

# A DAE approach to Feedforward Control of Flexible Manipulators

Stig Moberg and Sven Hanssen

**Abstract**—This work investigates feedforward control of elastic robot structures. A general serial link elastic robot model which can describe a modern industrial robot in a realistic way is presented. The feedforward control problem is discussed and a solution method for the inverse dynamics problem is proposed. This method involves solving a differential algebraic equation (DAE). A simulation example for an elastic two axis planar robot is also included and shows promising results.

## I. INTRODUCTION

High accuracy control of industrial robot manipulators is a challenging task which has been studied by academic and industrial researchers since the 1970's. Control methods for rigid direct drive robots are e.g. described in [6]. The two main approaches are feedforward control and computed torque control (i.e. feedback linearization and decoupling) respectively. Both are based on a rigid dynamic model and combined with a diagonal PD or PID controller. The methods show similar results as described e.g. in [18].

Control methods for flexible joint robots (i.e. elastic gear transmissions and rigid links) can e.g. be found in [13] and [12]. Experiments on industrial robots are described in [5] and [1]. The main approaches are the same as for direct drive robots.

The trend in industrial robots is towards lightweight robot structures with a higher degree of elasticity but with preserved payload capabilities. This results in lower mechanical resonance frequencies inside the control bandwidth. The sources of elasticity in such a manipulator are e.g. gearboxes, bearings, elastic foundations, elastic payloads as well as bending and torsion of the links. In [7], it is shown that there are cases when the total elasticity in a plane perpendicular to the preceding joint and the total elasticity out of this plane (bending and torsion) are of the same order.

In most publications concerning industrial robots, only gear elasticity in the rotational direction and, in some cases, link deformation restricted to a plane perpendicular to the preceding joint are included in the model. This restricted model simplifies the control design but limits the attainable performance.

This work presents a general serial link elastic model that includes joint elasticity in all directions and thus describes a modern industrial robot in a reasonable way. Furthermore, a feedforward approach based on the solution of a differential

algebraic equation (DAE) is proposed. The DAE formulation of the robot feedforward problem has been described previously by others but to the author's knowledge never been implemented in a simulation environment for such a complex and realistic model structure as mentioned above. Furthermore, a way of reducing the DAE complexity is also proposed. Finally, the model structure and the feedforward method are illustrated in a simulation example with a two axis robot model.

## II. AN INDUSTRIAL MANIPULATOR

The most common type of industrial manipulator has six serially mounted links, all controlled by electrical motors via gears. An example of a serial industrial manipulator is shown in Fig. 1. The dynamics of the manipulator change rapidly as the robot links move fast within its working range, and the dynamic couplings between the links are strong. Moreover, the robot system is elastic as described in Section I and the gears have nonlinearities such as backlash, friction and nonlinear elasticity. From a control engineering perspective a manipulator can be described as a nonlinear multivariable dynamical system having the six motor currents as the inputs and the six measurable motor angles as outputs. The goal of the motion control is to control the *orientation* and the *position* of the *tool* along a certain desired path.



Fig. 1. IRB6600 from ABB equipped with a spotwelding gun.

## III. ROBOT MODEL

In this section, a general serial link robot model capable of adequately describing the different sources of flexibility, as described in Section I, is proposed. This model structure will later be used for deriving a feedforward control law. The identification of an industrial robot using a model derived from this general model class is described in [7].

This work was supported by ABB Robotics and the Swedish Research Council (VR)

S. Moberg is with Division of Automatic Control, Department of Electrical Engineering, Linköpings Universitet, SE-581 83 Linköping, Sweden stig@isy.liu.se

S. Hanssen is with the Department of Solid Mechanics, Royal Institute of Technology, SE-100 44 Stockholm, Sweden soha@kth.se

### A. General Description

The model consists of a serial kinematic chain of  $R$  rigid bodies. One rigid body  $rb^i$  is illustrated in Fig. 2 and is described by its mass  $m^i$ , center of mass  $\xi^i$ , inertia tensor w.r.t. center of mass  $J^i$  and length  $l^i$ . Due to the symmetrical inertia tensor, only six components of  $J^i$  need to be defined. All parameters are described in a coordinate system  $a^i$ , fixed in  $rb^i$ , and are defined as follows:

$$\xi^i = [\xi_x^i \quad \xi_y^i \quad \xi_z^i] \quad (1a)$$

$$J^i = \begin{bmatrix} J_{xx}^i & J_{xy}^i & J_{xz}^i \\ J_{xy}^i & J_{yy}^i & J_{yz}^i \\ J_{xz}^i & J_{yz}^i & J_{zz}^i \end{bmatrix} \quad (1b)$$

$$l^i = [l_x^i \quad l_y^i \quad l_z^i] \quad (1c)$$

The rigid body  $rb^i$  is connected to  $rb^{i-1}$  by three torsional

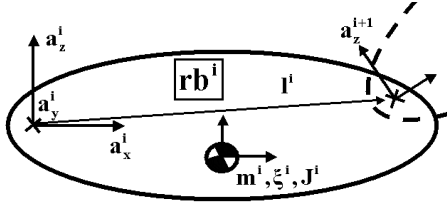


Fig. 2. Definition of rigid body

spring-damper pairs and adds 3 DOF to the model as its configuration can be described by three angular positions and the given position and orientation of  $a^{i-1}$ . The position of  $a^{i-1}$  is determined of the angular positions of all  $rb^k$ ,  $k < i$ . Thus, the arm system has  $3R$  DOF (i.e. the number of independent coordinates necessary to specify its configuration).  $M$  of the DOFs (maximum one per rigid body) are actuated and corresponding to a connection of two rigid bodies by a motor and a gearbox. Therefore, the total system has  $3R + M$  DOF. Note that one link (always actuated) can consist of one or more rigid bodies.

The angular position and inertia for motor  $i$  are denoted  $q_m^i$  and  $J_m^i$  respectively. The generalized coordinates  $q_d^i$  (arm angular positions) defines rotations of the coordinate systems  $a^i$ . Index  $d$  denotes the direction ( $x$ ,  $y$  or  $z$ ). The coordinate systems  $a^i$  are defined to have the same orientation in zero position, i.e.  $q_d^i = q_m^i = 0$ . The generalized speeds are defined as  $v_d^i = \dot{q}_d^i$  and  $v_m^i = \dot{q}_m^i$ . The motors are placed on the preceding body and the inertial couplings between the motors and the rigid bodies are neglected under the assumption of high gear ratio, see e.g. [19], but the motor mass and inertia are added to the corresponding rigid body.

The springs and dampers are generally nonlinear functions expressing the gearbox torque  $\tau_d^i = \tau_d^i(q_d^i, q_m^i, v_d^i, v_m^i)$  for an actuated DOF or the constraint torque  $\tau_d^i = \tau_d^i(q_d^i, v_d^i)$  for an unactuated DOF. For a linear spring,  $\tau_d^i = k_d^i(q_d^i - q_m^i) + d_d^i(v_d^i - v_m^i)$  or  $\tau_d^i = k_d^i q_d^i + d_d^i v_d^i$ .

The torque control of the motor is assumed to be ideal so the  $M$  input signals of the system, which are the motor

torque references  $u$  equal the motor torques  $\tau$ . The system has  $M$  controlled output variables, typically the position and orientation of the robot tool.

An extension of the model is to define the number of arm DOF as  $N$ , which can be maximum  $6R$ , by adding three linear springs for translational deformation. Furthermore, if some of the springs are defined as rigid,  $N$  can be lowered. As the purpose of this work is to study and control effects caused by elasticity, friction is omitted.

The equations of motion are derived by computing the linear and angular momentum. By using Kane's method [9] the projected equations of motion are derived to yield a system of ordinary differential equations (ODE) with minimum number of DOFs, see also [11].

The model equations can be described as a system of first order ODE's

$$M_a(q_a)\dot{v}_a = c(q_a, v_a) + g(q_a) + \tau_a(q_a, q_m, v_a, v_m) \quad (2a)$$

$$M_m\dot{v}_m = \tau_m(q_a, q_m, v_a, v_m) + u \quad (2b)$$

$$\dot{q}_a = v_a \quad (2c)$$

$$\dot{q}_m = v_m \quad (2d)$$

where  $q_a \in R^N$  are the arm angular positions ( $q_d^i$ ) and  $q_m \in R^M$  are the motor angular positions. The corresponding speeds are  $v_a$  and  $v_m$ . The vector of spring torques acting on the arm system is described by  $\tau_a \in R^N$  and of the spring torques on the motors by  $\tau_m \in R^M$ .  $M_a(q_a) \in R^{N \times N}$  is the inertia matrix for the arms and  $M_m \in R^{M \times M}$  is the diagonal inertia matrix of the motors. The Coriolis and centrifugal torques are described by  $c(q_a, \dot{q}_a) \in R^N$  and  $g(q_a) \in R^N$  is the gravity torque. The time  $t$  is omitted in the expressions.

For a complete model including the position and orientation of the tool,  $Z$ , the forward kinematic model of the robot must be added. The kinematic model is a mapping of  $q_a \in R^N$  to  $Z \in R^M$ . The complete model of the robot is then described by (2) and

$$Z = \Gamma(q_a). \quad (3)$$

### B. A Robot Model with 5 DOF: Description and Analysis

The general robot model described in Section III-A is here used to derive a specific model. All links are aligned along the  $x$  axis at zero position. There are in total 21 parameters associated to one rigid body, so the model has in total 63 possible parameters. The model has  $R = 3$ ,  $N = 3$  and  $M = 2$ , i.e. 3 rigid bodies, 3 arm DOF and 2 actuated DOF. This reduces the number of parameters in the model equations. All parameters are defined in SI units with the motor inertia transformed by the square of the gear ratio. Parameter values are listed in Table I. The model is illustrated in Fig. 3 and is a planar model with linear elasticity, constrained to work in the  $xz$ -plane with the gravitational constant  $g$  set to zero. Note that this model has its elasticity in a plane perpendicular to the preceding joint and do not demonstrate the type of elasticity described in Section I. However, it is a simple model for a first verification of the new feedforward control algorithm. All model equations can be found in [15].

One important restriction for the type of control considered in this work, i.e. perfect causal feedforward control, is that the system must be minimum phase. A linear analysis of the model for  $q_y^1 = q_m^1 = 0$ ,  $q_y^2 = q_m^2 = 0.3$  and  $q_y^3 = 0$  results in a controllable minimum phase system. A nonlinear analysis of controllability and minimum phase behavior is outside the scope of this article. The transfer functions of the linearized system are shown in Fig. 4.

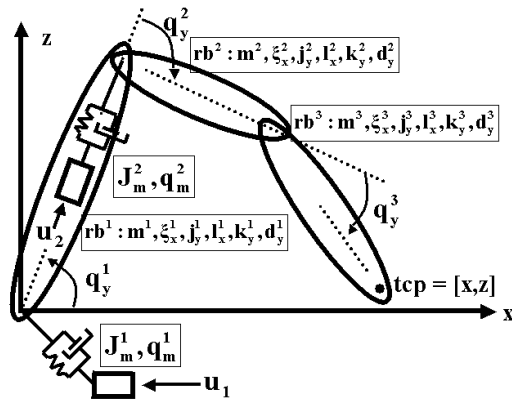


Fig. 3. The 5 DOF model ( $k_{3y}$  and  $c_{3y}$  not shown)

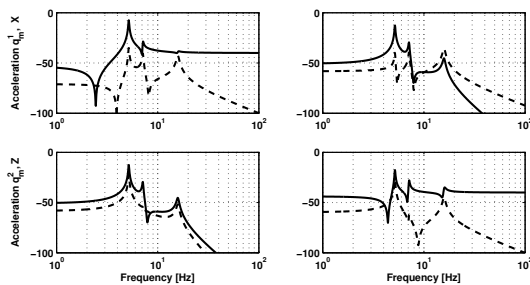


Fig. 4. Transfer Function Magnitude from  $u$  to motor acceleration (solid) and cartesian acceleration (dashed).

TABLE I  
PARAMETERS OF THE MODEL

	$rb^1$	$rb^2$	$rb^3$
$m^i$	100	100	200
$\xi_{sp}^i$	0.5	0.5	0.1
$l_{xp}^i$	1	1	0.2
$J_{yy}^i$	5	5	50
$k_y^i$	1E5	1E5	1E5
$d_y^i$	50	50	50
$J_m^i$	100	100	NA

#### IV. FEEDFORWARD CONTROL OF A FLEXIBLE MANIPULATOR

The controller structure considered is illustrated in Fig. 5.  $Z_d$  is the desired tool trajectory described in Cartesian coordinates and  $Z$  is the actual trajectory. The Reference and Feedforward Generation Block (FFW) computes the

feedforward torque  $u_{ffw}$  and the state references  $\bar{x}_d$  that are used by the feedback controller (FDB). The robot has uncertain parameters illustrated as a feedback with unknown parameters  $\Delta$  and is exposed to disturbances  $d$  and measurement noise  $e$ . The measured signals are denoted  $y_m$ . Note that the dimension of  $\bar{x}_d$  and  $y_m$  may differ if some states are reconstructed by FDB. The purpose of the FFW is to generate model based references for perfect tracking (if possible). The purpose of the FDB is, under the influence of measurement noise, to stabilize the system, reject disturbances and to compensate for errors in the FFW.

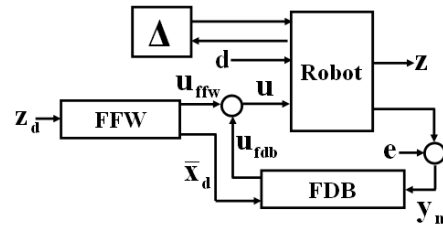


Fig. 5. Robot Controller Structure

In this section, we treat the design of the FFW block. The proposed feedback controller is a diagonal controller of PID type which often proves to be a suitable choice for realistic industrial systems, see e.g. [16]. Only the motor position is measured by the PID-controller (i.e.  $y_m = q_{m,measured}$ ,  $\bar{x}_d = q_{m,desired}$ ).

The forward dynamics problem, i.e. solving (2) for the state variables with the motor torque as input involves solving an ODE. The tool position can then be computed from the states according to (3). The inverse dynamics problem, i.e. solving for the states and the motor torque with the desired tool position as input to the system is generally much harder.

The inverse dynamics solution for flexible joint robots without damping and with linear elasticity is described e.g. in [12]. Flexible joint robots (gearbox elasticity only) have  $R = M = N$  in the model structure described in Section III. This is an example of a so-called differentially flat system (defined e.g. in [17]) 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. In this case the desired trajectory must be four times differentiable.

If damping is introduced an ODE must be solved and if the elasticity is allowed to be nonlinear the ODE becomes nonlinear. This is e.g. observed in [20] where a DAE formulation of the problem is suggested.

Generally, the inverse dynamics problem can be formulated as a differential algebraic equation (DAE). This is formulated and illustrated by some linear spring-mass systems in [2] and [3]. In [3], a solution based on flatness and a solution based on DAE yield the same result. However, it is also concluded that solutions based on flatness are not realistic for more complicated systems. The same conclusion is also presented in [8] for the case of feedback linearization for a flexible joint robot.

Another situation occurs when the number of arm DOF is greater than the number of controlled outputs ( $N > M$ ) which is the case for the model structure described in III-A. The kinematic relation (3) is then non-invertible, and solving the DAE, (2) and (3), is the natural solution. One approximation would be to invert the linearized system and to schedule the model used. However, for nonlinear inverse dynamics of the system, the DAE approach should be considered.

## V. DAE BACKGROUND

The presentation in this section is primarily based on [4] and [10]. Solving the inverse dynamics problem for states and control signals, given the desired output, generally involves solving the DAE described by (2) and (3). As the name implies, a DAE consists of differential and algebraic equations. A DAE can generally be expressed by the fully-implicit description

$$F(\dot{x}, x, u, t) = 0 \quad (4)$$

where  $x \in R^n$  is the state vector,  $u \in R^p$  is the control input and  $F : R^{2n+p+1} \rightarrow R^m$ . If  $F_{\dot{x}} = \partial F / \partial \dot{x}$  is nonsingular, (4) represents an implicit ODE. 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 index can somewhat simplified be defined as the *minimum number of times that all or part of (4) must be differentiated to determine  $\dot{x}$  as a function of  $t, x, u$  and higher derivatives of  $u$* . A semi-explicit DAE is a special case of (4) described as

$$\dot{x} = F(x, y) \quad (5a)$$

$$0 = G(x, y) \quad (5b)$$

where  $u$  and  $t$  are omitted. Differentiation of (5b) w.r.t. time yields

$$0 = G_y(x, y)\dot{y} + G_x(x, y)\dot{x} \quad (6)$$

and if  $G_y$  is nonsingular it is possible to solve for  $\dot{y}$  and the index is equal to 1. Further on, by the implicit function theorem, it is also possible (at least numerically) to solve for  $y = \phi(x)$ . This suggests a straightforward method for solving an Index-1 DAE, which in its simplest Forward Euler form yields ( $h$  is the step length):

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

For arbitrary initial conditions, the DAE solution exhibits an impulse behavior while an ODE solution is well-defined for any initial conditions. The reason is that the solutions are restricted to a space with dimension less than  $n$  by the algebraic equation (5b) and its derivatives up to order  $\nu - 1$ . These differentiated constraints are often denoted implicit constraints. Another difference from an ODE is that the DAE solution may depend on the derivatives of the input  $u$ .

The main method for solving DAEs is to reduce the index by some sort of repeated differentiation until Index-1 or Index-0 form is reached. Many index reduction techniques exist. In the numerical solution following the index reduction, a problem denoted as *drift off* can occur. This means that the solution diverges from the algebraic constraint as it is replaced by a differentiated constraint. In these cases a method for keeping the solution in the allowed solution space must be used. Several methods exists, see e.g. [14] and references therein.

One common software for solving Index-1 or Index-0 DAEs is called DASSL. The basic principle in DASSL is to replace the derivatives in (4) with a backwards differentiation formula (BDF) of order  $k$ . Some higher index DAEs can be solved directly by using the simple 1-step BDF (Euler Backwards). The *drift off* problem then disappears if the original algebraic constraint is kept.

## VI. INVERSE DYNAMICS SOLUTION BY INDEX REDUCTION

The previously described model, (2) and (3), can in principle be written in semi-explicit form since the mass matrices  $M_a$  and  $M_m$  are nonsingular. Repeated differentiation of the algebraic constraint (3) and substitution of differentiated states gives the following system where  $x^{[i]}$  denotes  $d^i x / dt^i$ :

$$\dot{v}_a = \eta_a(q_a, q_m, v_a, v_m) \quad (7a)$$

$$\dot{v}_m = \eta_m(q_a, q_m, v_a, v_m) + M_m^{-1}u \quad (7b)$$

$$\dot{q}_a = v_a \quad (7c)$$

$$\dot{q}_m = v_m \quad (7d)$$

$$0 = \Gamma(q_a) - Z \quad (7e)$$

$$0 = \dot{\Gamma}(q_a, v_a) - \dot{Z} \quad (7f)$$

$$0 = \ddot{\Gamma}(q_a, v_a, q_m, v_m) - \ddot{Z} \quad (7g)$$

$$0 = \Gamma^{[3]}(q_a, v_a, q_m, v_m, u) - Z^{[3]} \quad (7h)$$

$$0 = \Gamma^{[4]}(q_a, v_a, q_m, v_m, u, \dot{u}) - Z^{[4]} \quad (7i)$$

Note that the position and speed dependent terms from (2) multiplied by the inverse mass matrices are denoted  $\eta_a$  and  $\eta_m$ . The control signal  $u$  is here regarded as a state and the full state vector is  $x = [q_a^T, q_m^T, v_a^T, v_m^T, u^T]^T$ . If  $\dot{u}$  can be solved from (7i) we have an Index-4 DAE and the system consisting of (7a) - (7d) and (7h) (which is Index-1) can be solved with e.g. the DASSL software. One problem remains: the *drift off* problem described in Section V must be handled in some way.

## VII. INVERSE DYNAMICS SOLUTION BY 1-STEP BDF

The proposed solution method is the constant step size 1-step BDF applied on the original system where the DAE-index is reduced from 4 to 3 by discarding the motor torque equation (2b) from the system. When the remaining DAE is solved, the control signal  $u$  can be computed from the states. Note that a friction compensation, assuming friction in the gearbox and motor, can be added to the motor torque without

increasing the complexity of the solution. The DAE system to solve is then reduced to

$$M_a(q_a)\dot{v}_a = c(q_a, v_a) + g(q_a) + \tau_a(q_a, q_m, v_a, v_m) \quad (8a)$$

$$\dot{q}_a = v_a \quad (8b)$$

$$\dot{q}_m = v_m \quad (8c)$$

$$Z = \Gamma(q_a) \quad (8d)$$

with  $q_a \in R^N$ ,  $q_m \in R^M$ ,  $v_a \in R^N$  and  $v_m \in R^M$ , which gives  $2(N + M)$  states and  $2(N + M)$  equations, and thus we have a determined system. Note that state is somewhat misused here as all variables in a DAE do not hold a system memory. The *states* of the system are  $x = [q_a^T, q_m^T, v_a^T, v_m^T]^T$  and with reference  $Z_d = Z_d(t)$  implicit in the equation the system can then be described as

$$F(\dot{x}, x, t) = 0 \quad (9)$$

Consistent initial conditions w.r.t. all explicit and implicit constraints must be given to avoid initial transients. Moreover, the trajectory reference must be sufficiently smooth as it must be (implicitly) differentiated four times for the (hidden) Index-4 system. This is accomplished by using a reference  $Z_d$  with  $Z_d^{[4]}$ , i.e. jerk derivative, well defined. The final algorithm can be described as follows:

- 1) Consistent  $x(0)$  must be given
- 2) Solve  $F(\frac{x(t+h)-x(t)}{h}, x_{t+h}, t+h) = 0$
- 3) Compute control signal  $u_{ffw}(t+h)$
- 4) Repeat from 2 with  $t = t+h$

A nonlinear equation solver from Matlab (fsolve) is used in step 2. Experiments show that the solvability of the DAE using this method depends on the size of the system (method tested with some positive result for 12 DOF) as well as on the step size selection (short step size is hard) and that the described index reduction increases the solvability.

Some commercially available software packages (Dymola, Maple) 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 for the cases where a solution was found.

Thus, the suggested method works better than the commercial software packages tested and is adequate for a preliminary evaluation of the feedforward control method.

## VIII. PERFORMANCE REQUIREMENT SPECIFICATION

The problem specification illustrates a typical requirement for a dispensing application (e.g. gluing inside a car body) and is stated as follows:

- The programmed path should be followed by an accuracy of 2 mm (maximum deviation) at an acceleration of  $15 \text{ m/s}^2$  and a speed of 0.5 m/s
- The specification above must be fulfilled for model errors in the tool load by  $\pm 10 \text{ kg}$  (i.e.  $\pm 5\%$  of mass 3)
- The test path is a circular path with radius 25 mm

## IX. SIMULATION EXAMPLE

The robot model described in Section III-B is simulated with the controller structure from Section IV. The FFW block is implemented using the DAE solver from Section VII and the FDB block is a diagonal PID controller. The implementation is discrete time with the step size of DAE solver equal to the sample time of the feedback controller.

No measurement noise or disturbances are used in the simulation as the purpose is to verify the feedforward algorithm. The Cartesian Trajectory Reference  $Z_d$  is a circle computed in polar coordinates [radius r, angle Q] by integration of a desired jerk derivative  $Q^{[4]}(t)$  shown in Fig. 6. The initial position of the robot is according to Section III-B.

The proposed algorithm (DAEFFW) is compared to the standard *flexible joint* feedforward (FJFFW) approach described previously. The non-actuated DOF is then regarded as rigid by the controller. The remaining stiffness parameters  $k_y^1$  and  $k_y^2$  are computed to include the  $k_y^3$  in the best possible way.

The result of the simulations (control signals DAEFFW and path error for both methods) with nominal parameter values and sample time 0.5 ms can be seen in Fig. 7 and 8. The circular path is shown in Fig. 9. As an illustration of the need for high bandwidth modelbased feedforward at high speed path tracking, the result for the PID controller with no feedforward is included in the last figure.

The maximum error for DAEFFW is 0.32 mm (nominal parameters) and 1.39 mm (tool load  $\pm 10 \text{ kg}$ ). The corresponding values for FJFFW is 1.74 mm and 2.82 mm. The result confirms that feedforward is naturally sensitive to model errors but the result also shows that the specification could be fulfilled and that a more advanced feedforward still can yield better result than a less complex one. There are many ways to handle the robustness problem of robot feedforward control. Some suggestions:

- Reducing uncertainty by user identification of uncertain parameters (e.g. tool load identification)
- Improved feedback control with arm side sensors added
- Smoothing the trajectory on the expense of cycle time performance

Increasing the sample time to 1 ms increases the nominal error for DAEFFW to 0.64 mm. This sample time sensitivity could also be expected since we are using an Euler Backwards approximation of the derivatives and this could certainly be improved.

## X. CONCLUSIONS AND FUTURE WORK

The proposed feedforward method shows promising results. The method is sensitive to model errors as can be expected for a feedforward method. The sampling time selection is critical for good performance as well as for solvability of the DAE. The limitation of this method is that the system must be minimum phase.

Future work will include testing the method on a more complex robot model, e.g. by increasing the number of DOF and by introducing non linear elasticities, as well as testing

the method on real robot structures. Alternative DAE solvers and robustness issues should also be addressed in the future. Since the the model structure presented can become non-minimum phase, methods for dealing with this is of greatest

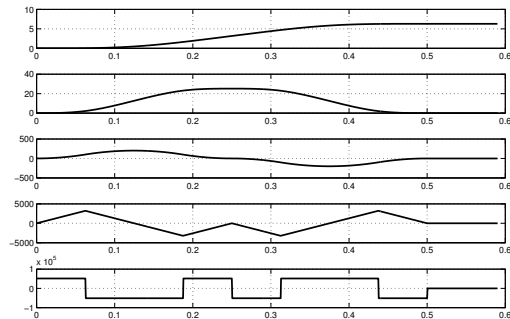


Fig. 6. The Circular Angle  $Q$  and its derivatives

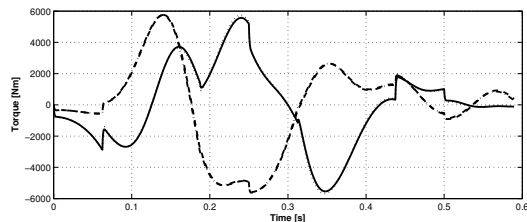


Fig. 7. DAE FFW: Torque feedforward (1-solid, 2-dashed) and simulated robot torque (1-dotted, 2-dashdot)

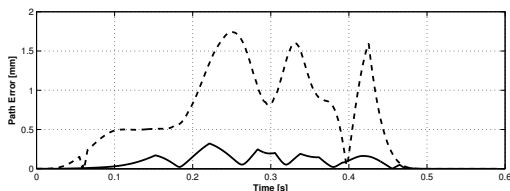


Fig. 8. Path Error for DAE FFW (solid) and Flexible Joint FFW (dashed)

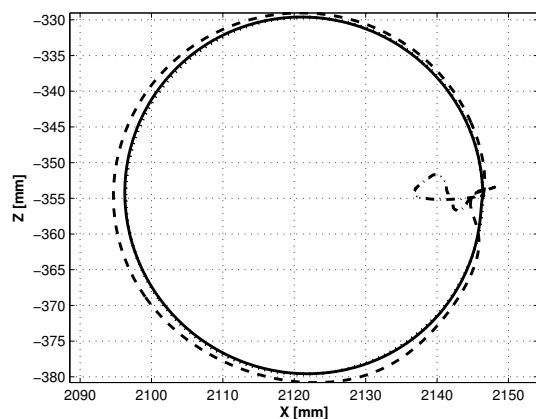


Fig. 9. Cartesian Path: Reference (dotted), DAE FFW (solid), Flexible Joint FFW (dashed) and PID Control (dashdot)

importance.

## REFERENCES

- [1] Alin Albu-Schäffer and Gerd Hirzinger. State feedback controller for flexible joint robots: A globally stable approach implemented on DLR's light-weight robots. In *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2000.
- [2] Wojciech Blajer. Dynamics and control of mechanical systems in partly specified motion. *Journal of the Franklin Institute*, 334B(3):407–426, May 1997.
- [3] Wojciech Blajer and Krzysztof Kolodziejczyk. A geometric approach to solving problems of control constraints: Theory and a dae framework. *Multibody System Dynamics*, 11(4):341–350, May 2004.
- [4] 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, 1996.
- [5] F. Caccavale and P. Chiacchio. Identification of dynamic parameters and feedforward control for a conventional industrial manipulator. *Control Eng. Practice*, 2(6):1039–1050, 1994.
- [6] John M. Hollerbach Chae H. An, Christopher G. Atkeson. *Model-Based Control of a Robot Manipulator*. The MIT press, 1988.
- [7] J. Öhr, S. Moberg, E. Wernholt, S. Hanssen, J. Pettersson, S. Persson, and S. Sander-Tavallaey. Identification of flexibility parameters of 6-axis industrial manipulator models. International Conference on Noise & Vibration Engineering, 2006.
- [8] K. P. Jankowski and H. Van Brussel. An approach to discrete inverse dynamics control of flexible-joint robots. *Journal of Dynamic Systems, Measurement, and Control*, 114:229–233, June 1992.
- [9] T. R. Kane and D. A. Levinson. *Dynamics: Theory and Applications*. McGraw-Hill Publishing Company, 1985.
- [10] A. Kumar and P. Daoutidis. *Control of Nonlinear Differential Algebraic Equation Systems with Applications to Chemical Processes*. Taylor & Francis Ltd, 1999.
- [11] M. Lesser. *The Analysis of Complex Nonlinear Mechanical Systems: A Computer Algebra assisted approach*. Singapore: World Scientific Series on Nonlinear Science, 2000.
- [12] Alessandro 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.
- [13] Alessandro De Luca and Pasquale Lucibello. A general algorithm for dynamic feedback linearization of robots with elastic joints. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, pages 504–510, Leuven, Belgium, May 1998.
- [14] Sven-Erik Mattson and Gustaf Söderlind. Index reduction in differential-algebraic equations using dummy derivatives. *SIAM Journal of Scientific Computing*, 14:677–692, 1993.
- [15] Stig Moberg and Sven Hanssen. A new approach to modelling and control of flexible manipulators. Technical Report LiTH-ISY-R-2769, Department of Electrical Engineering, Linköpings Universitet, 2007.
- [16] Stig Moberg and Jonas Öhr. Robust control of a flexible manipulator arm: A benchmark problem. 16th IFAC World Congress, 2005.
- [17] Pierre Rouchon, Michael Fliess, Jean Lévine, and Philippe 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.
- [18] Victor Santibanez and Rafael Kelly. PD control with feedforward compensation for robot manipulators: analysis and experimentation. *Robotica*, 19:11–19, 2001.
- [19] M. W. Spong. Modeling and control of elastic joint robots. *Journal of Dynamic Systems, Measurement, and Control*, 109:310–319, December 1987.
- [20] Michael Thümmel, Martin Otter, and Johann 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.