14] K.P. Jankowski and H. Van Brussel. An approach to discrete inverse dynamics control of flexible-joint robots. *IEEE Transactions on Robotics and Automation*, 8(5):651–658, October 1992.
- [15] T. R. Kane and D. A. Levinson. *Dynamics: Theory and Applications*. McGraw-Hill Publishing Company, 1985.
- [16] P. Kunkel and V. Mehrmann. Index reduction for differential-algebraic equations by minimal extension. *Z. Angew. Math. Mech.*, 84:579–597, 2004.
- [17] P. Kunkel and V. Mehrmann. *Differential-Algebraic Equations: Analysis and Numerical Solution*. European Mathematical Society, 2006.
- [18] P. Kunkel and V. Mehrmann. Regular solutions of nonlinear differential-algebraic equations and their numerical determination. *Numerische Mathematik*, 79:581–600, 1998.

- [19] D.-S. Kwon and W.J. Book. An inverse dynamic method yielding flexible manipulator state trajectories. In *Proceedings of the 1990 American Control Conference, vol 1*, pages 186–193, San Diego, CA, USA, 1990.
- [20] P. Lötstedt and L. R. Petzold. Numerical solution of nonlinear differential equations with algebraic constraints i: Convergence results for backwards differentiation formulas. *Mathematics of Computation*, 46(174):491–516, April 1986.
- [21] S.-E. Mattson and G. Söderlind. Index reduction in differential-algebraic equations using dummy derivatives. *SIAM Journal of Scientific Computing*, 14(3):677–692, May 1993.
- [22] S. Moberg and S. Hanssen. A DAE approach to feedforward control of flexible manipulators. In *Proc. 2007 IEEE International Conference on Robotics and Automation*, pages 3439–3444, Roma, Italy, April 2007.
- [23] S. Moberg and S. Hanssen. On feedback linearization for robust tracking control of flexible joint robots. In *Proc. 17th IFAC World Congress*, Seoul, Korea, July 2008.
- [24] S. Moberg and S. Hanssen. Inverse dynamics of flexible manipulators - a DAE approach. Technical Report LiTH-ISY-R-2866, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, October 2008.
- [25] S. Moberg, J. Öhr, and S. Gunnarsson. A benchmark problem for robust control of a multivariable nonlinear flexible manipulator. In *Proc. 17th IFAC World Congress*, Seoul, Korea, July 2008.
- [26] C. C. Pantelides. The consistent initialization of differential-algebraic systems. *SIAM Journal of Scientific and Statistical Computing*, 9(2):213–231, 1988.
- [27] L. R. Petzold and P. Lötstedt. Numerical solution of nonlinear differential equations with algebraic constraints ii: Practical implications. *SIAM Journal of Scientific and Statistical Computing*, 7(3):720–733, July 1986.
- [28] P. Rouchon, M. Fliess, J. Lévine, and P. Martin. Flatness, motion planning and trailer systems. In *Proceedings of the 32nd Conference on Decision and Control*, pages 2700–2705, San Antonio, Texas, December 1993.
- [29] A.A. Shabana. *Dynamics of Multibody Systems*. Cambridge University Press, Cambridge, United Kingdom, 1998.
- [30] M. W. Spong. Modeling and control of elastic joint robots. *Journal of Dynamic Systems, Measurement, and Control*, 109:310–319, December 1987.
- [31] M. Thümmel, M. Martin Otter, and J. Bals. Control of robots with elastic joints based on automatic generation of inverse dynamic models. In *Proceedings of the 2001 IEEE/RSJ International Conference in Intelligent Robots and Systems*, pages 925–930, Maui, Hawaii, USA, October 2001.
- [32] E. Wernholt and S. Moberg. Frequency-domain gray-box identification of industrial robots. In *Proceedings of 17th IFAC World Congress*, pages 15372–15380, Seoul, Korea, July 2008.

	<b>Avdelning, Institution</b> Division, Department  Division of Automatic Control Department of Electrical Engineering	<b>Datum</b> Date  2010-03-08
	<b>Språk</b> Language  <input type="checkbox"/> Svenska/Swedish <input checked="" type="checkbox"/> Engelska/English  <input type="checkbox"/> _____	<b>Rapporttyp</b> Report category  <input type="checkbox"/> Licentiatavhandling <input type="checkbox"/> Examensarbete <input type="checkbox"/> C-uppsats <input type="checkbox"/> D-uppsats <input checked="" type="checkbox"/> Övrig rapport <input type="checkbox"/> _____
<b>URL för elektronisk version</b>  <a href="http://www.control.isy.liu.se">http://www.control.isy.liu.se</a>		<b>ISSN</b> 1400-3902  LiTH-ISY-R-2939
<b>Titel</b> Title	Inverse Dynamics of Flexible Manipulators	
<b>Författare</b> Author	Stig Moberg, Sven Hanssen	
<b>Sammanfattning</b> Abstract		
<p>High performance robot manipulators, in terms of cycle time and accuracy, require well designed control methods, based on accurate dynamic models. Robot manipulators are traditionally described by the flexible joint model or the flexible link model. These models only consider elasticity in the rotational direction. When these models are used for control or simulation, the accuracy can be limited due to the model simplifications, since a real manipulator has a distributed flexibility in all directions. This work investigates different methods for the inverse dynamics of a more general manipulator model, called the extended flexible joint model. The inverse dynamics solution is needed for feedforward control, which is often used for high-precision robot manipulator control.</p> <p>The inverse dynamics of the extended flexible joint model can be computed as the solution of a high-index differential algebraic equation (DAE). One method is to solve the discretized DAE using a constant-stepsize constant-order backwards differentiation formula (BDF). This work shows that there is only a small difference between solving the original high-index DAE and the index-reduced DAE. It is also concluded that scaling of the algebraic equations and their derivatives is important.</p> <p>The inverse dynamics can be solved as an initial-value problem if the zero dynamics of the system is stable, i.e., minimum phase. For unstable zero dynamics, an optimization approach based on the discretized DAE is suggested. An optimization method, using a continuous DAE formulation, is also suggested and evaluated. The solvers are illustrated by simulation, using a manipulator with two actuators and five degrees-of-freedom.</p>		
<b>Nyckelord</b> Keywords	Manipulator, Control, Differential Algebraic Equation, Flexible Multibody Dynamics, Non-minimum Phase, Inverse Dynamics.	