

# Symbolic Computation of the Inverse Dynamics of Elastic Joint Robots

Robert Höpler and Michael Thümmel

Institute of Robotics and Mechatronics

DLR Oberpfaffenhofen

D-82234 Wessling, Germany

[Robert.Hoeppler|Michael.Thuemmel]@dlr.de

**Abstract**—Efficient calculation of the inverse dynamics of robotic mechanisms serves as a basis for a variety of control schemes. Classical algorithms for rigid multibody systems do not apply when more realistic modeling is required. The computational treatment of the emerging non-linear equations augmented by drivetrain dynamics and elasticity imposes various challenges on the algorithmic solutions. This paper presents a method for recursive calculation of the inverse dynamics of real-world robots considering elasticity and gyroscopic effects introduced by the drivetrains. A study of the equations of motion elucidates the demand for analytical derivatives of the dynamics up to a maximum order depending on the number of joints. This problem is solved for the first time by means of higher-order time derivatives of spatial operators proposed by the Spatial Operator Algebra.

## I. INTRODUCTION

In the context of robotics *inverse dynamics* denotes the computation of driving forces from predefined trajectories, e.g., joint positions. The ability to calculate the inverse dynamics is a basic prerequisite for path planning algorithms and feed forward controllers in robotic systems. The computation of the inverse dynamics for rigid robots is a standard task [1]. Using the well-known equations of motion for tree-structured rigid multibody systems (MBS)

$$\mathcal{M}(q)\ddot{q} + \mathcal{C}(q, \dot{q}) + \mathcal{G}(q) = \tau, \quad (1)$$

and given the desired joint positions  $q(t)$  and their derivatives w. r. t. time up to a order of two, the force exerted by the drives  $\tau(t)$  can be calculated algebraically. When more realistic models including drive dynamics and elasticity in the joints are taken into account the computation becomes more complex: Derivatives of the desired trajectory must be given up to a certain order [2], [3], because the equations of motion must be differentiated several times in order to calculate  $\tau(t)$ .

When formulating the dynamics of an elastic robot it is convenient to use the Euler-Lagrange formalism, as shown for the cases of elastic links [4] and elastic joints [5]. This approach reveals the equations' structure without the necessity to express them explicitly. But, on the other hand, application of the Euler-Lagrange formalism for non trivial robots having many degrees of freedom (dof) imposes several difficulties. While the demanding formulation of the Lagrangian can be handled with special tools one drawback is inherent: depending on the chosen coordinates the formalism results in a small system

of large equations. Differentiation additionally increases their size and simplification by symbolic computational tools is still hardly tractable.

From this point of view it seems favourable to employ a recursive formalism, which leads to more compact equations and is scalable to a large number of dof. Forming the required derivatives of recursive equations is non-trivial for elastic joint robots, but since Rodriguez, et al. have introduced the Spatial Operator Algebra [6] and Park et al., have employed geometric arguments [7], symbolic manipulation of recursive equations has become feasible. This approach has been used, e.g. to linearize forward and inverse dynamics of tree-structured systems by first-order derivatives of rigid MBS equations [8], [9], [10], [11] and to calculate the forward dynamics of a certain class of elastic joint robots [12].

This paper is organized as follows: In Section II the investigated class of robots is presented, rigid link flexible joint robots including the complex gyroscopic couplings between drives and links. The 7-axes DLR light-weight robot is chosen as an example. The properties of the equations of motion and standard solution strategies are described in Section III. Section IV presents a method to differentiate the Newton-Euler equations expressed by spatial operators an arbitrary number of times symbolically. These results are linked to the drivetrain model and lead to a new algorithm discussed in Section V and an interesting closed-form interpretation of expressions from known Lagrangian approaches. Its application to the example system is shown in Section VI and Section VII presents conclusions and ideas for further work.

## II. INVESTIGATED ROBOTS

The robots considered in this paper have fixed base, rigid links and linear elasticity in the joints. For simplicity the analysis shown is restricted to chain-structured mechanisms, though all arguments given are valid for tree-structured MBS, too. Joints and links are labelled sequentially from base to tip starting from index  $i = 1$  as shown in Fig. 1, where link  $i$  follows joint  $i$ . Index 0 denotes the robot's base. Joints are supposed to have one mechanical dof and the total number of joints and dof is  $N$ . The drives are rotatory and mounted on the links, and the motor driving joint  $i$  is located on link  $i - 1$  as shown in Fig. 1. The drivetrain actuating joint  $i$  is modeled by the rotor of drive  $i$  controlled by torque  $\tau_i$ , driving

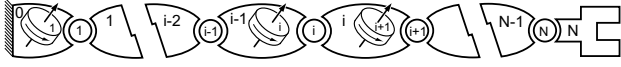


Fig. 1. Schematic view of an  $N$ -joint kinematic chain where each link is a gyrostat. Small circles are joints, ellipses rigid links, tilted cylinders are rotors. Numbering ascending from base to tip.

an ideal gear connected to a spring which is connected to link  $i$  as shown in Fig. 2. Dissipative effects like bearing friction and gear efficiency as well as nonlinear effects like spring stiffening, hysteresis and backlash are neglected. When the rotors of the drives show axial symmetry, they do not affect the mass distribution of the complete MBS. In consequence the robot can be viewed as a rigid MBS, where each link contains an internal source of angular momentum, i. e. each link is a *gyrostat* [13].

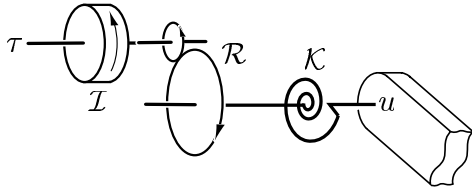


Fig. 2. Schematic view of drive-train model. From left to right: rotor with inertia  $\mathcal{I}$ , gear with ratio  $\mathcal{R}$ , torsional spring with spring constant  $\mathcal{K}$ , and link.

The method and algorithm presented will be applied to the DLR lightweight robot LBR 2 shown in Fig. 3a. This manipulator arm of approximately 1 m length is able to handle a payload of 8 kg with a total weight as low as 17 kg. The LBR 2 is a good example for practically relevant robots and clearly indicates the problems in computation of the inverse dynamics for non-trivial robots. It is a chain-structured mechanism with seven revolute joints as shown in Fig. 3b. The motors are mounted in the joints, their rotors' axes of rotation coincide with the joint axes, as depicted in Fig. 3c. The drivetrains are known to be elastic due to harmonic drive reduction gears and torque sensors. Joint stiffnesses are roughly  $10^4 \frac{\text{Nm}}{\text{rad}}$ . A more detailed description of the LBR 2 can be found in [14], [15].

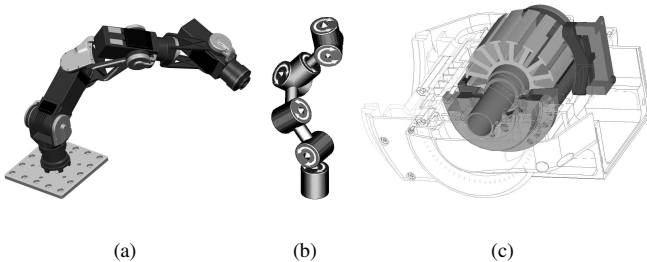


Fig. 3. DLR lightweight robot LBR 2. (a) Complete manipulator arm, (b) Sketch of kinematic structure, (c) View inside joint showing rotor and drive.

### III. CLASSICAL LAGRANGIAN APPROACH

A Lagrangian derivation of the equations of motion for an elastic joint robot using independent joint variables can be

found in, e. g. [16], and results in

$$\mathcal{M}(q)\ddot{q} + \mathcal{C}(q, \dot{q}) + \mathcal{G}(q) = u - u_{rotor} \quad (2)$$

$$-\mathcal{K}(q - \mathcal{R}\theta) = u \quad (3)$$

$$S(q)\ddot{\theta} + \mathcal{C}_{rotor}(q, \dot{q}, \dot{\theta}) = u_{rotor} \quad (4)$$

$$S(q)^T \ddot{q} + \mathcal{C}_{carrier}(q, \dot{q}) + \mathcal{I}\ddot{\theta} + \mathcal{R}u = \tau, \quad (5)$$

where  $q/\theta \in \mathbf{R}^N$  are joint/motor position variables,  $u/\tau \in \mathbf{R}^N$  are generalized joint/motor forces.  $\mathcal{M} \in \mathbf{R}^{N \times N}$  is the mass matrix,  $\mathcal{C} \in \mathbf{R}^N$  the vector of Coriolis and centrifugal terms of the links,  $\mathcal{K}$  the diagonal matrix of spring constants,  $\mathcal{R}$  the diagonal matrix of gear ratios,  $\mathcal{G}$  the vector of gravitational forces, and  $S$  is the matrix of inertial couplings between links and motors.  $\mathcal{C}_{rotor}$  is due to the spatial motion of the rotors' angular momentum,  $\mathcal{C}_{carrier}$  corrects for motion of the 'carrier' links and  $\mathcal{I}$  is the diagonal matrix of rotor inertias.  $\mathcal{R}$ ,  $\mathcal{K}$ , and  $\mathcal{I}$  are supposed to be constant. When the drives are mounted as described in Section II,  $S$  is upper triangular [16]:

$$S = \begin{pmatrix} 0 & S_{12}(q_1) & S_{13}(q_1, q_2) & \dots & S_{1N}(q_1, \dots, q_{N-1}) \\ 0 & 0 & S_{23}(q_2) & \dots & S_{2N}(q_2, \dots, q_{N-1}) \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \ddots & S_{N-1N}(q_{N-1}) \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix} \quad (6)$$

The  $i$ th component of  $\mathcal{C}_{rotor}$  explicitly depends on link variables and  $\theta_{i+1}, \dots, \theta_N$

$$\mathcal{C}_{rotor} = \begin{pmatrix} \mathcal{C}_{rotor1}(q, \dot{q}, \dot{\theta}_2, \dots, \dot{\theta}_N) \\ \mathcal{C}_{rotor2}(q, \dot{q}, \dot{\theta}_3, \dots, \dot{\theta}_N) \\ \vdots \\ \mathcal{C}_{rotorN-1}(q, \dot{q}, \dot{\theta}_N) \\ \mathcal{C}_{rotorN}(q, \dot{q}) \end{pmatrix}. \quad (7)$$

This model is similar to the one described in [3] but includes the more general case of non constant  $S$  and, hence a non-zero  $\mathcal{C}_{rotor}$ . Besides it shows the dependency on the gear ratios  $\mathcal{R}$ .

For calculation of the inverse dynamics using equations (2)-(7) one starts with the  $N$ th component equation of (2). Due to (6) and (7) this equation only depends on link variables and one is able to solve for  $\theta_N$ . Differentiating this equation twice w. r. t. time gives  $\dot{\theta}_N$  and  $\ddot{\theta}_N$ . Now one can solve the  $N$ th equation of (5) for  $\tau_N$ . Proceeding from tip to base it is possible to solve the second-last equation of (2) for  $\theta_{N-1}$  using  $\dot{\theta}_N$  and  $\ddot{\theta}_N$ . Again this equation needs to be differentiated twice to calculate  $\dot{\theta}_{N-1}$  and  $\ddot{\theta}_{N-1}$  what in turn requires  $\theta_N^{(3)}$  and  $\theta_N^{(4)}$  and therefore the third and fourth derivative of the  $N$ th equation of (2). Repeating this way one completes calculating  $\tau_1$  which requires derivatives of the equations of motion up to order  $\alpha = 2N$  and derivatives of the desired joint positions up to order  $2(N+1)$ . Greek index  $\alpha$  denotes the number of derivations.

This approach results in  $N$  very large algebraic equations for the motor forces which are awkward to handle and inefficient for computation. In the following sections a new non-Lagrangian approach is presented, which is amenable to efficient implementation while preserving the structure of the governing equations.

#### IV. SYMBOLIC HIGHER-ORDER TIME DERIVATIVES OF MULTIBODY EQUATIONS

In this section a method is developed to calculate derivatives of equation (2) in a symbolic and recursive manner employing the Spatial Operator Algebra (SOA) [6]. The SOA has proven to be a powerful formalism to derive equations for a wide range of MBS problems from spatial operator identities. It maintains maximum insight into the underlying physics and the structure of the equations, essential for further manipulation and efficient implementation. Key requisites are the notion of spatial vectors  $\in \mathbf{R}^6$ , a stacked notation of spatial operators [6], [17], and a flexible choice of coordinate representations for kinematics and dynamics equations [9]. Spatial vectors are composed of two vectors  $\in \mathbf{R}^3$ , where the first one represents rotational and the second one translational quantities, e. g. spatial velocity  $V := \begin{pmatrix} \omega \\ v \end{pmatrix}$ , with  $\omega$  angular and  $v$  translational velocity.

It turns out that an appropriate coordinate representation of the dynamics is a key issue to obtain compact expressions for higher-order spatial derivatives. The *body-fixed representation* [13] presents an efficient means, because expressing dynamics w.r.t. a body-fixed frame allows for recursions based on *local* time derivatives and essential spatial operators remain constant [17]. This leads to simpler construction of the recursions than in [7], so in the analysis below body-fixed representation will be applied. The *local* derivative of a spatial vector  $V_x$  w.r.t. an accelerated frame  $F_x$  is defined by

$$\overset{\circ}{V}_x := \frac{dV_x}{dt} - \tilde{\Omega}_x V_x \quad (8)$$

where  $\frac{dV_x}{dt}$  is the *absolute* time derivative w.r.t. to an inertial frame and the spatial angular velocity of  $F_x$  is  $\Omega_x := \begin{pmatrix} \omega_x \\ 0_3 \end{pmatrix}$ .  $\overset{\circ}{V}_x$  can be viewed as derivative w.r.t. a moving but non-rotating frame. The spatial generalization of the tilde-operator  $\tilde{\mathbf{a}}\mathbf{b} = \mathbf{a} \times \mathbf{b}$ ,  $\mathbf{a}, \mathbf{b} \in \mathbf{R}^3$  is used, the *spatial cross product operator* [17] defined as

$$\tilde{X} := \begin{pmatrix} \tilde{\mathbf{a}} & 0_{3 \times 3} \\ \mathbf{b} & \tilde{\mathbf{a}} \end{pmatrix}, \text{ where } X = \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix} \text{ and } \mathbf{a}, \mathbf{b} \in \mathbf{R}^3.$$

For further use the notion of the  $\alpha$ -times local derivative  $\overset{\circ}{V}^{(\alpha)}$  is introduced, with  $\alpha$  denoting the number of differentiations. The *rigid body transformation operator* [9], [6]

$$\phi_{i,i-1} := \begin{pmatrix} {}^i\mathbf{R}_{i-1} & 0_{3 \times 3} \\ -{}^{(i)}\tilde{\mathbf{p}}_{i-1,i} & {}^i\mathbf{R}_{i-1} \end{pmatrix}$$

relates spatial quantities expressed in local frame  $F_{i-1}$  located in the joint  $i-1$  to those expressed in  $F_i$  in link  $i$ , where  ${}^i\mathbf{R}_{i-1}(q_i)$  and  ${}^{(i)}\tilde{\mathbf{p}}_{i-1,i}$  are rotation and displacement from  $F_{i-1}$  to  $F_i$ . The spatial recursions are defined in terms of the stacked operators

$$\mathcal{E}_\phi := \begin{pmatrix} 0_{6 \times 6} & & & & & \\ \phi_{2,1} & \ddots & & & & \\ & \ddots & \ddots & & & \\ & & \ddots & \ddots & & \\ & & & \phi_{N,N-1} & 0_{6 \times 6} & \end{pmatrix} \quad (9)$$

$$\Phi := \begin{pmatrix} I_{6 \times 6} & & & & & \\ \phi_{2,1} & \ddots & & & & \\ \vdots & \ddots & \ddots & & & \\ \phi_{N,1} & \cdots & \phi_{N,N-1} & I_{6 \times 6} & & \end{pmatrix} \quad (10)$$

which are related by the identity [6]

$$\Phi^{-1} = I_{6N \times 6N} - \mathcal{E}_\phi. \quad (11)$$

Please note symbols and definitions slightly differ from the original ones introduced by Rodriguez, Jain, and Delgado [6], especially the numbering of bodies is reverted and follows robotics literature. When using a body-fixed dynamics representation and full stacked notation equations (2) and (4) can be expressed as the following recursive Newton-Euler equations

$$V = \Phi \Delta \quad (12)$$

$$\overset{\circ}{V} = \Phi(H\ddot{q} - \tilde{\Delta}V) \quad (13)$$

$$f = \Phi^T \{M(\overset{\circ}{V} + \dot{V}_g) - \tilde{V}^T M V + \dot{L}\} \quad (14)$$

$$u = H^T f. \quad (15)$$

The stacked operators used here are defined in the following, a right lower index  $X_{[i]}$  or  $X_{[i,j]}$  denotes the block matrix corresponding to link  $i$  or one pair of links  $i, j$ :

- spatial velocity  $V := \text{col}\{V_{[i]}\}$
- relative link velocity across  $i$ th joint  $\Delta_{[i]} := H_{[i]}\dot{q}_i$  and  $\Delta := \text{col}\{H_{[i]}\dot{q}_i\}$
- joint projection operator  $H := \text{diag}\{H_{[1]}, \dots, H_{[N]}\}$ , the linear mapping between joint and spatial coordinates. In case of revolute joints with axis  $\mathbf{n}_i$  it is  $H_{[i]} = \begin{pmatrix} \mathbf{n}_i \\ 0_3 \end{pmatrix}$ .
- spatial force  $f := \text{col}\{f_{[i]}\}$
- spatial gravitational acceleration  $\dot{V}_g := \text{col}\{\dot{V}_{g[i]}\}$
- spatial inertia matrix  $M := \text{diag}\{M_{[1]}, \dots, M_{[N]}\}$  where

$$M_{[i]} := \begin{pmatrix} \mathbf{J}_i & m_i \tilde{\mathbf{p}}_{i,cm_i} \\ -m_i \tilde{\mathbf{p}}_{i,cm_i} & m_i I_{3 \times 3} \end{pmatrix}$$

is the spatial inertia of link  $i$  including the motor mounted on it w.r.t. to frame  $F_i$ , where  $m_i$  is the total mass,  $\mathbf{J}_i$  the inertia matrix w.r.t.  $F_i$  and  $\tilde{\mathbf{p}}_{i,cm_i}$  the vector from  $F_i$  to the center of mass.

- rotor axis projection operator  $H_r$ , that shows the directions of the rotors' axes and on which link each motor is mounted. In case rotor  $i$  is mounted on link  $i-1$  the operator writes

$$H_r := \begin{pmatrix} 0_6 & H_{r[2]} & & & & \\ & \ddots & \ddots & & & \\ & & \ddots & \ddots & & \\ & & & H_{r[N]} & & \\ & & & & 0_6 & \end{pmatrix}, \quad (16)$$

- i.e., changing angular momentum  $\dot{L}_{[i]}$  contributes to  $f_{[i]}$ .
- vector of internal angular momentum of each gyrostatic link  $L := \text{col}\{L_{[i]}\}$ .

Except for operators  $M$  and  $\Phi$  which are  $\in \mathbf{R}^{6N \times 6N}$ , and  $H, H_r \in \mathbf{R}^{6N \times N}$ , all stacked spatial entities are  $\in \mathbf{R}^{6N}$ . For any stacked  $X \in \mathbf{R}^{6N}$  the stacked tilde operator  $\tilde{X}$  denotes a block-diagonal matrix  $\in \mathbf{R}^{6N \times 6N}$  where  $\tilde{X} = \text{diag}(\tilde{X}_{[1]}, \dots, \tilde{X}_{[N]})$ . For a more comprehensive treatment of this notation the reader is referred to, e.g., [9],

[6], [17]. A common approach to account for gravitation is to consider the base system accelerated by a gravitational acceleration [1] of  $\mathring{V}_{[0]} \equiv \mathring{V}_{g[0]}$  and the gravitational acceleration matrix  $\mathring{V}_g$  can be omitted. Important to note is, that  $\mathbf{H}$ ,  $\mathbf{H}_r$  and  $\mathbf{M}$  remain constant in body-fixed representation which simplifies derivatives w. r. t. time drastically.

Expanding  $\Phi$  using (11) leads to the well-known two sweep calculation, one outboard sweep from base to tip (12) and (13) and one tip-to-base inboard sweep (14) and (15). The main idea is to establish a recursion w. r. t. differentiation order  $\alpha$  for the standard recursive equations w. r. t. link index  $i$  analogously to equations (12)-(15). The goal is to arrive at an algorithm which is amenable to an efficient and straightforward implementation.

#### A. Kinematics derivatives

When restricting to the rigid MBS model described by (1) the calculation of the  $\alpha$ -times derivative of the dynamics requires derivatives of  $\mathcal{M}$  and  $\mathcal{C}$  up to an order of  $\alpha$ . As a consequence the spatial velocity  $\mathbf{V}$  has to be differentiated for  $\alpha + 1$  times. In absence of collision and contact it is possible to obtain derivatives of the equations of motion of arbitrary order what is due to the structure of the underlying smooth kinematics and dynamics equations [18]. To obtain a recursion w. r. t.  $\alpha$  we assume the  $\alpha$ -times local derivative of  $\mathbf{V}$  obeys the identity

$$\mathring{V}^{(\alpha)} = \Phi(A_\alpha + B_\alpha). \quad (17)$$

From (12) follow

$$A_0 = \Delta = \text{H}q^{(1)} \quad \text{and} \quad B_0 = 0 \quad (18)$$

needed to start the recursion. Differentiating (17) once locally w. r. t. time and using operator identities for the time derivatives of (9) and (10) found in [17]

$$\mathring{\mathcal{E}}_\phi = -\tilde{\Delta}\mathcal{E}_\phi \quad (19)$$

$$\mathring{\Phi} = -\Phi\tilde{\Delta}\mathcal{E}_\phi\Phi \quad (20)$$

leads to

$$\mathring{V}^{(\alpha+1)} = \Phi\left(A_\alpha - \tilde{\Delta}\mathcal{E}_\phi\mathring{V}^{(\alpha)} + B_\alpha\right). \quad (21)$$

The equivalences

$$\begin{aligned} A_{\alpha+1} &\equiv \mathring{A}_\alpha = \text{H}q^{(\alpha+2)} \\ B_{\alpha+1} &\equiv -\tilde{\Delta}\mathcal{E}_\phi\mathring{V}^{(\alpha)} + \mathring{B}_\alpha \end{aligned}$$

formally lead to the desired form of equation (17). By construction  $B_\alpha$  always fulfills  $B_\alpha = \sum_i b_{\alpha,i}$  where each summand is a product  $b_{\alpha,i} := k_i \prod_{j \in J_i} [\tilde{\Delta}^{(\alpha_j)}] \mathcal{E}_\phi \mathring{V}^{(\alpha_i)}$  with  $k_i$  is scalar and  $J_i \subset \{0, 1, \dots, \alpha - 1\}$ . This follows from identities (18) and (19).

#### B. Dynamics derivatives

If motor  $j$  is mounted on link  $i$  all of its inertia properties  $\mathbf{M}_{r[j]}$  can be added to the link resulting in a total  $\mathbf{M}_{[i]}$ . The rotor angular momentum *relative* to the link  $\mathbf{L}_i := \mathcal{I}_j \mathbf{n}_{r_j} \dot{\theta}_j$  has to be considered separately in order to calculate the correct

dynamics. Hereby  $\mathbf{n}_{r_j}$  is the rotor axis of rotation and  $\mathcal{I}_j$  the inertia about  $\mathbf{n}_{r_j}$ . The rotor spatial angular momentum hence is

$$\mathbf{L}_{[i]} := \mathbf{H}_{r[i,j]} \mathcal{I}_j \dot{\theta}_j = \begin{pmatrix} \mathbf{L}_i \\ 0_3 \end{pmatrix}.$$

A change in angular momentum

$$\frac{d}{dt} \mathbf{L}_{[i]} = \mathring{\mathbf{L}}_{[i]} + \tilde{\Omega}_{[i]} \mathbf{L}_{[i]} = \mathcal{I}_i (\mathbf{H}_{r[i,j]} \dot{\theta}_j + \tilde{\Omega}_{[i]} \mathbf{H}_{r[i,j]} \dot{\theta}_j) \quad (22)$$

expressed conveniently with circle derivative and stacked representation

$$\mathring{\mathbf{L}} = \mathring{\mathbf{L}} + \tilde{\Omega} \mathbf{L}$$

causes a torque which contributes to the dynamics. There is no need to apply a rigid body transformation from the center of rotor to the center of link, because this pure torque is independent of the point of application. When calculating the MBS dynamics the contribution of each link to the spatial force

$$\mathbf{f}_\delta := \mathbf{M} \mathring{\mathbf{V}} + \mathring{\mathbf{L}} - \tilde{\mathbf{V}}^T \mathbf{M} \mathbf{V} + \tilde{\Omega} \mathbf{L} \quad (23)$$

allows for a compact expression of the spatial recursion (14),  $\mathbf{f} = \Phi^T \mathbf{f}_\delta$ . Analogously to (17) one again can assume

$$\mathring{\mathbf{f}}^{(\alpha)} = \Phi^T (C_\alpha + D_\alpha) \quad (24)$$

and for  $\alpha = 0$  follows

$$C_0 = \mathbf{f}_\delta^{(0)} \quad \text{and} \quad D_0 = 0.$$

Differentiating (23)  $\alpha$ -times and using a binomial expansion one arrives at a compact non-recursive

$$\begin{aligned} \mathring{\mathbf{f}}_\delta^{(\alpha)} &= \mathbf{M} \mathring{V}^{(\alpha+1)} + \mathring{\mathbf{L}}^{(\alpha+1)} \\ &+ \sum_{j=0}^{\alpha} \binom{\alpha}{j} \left[ -\mathring{\tilde{\mathbf{V}}}^{(j)T} \mathbf{M} \mathring{V}^{(\alpha-j)} + \mathring{\tilde{\Omega}}^{(j)} \mathring{\mathbf{L}}^{(\alpha-j)} \right]. \end{aligned} \quad (25)$$

Differentiating (24) w. r. t. time

$$\mathring{\mathbf{f}}^{(\alpha+1)} = \Phi^T \left( \mathring{C}_\alpha - \mathcal{E}_\phi^T \tilde{\Delta}^T \mathring{\mathbf{f}}^{(\alpha)} + \mathring{D}_\alpha \right)$$

again leads to a symbolic recursion when identifying

$$\begin{aligned} C_{\alpha+1} &\equiv \mathring{C}_\alpha = \mathbf{f}_\delta^{(\alpha+1)} \\ D_{\alpha+1} &\equiv \mathring{D}_\alpha - \mathcal{E}_\phi^T \tilde{\Delta}^T \mathring{\mathbf{f}}^{(\alpha)}. \end{aligned}$$

## V. RECURSIVE ALGORITHM

In this section it is shown how elastic drivetrain and rigid multibody models are combined for computation of the inverse dynamics.

#### A. Combining elastic drivetrain and rigid multibody models

A strategy for solution of the complete problem becomes clear when expressions from (2)-(5) are identified in (12)-(15). Combining the latter together and using (22) gives

$$\mathbf{u} = \mathbf{H}^T \Phi^T \mathbf{M} \Phi \mathbf{H} \ddot{\mathbf{q}} \quad (26)$$

$$+ \mathbf{H}^T \Phi^T (-\mathbf{M} \Phi \tilde{\Delta} - \tilde{\mathbf{V}}^T \mathbf{M}) \mathbf{V} \quad (27)$$

$$+ \mathbf{H}^T \Phi^T \mathbf{H}_r \mathcal{I} \ddot{\theta} + \mathbf{H}^T \Phi^T \tilde{\Omega} \mathbf{H}_r \mathcal{I} \dot{\theta} \quad (28)$$

where (26) and (27) are standard factorizations of  $\mathcal{M}$  and  $\mathcal{C}$  [8]. Comparing expression (28) with (4) leads to the important operator factorizations of  $S$  and  $\mathcal{C}_{rotor}$ :

$$S(q) = \mathbf{H}^T \Phi^T \mathbf{H}_r \mathcal{I} \quad (29)$$

$$\mathcal{C}_{rotor}(q, \dot{q}, \dot{\theta}) = \mathbf{H}^T \Phi^T \tilde{\Omega} \mathbf{H}_r \mathcal{I} \dot{\theta}. \quad (30)$$

Equation (29) restates (6) in an *explicit* manner and it obviously results in the upper triangular shape derived in (6) from pure structural arguments. An arbitrary axis of rotation  $\mathbf{n}_i$  is invariant under its generated rotation,  ${}^i\mathbf{R}_{i-1}\mathbf{n}_i = \mathbf{n}_i$ . It follows that  $\phi_{i,i-1}(q_i)\mathbf{H}_{[i]} = \begin{pmatrix} \mathbf{n}_i \\ -{}^{(i)}\tilde{\mathbf{p}}_{i-1,i}\mathbf{n}_i \end{pmatrix}$  is independent of  $q_i$ . This observation shows that row  $i$  of the operator product  $\mathbf{H}^T \Phi^T$  in (29) is independent of  $q_i$ , thus revealing one more structural property of  $S(q)$ :

$$S_{i,j} = S(q_{i+1}, \dots, q_{j-1})_{i,j}, \quad 1 < i < j < N. \quad (31)$$

Drivetrain dynamics (5) depends on the *absolute* spatial angular momentum of rotor  $j$  mounted on link  $i$

$$\mathbf{L}_{abs[i]} = \mathbf{L}_{[i]} + \Omega_{[i]} \mathcal{I}_j.$$

Its *absolute* time derivative projected onto its axis of rotation using (13) is in case of revolute joints

$$\mathbf{H}_r^T \dot{\mathbf{L}}_{abs} = \mathcal{I} \ddot{\theta} + \mathbf{H}_r^T \Phi (\mathbf{H} \ddot{q} - \tilde{\Delta} \mathbf{V}) \mathcal{I}.$$

Now it is possible to establish the torque balance of the rotors

$$\tau = \mathcal{R}u + \mathcal{I} \ddot{\theta} + \mathbf{H}_r^T \Phi \mathbf{H} \ddot{q} \mathcal{I} - \mathbf{H}_r^T \Phi \tilde{\Delta} \mathbf{V} \mathcal{I} \quad (32)$$

which allows to identify

$$\mathcal{C}_{carrier}(q, \dot{q}) = -\mathbf{H}_r^T \Phi \tilde{\Delta} \mathbf{V} \mathcal{I} \quad (33)$$

using (5) and (29) and reveals

$$\mathcal{C}_{carrier}(q, \dot{q}) + S(q)^T \ddot{q} = \mathbf{H}_r^T \hat{\Omega} \mathcal{I}.$$

These were the last missing identities to express equations (2)-(5) completely in closed form by means of spatial operators. The algorithm presented in the following section is based on these symbolic operator factorizations.

### B. Numerical computation of the inverse dynamics

From operator expressions (17) and (24) one concludes the recursive algorithm will comprise two sweeps, one outboard to calculate velocities and all required higher order derivatives followed by one inboard sweep to do force and force derivative computations for rigid body and drive-train parts. The arguments discussed in Section III show how the structural properties of  $\mathbf{H}_r$  and  $S$  modify the simple two sweep execution logic of rigid MBS inverse dynamics. This leads to Algorithm 1.

It is important to note, that the required number of differentiations of the dynamics decreases from tip to base. The maximum order  $\alpha_{max}$  is required just for the outermost link.

Concerning code generation it is worthwhile to note a great potential for optimization.  $\mathbf{H}_r$  is sparse when each  $\mathbf{H}_{r[i]}$  is a unit vector in local coordinates and  $\widetilde{\mathbf{H}_{r[i]}} \widetilde{\mathbf{H}_{r[i]}} = \mathbf{I}_{6 \times 6}$  results in simple expressions for  $B_\alpha$  needed in (17), even for large  $\alpha$ . Loss in numerical precision might occur likewise for all evaluations of higher-order Taylor expansions. The explicitness of spatial operator expressions helps in taking precautions against that problem.

---

### Algorithm 1 Inverse dynamics algorithm

---

Detect permanent zeros of  $S(q)$  from MBS topology

**Require:**  $S(q)$  is upper triangular

Calc. max. order  $\alpha_{max}[i]$  for each  $u_i$  from (4)

$$\alpha_{max} = \max_{i=1, \dots, N} \alpha_{max}[i]$$

$$\mathring{\mathbf{V}}^{(\alpha)}_{[0]} = 0_6, \quad \alpha \in \{0, \dots, \alpha_{max} + 1\}$$

**for**  $i=1$  to  $N$  **do**    */\* Outboard sweep \*/*

**for**  $\alpha=0$  to  $\alpha_{max}+1$  **do**

$$\mathring{\mathbf{V}}^{(\alpha)}_{[i]} = \phi_{i,i-1} \mathring{\mathbf{V}}^{(\alpha)}_{[i-1]} + A_\alpha[i] + B_\alpha[i]$$

**end for**

**end for**

$$\mathring{\mathbf{f}}^{(\alpha)}_{[N+1]} = 0_6, \quad \alpha \in \{0, \dots, \alpha_{max}\}$$

**for**  $i=N$  to  $1$  **do**    */\* Inboard sweep \*/*

**for**  $\alpha=0$  to  $\alpha_{max}[i]$  **do**

$$\mathring{\mathbf{f}}^{(\alpha)}_{[i]} = \phi_{i+1,i}^T \mathring{\mathbf{f}}^{(\alpha)}_{[i+1]} + C_\alpha[i] + D_\alpha[i]$$

$$\theta_i^{(\alpha)} = \left( \frac{u_i^{(\alpha)}}{\mathcal{K}_{i,i}} + q^{(\alpha)} \right) \frac{1}{\mathcal{R}_{i,i}}$$

**end for**

$$\tau_i = \mathcal{R}_{i,i} u_i + \mathcal{I}_{i,i} \ddot{\theta}_i + \mathbf{H}_{r[i]}^T \hat{\Omega}_{[i]} \mathcal{I}_{i,i}$$

**end for**

---

The complexity of the Newton-Euler inverse dynamics for a rigid robot is  $\mathcal{O}(N)$ . The maximum possible number of differentiations is  $2(N+1)$  as discussed in Section III. The complexity of expressions (17) and (24) grows linearly in  $\alpha$ . Analysis of the nested loops in outboard and inboard sweeps in Algorithm 1 shows that without any further symbolic simplifications this algorithm is at the worst  $\mathcal{O}(N^3)$ .

## VI. INVERSE DYNAMICS FOR THE DLR LBR 2

Application of the methods introduced in Sections IV and V to the LBR 2 model reveals additional properties of this robot. As pointed out in Section III the equations of motion need to be differentiated up to an order of  $2N$  in the general case. The first off-diagonal of  $S(q)$  vanishes according to equation (29) because consecutive axes of the robot are orthogonal as depicted in Figure 3b. Considering (31)  $S(q)$  gives

$$S(q) = \begin{pmatrix} 0 & 0 & S_{13}(q_2) & \dots & S_{17}(q_2, \dots, q_6) \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & S_{57}(q_6) \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix}.$$

This reduces the required number of derivatives to 12 since solution of the sixth equation of (2) no longer depends on  $\theta_7$ .

The example trajectory  $q_1(t) = \dots = q_7(t)$ ,  $t \in [0, 2.5s]$  shown in the inset of Figure 4 was chosen to be a step input of 2 rad at time  $t = 0$  filtered by a filter of order 16 what ensures the required smoothness for differentiation [19]. The filter dynamics chosen provides a trajectory that satisfies motor torque and velocity constraints. Algorithm 1 has been implemented using C++ methods presented in [20] to calculate the elastic joint inverse dynamics. The obtained motor torques of three axes  $\tau_1, \tau_2, \tau_3$  are shown in Figure 4. The influence of drivetrain effects on the dynamics are shown in Fig. 5

illustrated by the difference in motor torques between the full elastic model and the rigid model. The results significantly differ even for the very smooth trajectory chosen. For faster motion this effect increases due to larger derivatives. In case of high-speed movements and high precision requirements drivetrain effects should be taken into account.

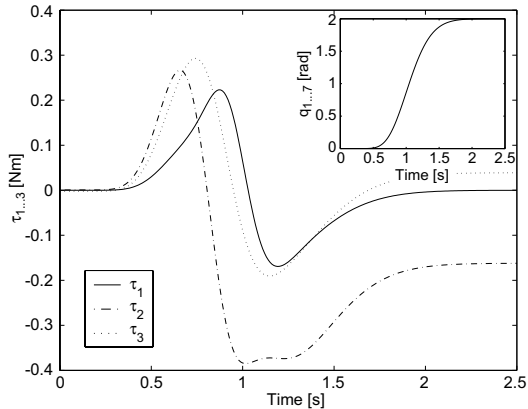


Fig. 4. Inset: Example trajectory used in simulations. Large plot: Resulting motor torques  $\tau_1, \tau_2$ , and  $\tau_3$  from elastic joint inverse dynamics computed using Algorithm 1.

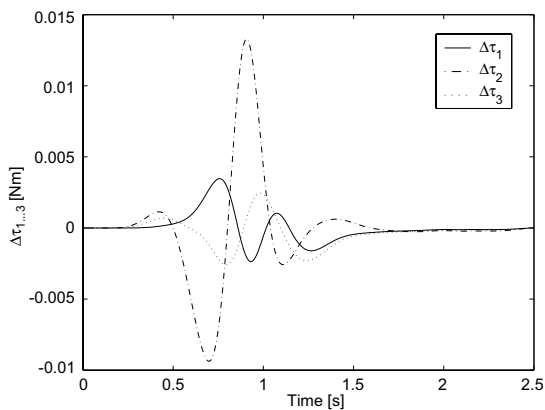


Fig. 5. Differences between elastic model and rigid model inverse dynamics for the trajectory from Fig. 4.

## VII. CONCLUSION

This paper presented a new method for symbolic, recursive calculation of the inverse dynamics of elastic joint robots including gyroscopic effects introduced by the drivetrains. It was shown that the computation of higher-order time derivatives of the rigid-body equations is a key problem. To overcome the difficulties arising from standard Lagrangian approaches the problem was solved for the first time using higher-order time derivatives of spatial operators. This approach results in an efficient algorithm while preserving the structure of the system's equations leading to valuable explicit operator factorizations for expressions stemming from the Lagrangian approach. These permit analysis of, e.g., gyroscopic effects directly and ameliorates the potential for efficient code generation. The algorithm presented applies to a wide range of schemes, e.g., path planning and feedforward control. The relevance of the considered physical effects was shown using simulation results. For the LBR 2 motor torques differ significantly when comparing standard rigid-body inverse dynamics

and the presented elastic joint inverse dynamics. Future work will complete the robot model by adding drive train effects like non-linear springs and friction and increase efficiency by further exploitation of the complex structure of the elastic joint dynamics equations.

## ACKNOWLEDGEMENTS

The authors would like to thank the DLR light-weight robot team for providing data and drawings of the LBR 2, and Michael Hardt, PhD, (Siemens VDO) for valuable comments.

## REFERENCES

- [1] J. Y. S. Luh, M. W. Walker, and R. P. C. Paul, "On-line computational scheme for mechanical manipulators," *ASME J. of Dynamic Systems Meas. and Control*, vol. 102, pp. 69–76, 1980.
- [2] A. De Luca and P. Lucibello, "A general algorithm for dynamic feedback linearization of robots with elastic joints," in *Proc. IEEE Conf. on Robotics and Automation*, Leuven, Belgium, May 1998, pp. 504–510.
- [3] A. De Luca, "Feedforward/feedback laws for the control of flexible robots," in *Proc. IEEE Conf. on Robotics and Automation*, San Francisco, CA, USA, Apr. 2000, pp. 233–240.
- [4] W. J. Book, "Recursive Lagrangian dynamics of flexible manipulator arms," *The Int. J. Robotics Res.*, vol. 3, no. 3, pp. 87–101, 1984.
- [5] M. W. Spong, "Modeling and control of elastic joint robots," *Transactions of the ASME*, vol. 109, pp. 310–319, 1987.
- [6] G. Rodriguez, A. Jain, and K. Kreutz-Delgado, "A spatial operator algebra for manipulator modeling and control," *The Int. J. Robotics Res.*, vol. 10, no. 4, pp. 371–381, 1991.
- [7] F. C. Park, J. E. Bobrow, and S. Ploen, "A Lie group formulation of robot dynamics," *The Int. J. Robotics Res.*, vol. 14, no. 6, pp. 609–618, Dec. 1995.
- [8] A. Jain and G. Rodriguez, "Linearization of manipulator dynamics using spatial operators," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 1, pp. 239–248, 1993.
- [9] M. W. Hardt, "Multibody dynamics algorithms, numerical optimal control, with detailed studies in the control of jet engine compressors and biped walking," Ph.D. dissertation, University of California, San Diego, 1999.
- [10] A. Jain and G. Rodriguez, "Sensitivity analysis of multibody dynamics using spatial operators," in *Proceedings of the 6th international conference on methods and models in automation and robotics*, Miedzyzdroje, Poland, 2000, pp. 573–578.
- [11] G. A. Sohl and J. E. Bobrow, "A recursive multibody dynamics and sensitivity algorithm for branched kinematic chains," *ASME J. of Dynamic Systems Meas. and Control*, vol. 123, pp. 391–399, Sept. 2001.
- [12] S. H. Murphy, J. T. Wen, and G. N. Saridis, "Simulation and analysis of flexibly jointed manipulators," in *Proc. IEEE Conf. on Decision and Control*, Honolulu, Hawaii, USA, Dec. 1990, pp. 545–550.
- [13] J. Wittenburg, *Dynamics of systems of rigid bodies*. Stuttgart: B. G. Teubner, 1977.
- [14] G. Hirzinger, A. Albu-Schäffer, M. Hähle, I. Schaefer, and N. Sporer, "On a new generation of torque controlled light-weight robots," in *Proc. IEEE Conf. on Robotics and Automation*, Seoul, Korea, May 2001, pp. 3356–3363.
- [15] A. Albu-Schäffer, "Regelung von Robotern mit elastischen Gelenken am Beispiel der DLR-Leichtbauarme," Ph.D. dissertation, Technische Universität München, 2002.
- [16] A. De Luca and P. Tomei, "Elastic joints," in *Theory of robot control*, C. C. de Wit, B. Siciliano, and G. Bastin, Eds. Berlin, Heidelberg, New York, Tokyo: Springer Verlag, 1996, pp. 179–217.
- [17] A. Jain and G. Rodriguez, "Diagonalized Lagrangian robot dynamics," *IEEE Trans. Robotics and Automat.*, vol. 11, pp. 571–584, Aug. 1995.
- [18] R. M. Murray, Z. Li, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. Boca Raton: CRC Press, 1993.
- [19] M. Thümmel, M. Otter, and J. Bals, "Control of robots with elastic joints based on automatic generation of inverse dynamics models," in *Proc. IEEE Conf. on Intelligent Robots and Systems*, Maui, Hawaii, USA, 2001, pp. 925–930.
- [20] R. Höppler and M. Otter, "A versatile C++ toolbox for model based real-time control systems of robotic manipulators," in *Proc. IEEE Conf. on Intelligent Robots and Systems*, Maui, Hawaii, USA, 2001, pp. 2208–2214.