Appl. Math. and Comp. Sci., 1995, vol.5, No.2, 305–328

ROBOT DYNAMICS MODELS IN TERMS OF GENERALIZED AND QUASI–COORDINATES: A COMPARISON[†]

KRZYSZTOF KOZŁOWSKI*

Lagrangian robot dynamics derived by making use of the generalized and quasi-coordinates is discussed. A comparison between two sets of equations is presented, which is motivated by recently developed (Jain and Rodriguez, 1994) diagonalized equations of motion for multibody open kinematic chains with N degrees of freedom. These equations are derived by making use of new joint velocity variables ν_i, which can be viewed as time derivatives of the Lagrangian 'quasi-coordinates, similar to those in classical mechanics. Several remarks, which simplify the derivation of the new set of equations are presented. Next, physical interpretation of both sets of equations is widely discussed. A detailed computational complexity analysis of the standard and diagonalized Lagrangian robot dynamics algorithms is presented in the paper. The results have been compared with those existing in the literature on robotics.

Nomenclature

Individual Body Quantities

- N number of joints and, at the same time, number of degrees of freedom,
- O_k origin of the coordinate frame attached to the k-th link, which is located on the negative side of the k-th joint, namely inboard the k-th joint,
- $p(k) \in \mathbb{R}^3$ vector from O_k to the k-th link's centre of mass,
- $l(k,k-1) \in \mathbb{R}^3$ vector from O_{k-1} to O_k,
- $\tilde{l}(k,k-1) \in \mathbb{R}^{3\times 3}$, $\tilde{l}(k,k-1) = \begin{bmatrix} 0 & -l_{kz} & l_{ky} \\ l_{kz} & 0 & -l_{kx} \\ -l_{ky} & l_{kx} & 0 \end{bmatrix}$ where l_{kx}, l_{ky}, l_{kz} are components of vector $l(k,k-1)$,
- I_3 3×3 identity matrix,

[†] This paper describes research done at Jet Propulsion Laboratory in 1993. Support for the author was provided by the Fulbright Award.

* Technical University of Poznań, Department of Control, Robotics and Computer Science, ul. Piotrowo 3A, 60–965 Poznan, Poland

- $\phi(k,k-1) \in \mathbb{R}^{6\times 6}$ composite body transformation operator from O_{k-1} to O_k,
$\phi(k,k-1) = \begin{bmatrix} I_3 & \tilde{l}(k,k-1) \\ 0 & I_3 \end{bmatrix}$,
- $m(k)$ mass of the k-th link,
- $I(k) \in \mathbb{R}^{3\times 3}$ inertia tensor of the k-th link with respect to O_k,
- $M(k) \in \mathbb{R}^{6\times 6}$ spatial inertia matrix of the k-th link expressed in the coordinate O_k,
$M(k) = \begin{bmatrix} I(k) & m(k)\tilde{p}(k) \\ -m(k)\tilde{p}(k) & m(k)I_3 \end{bmatrix}$,
- $V(k) = \mathrm{col}\,[\omega(k), v(k)] \in \mathbb{R}^6$ spatial velocity of the k-th body frame O_k, where $\omega(k)$ and $v(k)$ are the angular and linear velocities of O_k, col denotes a column vector,
- $\alpha(k) = \mathrm{col}\,[\dot{\omega}(k), \dot{v}(k)] \in \mathbb{R}^6$ spatial acceleration vector of the k-th body frame O_k, where $\dot{\omega}(k)$ and $\dot{v}(k)$ are local time derivatives of $\omega(k)$ and $v(k)$,
- $H^*(k)\dot{\theta}(k)$ relative spatial velocity across the k-th joint, where $H^*(k) = \begin{bmatrix} h(k) \\ 0 \end{bmatrix} \in \mathbb{R}^6$ is the joint map matrix for the k-th joint with 0 as 3×1 zero matrix and $h(k)$ axis of rotation for the k-th joint, and $H^*(k) = \begin{bmatrix} 0 \\ h(k) \end{bmatrix}$ for a translational joint,
- $f(k) = \mathrm{col}\,[N(k), F(k)] \in \mathbb{R}^6$ spatial force of interaction across the k-th joint, where $N(k)$ and $F(k)$ are the moment and force components, respectively,
- $b(k) \in \mathbb{R}^6$ spatial bias forces vector (for both types of joints),
$$b(k) = \begin{bmatrix} \omega(k)\times I(k)\omega(k) + m(k)p(k)\times(\omega(k)\times v(k)) - N(ke) - p(k)\times F(ke) \\ m(k)p(k)\times v(k) + m(k)\omega(k)\times(\omega(k)\times p(k)) - F(ke) \end{bmatrix}$$
where $F(ke)$ and $N(ke)$ stand for the external force and moment acting at the centre of mass of the k-th link,
- $a(k) \in \mathbb{R}^6$ spatial bias accelerations vector; $a(k) = \begin{bmatrix} \omega(k)\times h(k)\dot{\theta}(k) \\ v(k)\times h(k)\dot{\theta}(k) \end{bmatrix}$ for a rotational joint, and $a(k) = \begin{bmatrix} 0 \\ \omega(k)\times h(k)\dot{\theta}(k) \end{bmatrix}$ for a translational joint,
- $P(k) \in \mathbb{R}^{6\times 6}$ articulated inertia of the manipulator outboard (toward the tip) of the k-th joint,
- $D(k) \in \mathbb{R}^1$ inertia along the k-th joint axis of the bodies formed by the links outboard of this joint; notice that all of the joints outboard of the k-th joint are unlocked in defining the inertia $D(k)$,

- $G(k) \in \mathbb{R}^6$ gain computed from the articulated inertia $P(k)$,
- $\Psi(k+1,k) \in \mathbb{R}^{6\times 6}$ operator which transforms articulated quantities
$$\Psi(k+1,k) = \Phi(k+1,k)\,[I - G(k)H(k)]\,,$$
- $r(k) \in \mathbb{R}^{6\times 6}$ composite body inertia matrix seen from the k-th coordinate frame,
- $A(k) \in \mathbb{R}^{6\times 6}$ spatial direction cosine matrix between the k-th coordinate frame and the $(k+1)$-th coordinate frame, $A(k) = \begin{bmatrix} {}^{k+1}_{k}R & 0 \\ 0 & {}^{k+1}_{k}R \end{bmatrix}$, where ${}^{k+1}_{k}R$ is the direction cosine matrix between the frames assigned according to the modified Denavit–Hartenberg notation,
- $A^{-1}(k) \in \mathbb{R}^{6\times 6}$ spatial direction cosine matrix between the $(k+1)$-th coordinate frame and the k-th coordinate frame,
- $[A, B] = AB - BA$ commutator defined for two square matrices.

Spatial block diagonal quantities used in "stacked" notation

- $\theta = \mathrm{col}\,[\theta(1), \ldots, \theta(N)]$ vector of generalized positions. Similarly one can define the following vectors; $\tau = \mathrm{col}\,[\tau(1), \ldots, \tau(N)]$, $V = \mathrm{col}\,[V(1), \ldots, V(N)]$,
$$f = \mathrm{col}\,[f(1), \ldots, f(N)]\,, \quad \alpha = \mathrm{col}\,[\alpha(1), \ldots, \alpha(N)]\,, \quad \dot\theta = \mathrm{col}\left[\dot\theta(1), \ldots, \dot\theta(N)\right],$$
$$a = \mathrm{col}\,[a(1), \ldots, a(N)]\,, \quad b = \mathrm{col}\,[b(1), \ldots, b(N)]\,, \quad \ddot\theta = \mathrm{col}\left[\ddot\theta(1), \ldots, \ddot\theta(N)\right],$$
where τ is the vector of generalized forces,
- $H = \mathrm{diag}\,[H(k)] \in \mathbb{R}^{6N\times N}$ diagonal matrix of the joint map,
- $M = \mathrm{diag}\,[M(k)] \in \mathbb{R}^{6N\times 6N}$ diagonal matrix of the spatial inertia matrix of all the links; similarly one can define $P \in \mathbb{R}^{6N\times 6N}$ and $G \in \mathbb{R}^{6N\times N}$ matrices,
- $\Phi = (I - \mathcal{E}_\Phi)^{-1} \in \mathbb{R}^{6N\times 6N}$ causal (lower triangular) matrix defined as

$$\mathcal{E}_\Phi = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 & 0 \\ \Phi(2,1) & 0 & 0 & \ldots & 0 & 0 \\ 0 & \Phi(3,2) & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & \Phi(N,N-1) & 0 \end{bmatrix}$$

$$\Phi = \begin{bmatrix} I & 0 & \ldots & 0 \\ \Phi(2,1) & I & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \Phi(N,1) & \Phi(N,2) & \ldots & I \end{bmatrix}$$

with $\Phi(i,j) = \Phi(i,i-1)\cdots\Phi(j+1,j)$ for $i > j$, $\mathcal{E}_\Phi$ is the rigid shift operator and I is the 6×6 identity matrix. Similarly, we can define Ψ and $\mathcal{E}_\Psi$ matrices with $\Psi(i,j)$ as entries as a lower block triangular operator and the corresponding shift operator which transfer the articulated quantities.

1. Introduction

In this paper, the Lagrangian robot dynamics derived by making use of the generalized and quasi-coordinates is discussed. The standard equations of motion for a manipulator with N degrees of freedom can be found in many textbooks, see e.g. (Craig, 1986). They can also be recast as a result of the Kalman filtering and smoothing algorithms when considering the manipulator as a sequence of links, where an individual link is a discrete step in space. This approach was developed by Rodriguez (1987) where spatial operator techniques (Rodriguez and Kreutz, 1988) were used to develop the equations of motion.

For the purpose of control the mass matrix of the manipulator should be diagonal. The diagonalization of the equations of motion is the heart of the Hamiltonian mechanics. Koditschek (1985) and Bedrossian (1992) have noted that if the inertia matrix $\mathcal{M}(\theta)$ of an N-link manipulator can be factored as $L^*(\theta)L(\theta)$, where $L(\theta)$ is integrable, i.e. the Jacobian of a function $Q(\theta)$, then Q and $P = L(\theta)\dot{\theta}$ define a canonical transformation relative to which the robot dynamics equations are particularly simple. Spong (1992) found the necessary and sufficient conditions for the existence of such factorization. The solution to this problem is known from the Riemannian geometry. The mass matrix of the manipulator defines a Riemannian metric on the configuration manifold of an N-link robot. Koditschek (1985) has noted that the function $Q(\theta)$, if exists, is an isometry between the Riemannian manifold and Euclidean manifold. The necessary and the sufficient condition for the local existence of the isometry $Q(\theta)$ is that the Riemannian manifold defined by the manipulator matrix $\mathcal{M}(\theta)$ must be locally flat. This condition can be checked by making use of the Riemann symbols of the first and second kind (Levi-Civita, 1950). It happens that this condition is restrictive and rarely satisfied in practice (Spong, 1992).

Jain and Rodriguez (1994) proposed a diagonalization in velocity space. This new diagonalization uses a concept of quasi-coordinates which result from the factorization of the mass matrix developed by Rodriguez and Kreutz (1988). A new set of velocities ν, are not time derivatives of any vector of configuration variables.

In this paper, we review the equations obtained by Jain and Rodriguez (1994) and give some insight into derivation and physical interpretation of these equations. We examine time differentiation of different quantities such as scalars, spatial vectors and spatial matrices. We show that spatial matrices can be treated as tensors (Wittenburg, 1977) and time differentiation of tensors can be implemented. We discuss time differentiation in both local and global coordinate frames and explain why some of the derivatives are the same. The diagonalized equations of motion involve articulated body quantities (Rodriguez and Kreutz, 1988) and standard spatial quantities such as bias forces and accelerations vectors. The interpretiation of these quantities in both standard and diagonalized equations of motion are widely discussed in the paper. It is pointed out that there exists some relevance between these two sets of equations. It is also shown how to write both sets of equations in the component form, without knowing their derivation.

In the paper, two examples are considered: one for the standard equations and the other for the diagonalized equations of motion. Each component of these equations is clearly interpreted from a physical point of view.

Apart from that, we investigate the computational complexity of the presented algorithms. Results of this analysis are compared to those existing in the robotics literature on robotics (Brandl *et al.*, 1986; Featherstone, 1987; Khosla, 1986; Kozłowski 1992; 1993; Walker and Orin, 1982). It is shown that the diagonalized equations of motion have the complexity comparable to those presented by Walker and Orin (1982) for $N = 6$. They are not as fast as those analysed by Brandl *et al.* (1986). This is mainly due to the time differentiation of the articulated body inertia matrix and rather costly calculations of $C(\theta, \nu)$, which accounts for all quadratic terms $\nu_i \nu_j$. We optimize these calculations taking into account a particular structure of the matrices involved in the calculations.

The paper is organized as follows. In Section 2 time derivatives of different quantities are discussed. The next section is devoted to the standard equations of motion. A simple example is considered. Interpretation of each term of these equations is discussed. Diagonalized equations of motion (both normalized and unnormalized) are presented in Section 4. In Section 5, an example of the diagonalized equations of motion is presented and very detailed interpretation is developed. Computational complexity of the related algorithms is the main topic of the next section. Finally, concluding remarks end the paper.

2. Time Derivatives of Different Quantities

Time derivatives of various quantities such as scalars, vectors, matrices, and block diagonal matrices, play a very important role in deriving equations of motion. Time differentiation can be calculated in any coordinate frame. One can perform time differentiation in local coordinate frame in which a particular quantity is observed, and also at the base (inertial) coordinate frame. It is obvious that the results are different due to the movement of the local frame. Through the paper we deal with spatial velocities, vectors, and matrices. Therefore, a spatial vector has six elements, the first three of which are "responsible" for rotational movement and three others are "responsible" for translational movement.

Consider an arbitrary spatial vector $x \in \mathrm{IR}^{6\times 1}$, which is a column vector having six components. A local time derivative is denoted by an operator $\frac{\mathrm{d}}{\mathrm{d}t}$, and $\frac{\mathrm{d}^k}{\mathrm{d}t} x$ denotes the time derivative of the vector x, where the time differentiation is performed in the k-th coordinate system in which the vector x is defined. The integer k can be any number between 1 and N. For a particular situation where $k = N + 1$ (the links are numbered from the tip of the manipulator towards its base) we denote time differentiation operator by $\frac{\mathrm{D}}{\mathrm{Dt}}$, which is equivalent to the inertial time differentiation.

The inertial and local time derivatives are expressed by the following formula:

$$\frac{\mathrm{D}x}{\mathrm{D}t} = \frac{\mathrm{d}^k x}{\mathrm{d}t} + \Omega(k)x \tag{1}$$

where $\Omega(k)$ is defined by

$$\Omega(k) = \begin{bmatrix} \tilde{\omega}(k) & 0 \\ 0 & \tilde{\omega}(k) \end{bmatrix} \tag{2}$$

In eqn. (2) $\omega(k)$ denotes the angular velocity of the k-th coordinate frame with respect to the inertial (base) coordinate frame; $\tilde{\omega}(k)$ stands for a 3×3 skew symmetric matrix, compare Nomenclature, which represents an equivalent matrix operation for the cross product operation. In order to prove eqn. (1) notice that it is a superposition of two time derivatives of an arbitrary vector c in two different bases to which a simple formula $\frac{\mathrm{D}c}{\mathrm{D}t} = \frac{\mathrm{d}^k c}{\mathrm{d}t} + \tilde{\omega}(k)c$ can be applied; compare with (Wittenburg, 1977). As another example of eqn. (1) one can consider time differentiation of the vector x defined in the k-th coordinate frame with respect to the next coordinate frame, namely $k+1$. This operation results in the following expression:

$$\frac{\mathrm{d}^{k+1} x}{\mathrm{d}t} = \frac{\mathrm{d}^k x}{\mathrm{d}t} + \Omega_\delta(k)x \tag{3}$$

where $\Omega_\delta(k) = \tilde{H}(k)\dot{\theta}(k)$ and $\tilde{H}(k)$ is a 6×6 matrix

$$\tilde{H}(k) = \begin{bmatrix} \tilde{h}(k) & 0 \\ 0 & \tilde{h}(k) \end{bmatrix} \tag{4}$$

Recall that $h(k)$ denotes an axis of rotation or translation for the k-th joint (compare Nomenclature), which is normally the z-axis of the local k-th coordinate frame. Notice that for the manipulator with only rotational joints, the angular velocity $\omega(k)$ of the k-th coordinate frame with respect to the inertial frame has the following form (in coordinate-free notation)

$$\omega(k) = \sum_{i=k}^{N} h(i)\dot{\theta}(i) \tag{5}$$

Therefore, the k-th frame rotates with respect to the $(k+1)$-th frame with angular velocity $h(k)\dot{\theta}(k)$. This explains eqns. (3) and (4).

Consider now an arbitrary spatial 6×6 matrix X, e.g. the inertia matrix of the k-th link. Then local (with respect to the k-th coordinate frame) and inertial time derivatives are related by

$$\frac{\mathrm{D}X}{\mathrm{D}t} = \frac{\mathrm{d}^k X}{\mathrm{d}t} + \Omega(k)X - X\Omega(k) = \frac{\mathrm{d}^k X}{\mathrm{d}t} + \left[\Omega(k), X\right] \tag{6}$$

where $\Omega(k)$ is defined by eqn. (2) and we have used a notion of a commutator of two matrices (Arnold, 1978). In particular, differentiation of X with respect to the next coordinate frame results in

$$\frac{\mathrm{d}^{k+1}X}{\mathrm{d}t} = \frac{\mathrm{d}^{k}X}{\mathrm{d}t} + \left[\Omega_\delta(k), X\right] \tag{7}$$

In order to prove eqn. (6) consider an arbitrary spatial matrix X which has dimensions 6×6, namely consists of four block matrices each being 3×3. Suppose that X is expressed in the k-th coordinate frame. Then in order to express this matrix in $(k+1)$-th coordinate frame, it is necessary to perform the operation $A(k)XA^{-1}(k)$, where $A(k)$ is the spatial direction cosine matrix (compare Nomenclature) between two successive coordinate frames k and $k+1$. From this it is easy to conclude that each block matrix is multiplied on the left and right hand sides by the ${}_{k}^{k+1}R$ and ${}_{k+1}^{k}R$ matrices (we denote elements of matrix ${}_{k}^{k+1}R$ by r_{ij}, $i,j = 1,2,3$). Now, denoting elements of the block matrix by b_{ij}^{m}, $i,j = 1,2,3$ for $m = 1,2,3,4$, the transformed elements $b_{kl}^{m'}$ can be written in the following form:

$$b_{kl}^{m'} = \sum_{i=1}^{3}\sum_{j=1}^{3} r_{ki} r_{lj} b_{ij}^{m}, \qquad k,l = 1,2,3, \text{ and } m = 1,2,3,4 \tag{8}$$

From the last equation it is clear that b_{ij} are the elements of the metric tensor of the second order (Levi-Civita, 1950). Concluding it is easy to notice that eqn. (6) denotes time differentiation of tensor quantities in two different coordinate frames, local and inertial, which are Cartesian coordinates (not curvilinear coordinates as considered in general by Levi-Civita (1950)).

As an example, consider a spatial inertia matrix $M(k)$. Its local time derivative (with respect to the k-th coordinate frame) is zero because the components of the matrix $M(k)$ are constant when calculated with respect to the k-th coordinate frame. Inertial time derivative has the following form (compare the structure of the matrix $M(k)$ as having tensor quantities):

$$\frac{\mathrm{D}M(k)}{\mathrm{D}t} = \left[\Omega(k), M(k)\right]$$

Now, consider a block diagonal matrix M which consists of matrices $M(k)$ as the entries on the diagonal. Applying the rule given by eqn. (6) it is easy to notice that (index k in local time derivative has been omitted)

$$\frac{\mathrm{D}M}{\mathrm{D}t} = \frac{\mathrm{d}M}{\mathrm{d}t} + [\Omega, M] = [\Omega, M] \tag{9}$$

where Ω is the block diagonal matrix with $\Omega(k)$ as entries on the diagonal. In general, for any block diagonal matrix B consisting of spatial matrices one can write

$$\frac{\mathrm{D}B}{\mathrm{D}t} = \frac{\mathrm{d}B}{\mathrm{d}t} + [\Omega, B] \tag{10}$$

In the last expression, $\frac{\mathrm{d}B}{\mathrm{d}t}$ denotes the local time derivative of each entry of the matrix calculated with respect to the local coordinate frame attached to the k-th body.

When a block diagonal matrix contains the spatial vectors as entries, the following rule, which is a generalization of eqn. (1), applies:

$$\frac{\mathrm{D}G}{\mathrm{D}t} = \frac{\mathrm{d}G}{\mathrm{d}t} + \Omega G \tag{11}$$

where G is an arbitrary block diagonal matrix with spatial vectors as entries on the diagonal. Here G is the block Kalman gain operator (Rodriguez and Kreutz, 1988).

Based on the above considerations, the most important time derivative which we are going to calculate is the time derivative of the articulated body inertia $P(k)$, which can be expressed in terms of $P(k-1)$ and $M(k)$ as follows (Rodriguez, 1987):

$$P(k) = \Phi(k,k-1)\bar{\tau}(k-1)P(k-1)\bar{\tau}^*(k-1)\Phi^*(k,k-1) + M(k) \tag{12}$$

Notice that the last equation is written in coordinate-free notation, namely matrices $A(k-1)$ and $A^{-1}(k-1)$ are not present. Recall also that $\bar{\tau}(k-1) = I - G(k-1)H(k-1)$. To calculate the time derivative of eqn. (12) in the k-th coordinate frame, first notice that $M(k)$ and $\Phi(k,k-1)$ are constant, as expressed in the k-th coordinate frame, and therefore their derivatives are zero. Another observation we have to make here is that $\frac{\mathrm{d}^{k-1}}{\mathrm{d}t}\left[\bar{\tau}(k-1)\right]P(k-1)\bar{\tau}^*(k-1) = \bar{\tau}(k-1)P(k-1)\frac{\mathrm{d}^{k-1}}{\mathrm{d}t}\left[\bar{\tau}^*(k-1)\right] = 0$. Note also that $\Psi(k,k-1) = \Phi(k,k-1)\bar{\tau}(k-1)$. Now, taking time derivatives of both sides of eqn. (12) with respect to time in the k-th coordinate one can get

$$\begin{aligned}\frac{\mathrm{d}^k}{\mathrm{d}t}P(k) &= \Psi(k,k-1)\frac{\mathrm{d}^{k-1}}{\mathrm{d}t}\Big[P(k-1)\Big]\Psi^*(k,k-1)\\ &\quad +\Psi(k,k-1)\Omega_\delta(k-1)P(k-1)\Psi^*(k,k-1)\\ &\quad -\Psi(k,k-1)P(k-1)\Omega_\delta(k-1)\Psi^*(k,k-1)\end{aligned} \tag{13}$$

In the last equation we have used $\bar{\tau}(k-1)\Omega_\delta(k-1) = \Omega_\delta(k-1)$. From eqn. (13) it is clear that the time derivative of $P(k)$ is expressed as the time derivative of $P(k-1)$ calculated in $(k-1)$-th coordinate system. Notice also that the second and third terms result from applying eqn. (7).

Here we make two comments which will be very useful is subsequent sections. Notice that the result given by eqn. (13) would be the same if we substituted $\bar{\tau}(k-1)P(k-1)\bar{\tau}^*(k-1) = \bar{\tau}(k-1)P(k-1) = P(k-1)\bar{\tau}^*(k-1)$ in eqn. (12) and calculated its time derivative. Remember that in calculating the articulated body inertia the above substitution can be applied (Rodriguez, 1987). It cannot be implemented in calculating the time derivative. This has a straightforward consequence in the computational load of eqn. (13), which is more intensive in comparison with eqn. (12) and cannot be easily simplified. Suppose now that the right-hand side of eqn. (12) is expressed in the k-th coordinate system. Then we have to include in eqn. (12) the spatial orientation matrix between the $(k-1)$-th and k-th coordinate frames. The best way to do it is to "stick" the matrix $A(k-1)$ to the link operator $\Phi(k,k-1)$. Then one can get

$$\frac{\mathrm{d}^k}{\mathrm{d}t}\Phi(k,k-1)A(k-1) = \Phi(k,k-1)A(k-1)\Omega_\delta(k-1) \tag{14}$$

Consequently, using eqn. (7) to calculate the time derivative of $\bar{r}(k-1)P(k-1)\bar{r}^*(k-1)$ in the $(k-1)$-th coordinate frame we do not have to calculate the expression which contains the $\Omega_\delta(k-1)$ operator since it is present in eqn. (14). Obviously, the results are the same. "Sticking" the cosine matrix $A(k-1)$ to the link operator $\Phi(k,k-1)$ is also useful for any equation in the operator form at the manipulator level.

Now we write the closed-form solution of eqn. (13). Integrating eqn. (13) from $i=0$ to $i=k-1$ with the initial condition $\frac{\mathrm{d}^0}{\mathrm{d}t}P(0)=0$ one can get

$$\frac{\mathrm{d}^k}{\mathrm{d}t}P(k) = \sum_{i=1}^{k-1}\Psi(k,i)\Big(\big[\Omega_\delta(i),P(i)\big]\Big)\Psi^*(k,i) \tag{15}$$

From eqn. (15) and from the structure of the articulated body transformations Ψ and $\mathcal{E}_\Psi$ we can write the closed-form solution

$$\frac{\mathrm{d}}{\mathrm{d}t}P = \dot{P} = \mathcal{E}_\Psi\Big(P + [\Omega_\delta, P]\Big)\mathcal{E}_\Psi^* \tag{16}$$

or

$$\dot{P} = \mathrm{diag}\left[\tilde{\Psi}\Big([\Omega_\delta, P]\Big)\tilde{\Psi}^*\right] \tag{17}$$

where $\tilde{\Psi} = \Psi - I$. From eqn. (17) it is clear that $[\Omega_\delta, P]$ is a driving term for calculating the local time derivative.

Suppose now that we want to calculate the time derivative of $P(k)$ with respect to the next coordinate frame (inwardly). Then eqns. (16) and (17) have to be rewritten as follows:

$$\dot{P} = \mathcal{E}_\Psi\Big(\dot{P} + [\Omega_\delta, P]\Big)\mathcal{E}_\Psi^* + [\Omega_\delta, P] \tag{18}$$

and

$$\dot{P} = \mathrm{diag}\Big[\Psi\Big([\Omega_\delta, P]\Big)\Psi^*\Big] \tag{19}$$

Introducing a new notation,

$$\dot{\lambda} = \dot{P} + [\Omega_\delta, P] \tag{20}$$

eqn. (18) can be written as

$$\dot{\lambda} = \mathcal{E}_\Psi\dot{\lambda}\mathcal{E}_\Psi^* + [\Omega_\delta, P] \tag{21}$$

Consequently, the upper summation index in eqn. (15) is equal to k. The local and global time derivatives of the spatial operators are summarized in Table 1.

In the next two sections we show how to implement the results collected in Tab. **1**. The results summarized in Tab. 1 were originally presented by Jain and Rodriguez (1994). In this paper, we derived them independently by making use of the considerations presented at the beginning of this section.

Tab. 1. Local and inertial time derivatives.

	Operator	Local derivative	Inertial derivative
1	$\mathcal{E}_\Phi$	$\dot{\mathcal{E}}_\Phi = \mathcal{E}_\Phi \Omega_\delta \Phi$	$\dot{\mathcal{E}}_\Phi + [\Omega, \mathcal{E}_\Phi]$
2	$\Phi = (I - \mathcal{E}_\Phi)^{-1}$	$\dot{\Phi} = \Phi \mathcal{E}_\Phi \Omega_\delta \Phi$	$\dot{\Phi} + \Phi([\Omega, \mathcal{E}_\Phi])\Phi$
3	H	$\dot{H} = 0$	$-H\Omega$
4	M	$\dot{M} = 0$	$[\Omega, M]$
5	$D = HPH^*$	$\dot{D} = H\dot{P}H^*$	$H\dot{P}H^*$
6	G	$\dot{G} = \bar{\tau}\dot{P}H^* D^{-1}$	$\dot{G} + \Omega G$
7	$\bar{\tau} = (I - GH)$	$\dot{\bar{\tau}} = -\dot{G}H$	$\dot{\bar{\tau}} + [\Omega, \bar{\tau}]$
8	$\mathcal{E}_\Psi = \mathcal{E}_\Phi \bar{\tau}$	$\dot{\mathcal{E}}_\Psi = \mathcal{E}_\Psi(-\dot{P}H^* D^{-1} H + \Omega_\delta \bar{\tau})$	$\dot{\mathcal{E}}_\Psi + [\Omega, \mathcal{E}_\Psi]$

2.1. Local and Global Time Derivatives of the Mass Factor

One of the advantages of using the operators in the equations of motion is that they are very concise and easy to interpret. For example, the mass matrix of the manipulator can be written in the following form (Rodriguez and Kreutz, 1988):

$$\mathcal{M} = H\Phi M \Phi^* H^* \tag{22}$$

$\mathcal{M}$ is positive definite and any factorization of it can be implemented. Rodriguez and Kreutz (1988) showed that the following factorization, which has strong relation to the filtering and smoothing Kalman algorithms, is true:

$$\mathcal{M} = (I + H\Phi K)D(I + H\Phi K)^* \tag{23}$$

$$(I + H\Phi K)^{-1} = (I - H\Psi K) \tag{24}$$

and

$$\mathcal{M}^{-1} = (I - H\Psi K)^* D^{-1}(I - H\Psi K) \tag{25}$$

where $K = \mathcal{E}_\Phi G$. From eqn. (23) it is clear that there exists the following factorization:

$$\mathcal{M} = m(\theta) m^*(\theta) \tag{26}$$

where $m(\theta) = (I + H\Phi K)D^{\frac{1}{2}}$.

The system kinetic energy $K(\theta,\dot{\theta}) = \frac{1}{2}\dot{\theta}^* \mathcal{M}\dot{\theta}$ results in

$$K(\theta,\dot{\theta}) = \frac{1}{2}\dot{\theta}^*(I + H\Phi K)D(I + H\Phi K)^*\dot{\theta} = \frac{1}{2}\sum_{k=1}^{N} \nu^2(k) \tag{27}$$

where $\nu = [\nu(1), \dots, \nu N]^*$ is a new set of variables related to the joint-angle rates (Jain and Rodriguez, 1994) by:

$$\nu = D^{\frac{1}{2}}(I + H\Phi K)^*\dot{\theta} \tag{28}$$

Jain and Rodriguez (1994) showed that the local and inertial time derivatives of the mass factor m are the same and are given by

$$\dot{m} = H\Phi\Big[\Omega_\delta \Phi P + \frac{1}{2}(I + \bar{\tau})\dot{P}\Big]H^* D^{-\frac{1}{2}} \tag{29}$$

This result can be easily verified by making use of the time derivatives of both columns from Tab. 1. Here we focus our attention on interpretation of this fact. $m(\theta)$ is the factor of the manipulator inertia matrix and has dimensions $N \times N$. Each element of this matrix is a scalar which depends on the generalized position vector θ. Time diferentiation of the matrix $m(\theta)$ means time differentiation of each component of this matrix, which gives the same result, regardless of the coordinate frame in which it is performed. On the contrary, when we have block diagonal matrix, then each entry of the matrix is a matrix itself and when differentiated, the rule of tensor differentiation has to be applied (compare the previous section). For this reason, local and inertial time derivatives are not the same.

Notice also that the mass factor derivative $\dot{m}$, expressed in terms of $\dot{\lambda}$, can be written as follows:

$$\dot{m} = H\Phi\left[\Omega_\delta \tilde{\Phi} P + \frac{1}{2}(I + \bar{\tau})\dot{\lambda}\right]H^* D^{-\frac{1}{2}} \tag{30}$$

If we multiply $\dot{m}$ by ν, we get

$$\dot{m}\nu = H\Phi\left[\Omega_\delta \Phi K H P + \frac{1}{2}\Big([\Omega_\delta, P]\Big) + \frac{1}{2}\Big([\mathcal{E}_\Psi, \dot{\lambda}]\Big)\right]V \tag{31}$$

Following the arguments described above, the components of the vector $\dot{m}\nu$ in local and inertial frames are the same.

Jain and Rodriguez (1994) showed that $\dot{\theta}\mathcal{M}_\theta\dot{\theta}$, where $\mathcal{M}_\theta$ denotes the derivative of $\mathcal{M}$ with respect to the vector of generalized positions, has the following form:

$$\dot{\theta}\mathcal{M}_\theta\dot{\theta} = 2H\Phi\Big[\Omega_\delta(I + \Phi K H)P - V \times M\Big]V \tag{32}$$

where $V\times$ can be viewied as a generalization of the traditional cross product in three dimensions to $6N$ dimensions. The vector $V\times$ is defined as the linear transformation $S[V] = \text{diag}[\tilde{V}(1), \dots, \tilde{V}(N)]$, where

$$\tilde{V}(k) = \begin{bmatrix} \tilde{\omega}(k) & \tilde{v}(k) \\ 0 & \tilde{\omega}(k) \end{bmatrix} \tag{33}$$

is the 6×6 matrix formed by the k-th link spatial velocity vector $V(k) = \mathrm{col}[\omega(k), v(k)]$. Interpretation of the terms involving $V\times$ will be discussed later.

2.2. Local and Global Time Differentiation of the Generalized Velocity Vector

In this subsection we present another application of the results contained in Tab. 1 which deals with time differentiation of a new velocity vector ξ.

In an alternative formulation of the diagonalized equations of motion Jain and Rodriguez (1994) considered the following velocity vector:

$$\xi = D^{-\frac{1}{2}}\nu = (I + H\Phi K)^{*}\dot{\theta} \tag{34}$$

We calculate the local time derivative of the above expression. As we already know, both local and inertial time derivatives are the same (recall that ξ is a vector consisting of N elements). Therefore, we get

$$\dot{\xi} = \frac{\mathrm{d}}{\mathrm{d}t}\xi = \frac{\mathrm{D}}{\mathrm{d}t}\xi = \frac{\mathrm{d}}{\mathrm{d}t}[I + H\Phi K]^{*}\dot{\theta} + (I + H\Phi K)\ddot{\theta} \tag{35}$$

Having in mind that $(I + H\Phi K) = I + H\Phi\mathcal{E}_{\Phi}G = H\Phi G$, we get

$$\dot{\xi} = D^{-1}HP\dot{\tau}^{*}V + D^{-1}HP\tilde{\Phi}^{*}\Omega_{\delta}^{*}V + G^{*}\Phi^{*}H^{*}\ddot{\theta} \tag{36}$$

Eqn. (36) describes the time derivative of the new velocity vector ξ, which is written in the operator form at the manipulator level. Obviously, one can write this equation at the link level, which is omitted here.

3. Standard Equations of Motion

Standard equations of motions for a manipulator with N degrees of freedom can be written as (Craig, 1986)

$$\tau = \mathcal{M}(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) \tag{37}$$

where $\theta, \dot{\theta}$, and $\ddot{\theta}$ are vectors of joint positions, velocities, and accelerations, respectively, τ is a vector of joint torques, $\mathcal{M}(\theta)$ is the mass matrix of the manipulator, and $C(\theta, \dot{\theta})$ is the matrix which consists of the centrifugal and Coriolis terms.

Equations of motion can be written in an alternative form as (Jain and Rodriguez, 1994)

$$\mathcal{M}\ddot{\theta} + H\Phi M\Phi^{*}a + H\Phi b = \tau \tag{38}$$

where $\mathcal{M} = H\Phi M\Phi^{*}H^{*}$ (for the definitions of the matrices see Nomenclature). In eqns. (37) and (38) friction torques have been neglected. Notice that eqn. (38) is written in a matrix (operator) form at the manipulator level.

In order to illustrate the above equations in the component form, we consider a manipulator with three degrees of freedom. Taking into account that

$\mathcal{M} = H\Phi M\Phi^* H^*$ and expanding eqn. (38) we can write the following expression for the actuating torque at the second joint, e.g. (links are numbered from the tip of the manipulator to its base)

$$\begin{aligned}\tau(2) &= H(2)\Phi(2,1)A(1)M(1)H^*(1)\ddot{\theta}(1) + H(2)r(2)H^*(2)\ddot{\theta}(2)\\ &+H(2)r(2)A^{-1}(2)\Phi^*(3,2)H^*(3)\ddot{\theta}(3) + H(2)\Phi(2,1)A(1)M(1)A^{-1}(1)n(1)\\ &+H(2)r(2)A^{-1}(2)n(2) + H(2)r(2)A^{-1}(2)\Phi^*(3,2)A^{-1}(3)n(3)\\ &+H(2)\Phi(2,1)A(1)b(1) + H(2)b(2)\end{aligned} \tag{39}$$

where $r(2)$ is the composite body inertia matrix at the second coordinate frame (for composite body inertia compare (Rodriguez and Kreutz, 1988)).

In the above equation we have included the direction cosine matrices $A(k)$ and $A(k)^{-1}$ (6 by 6) wherever necessary. Each term in eqn. (39) has a physical interpretation and can be written by hand without knowing the matrix form given by eqn. (38).

As an example take the component

$$H(2)\Phi(2,1)A(1)M(1)A^{-1}(1)n(1)$$

which appears in eqn. (39). The quantity $n(1)$, being the spatial bias acceleration, is expressed on the positive side of the first joint (cf. Kozłowski, 1992). Therefore, vector $A^{-1}(1)n(1)$ is expressed on the negative side of the first joint. The next vector, $M(1)A^{-1}(1)n(1)$ is expressed on the negative side of the first joint due to the fact that $M(1)$ is the inertia tensor on the negative side of the first joint. Left-hand side multiplication by the matrix $A(1)$ allows us to express the vector $A(1)M(1)A^{-1}(1)n(1)$ on the positive side of the first joint. Left-hand side multiplication by the matrix $\Phi(2,1)$ allows us to transform this vector to the second coordinate frame. Finally, this vector is projected on the axis of rotation of the second link in order to calculate its contribution to the second generalized force $\tau(2)$. In a similar manner, each term which appears in the equations of motion (for any torque) can be interpreted.

Notice that each equation has three components which result from the mass matrix components multiplied by corresponding accelerations. The simplest are the diagonal elements of the mass matrix which are the components of the composite body inertia matrix. In order to calculate their contribution to the individual joints we have to project them on both sides of $H(k)$ and its transpose $H^*(k)$. In each equation of motion there appears one diagonal element from the mass matrix of the manipulator. The other elements which are associated with the accelerations (notice that all of them appear in each equation for generalized force) are multiplied by the corresponding transpose of the joint axis projection vector and then transformed until the axis of rotation in which the forces are calculated is reached. Compare this rule with the definition of the matrix $\mathcal{M}$ and eqn. (38).

The spatial bias accelerations are transformed according to the same rule as the joint accelerations, but instead of vector $H^*(k)\ddot{\theta}(k)$, $A^{-1}(k)n(k)$ is used.

Finally, we have to discuss the contributions of the spatial bias forces which appear at each level of calculation. Notice that the k-th generalized force includes the contribution from the vectors $b(k)$ to $b(1)$. Each of these vectors has to be transformed until a corresponding axis of rotation, in which the force is calculated, is reached.

This mechanism is simple and allows one to write down the equations of motion in the component form. This is one of the advantages of the component formulation of the equations of motion for the manipulator with N degrees of freedom.

4. Diagonalized Langrangian Equations of Motion

In Section 2 we have introduced the new set of variables ν related to the joint-angle generalized velocities $\dot{\theta}$ given by eqn. (28). Notice that the new set of variables is expressed in terms of the articulated quantities such as D and P, and the spatial operator for all the links Φ. From this expression it is clear that by performing integration operation on both sides of eqn. (28) we do not obtain generalized positions θ. The componets of the vector ν are referred to as time derivatives of quasi-coordinates (Gutowski, 1971). The new set of velocities constitutes the system kinetic energy given by eqn. (27). Due to the factorization given by eqn. (26) differentiability of m ensures that the vector $\nu = m^*(\theta)\dot{\theta}$ is also differentiable. Invertibility of $m(\theta)$ (compare eqn. (25)) ensures that time derivatives $\dot{\theta}$ of the configuration variables can be recovered from ν. Under these conditions ν is a valid choice as a new generalized velocity vector. It was proved (Jain and Rodriguez, 1994) that the equations of motion can be written in the following form:

$$\dot{\nu} + C(\theta, \nu) = \varepsilon \tag{40}$$

with

$$C(\theta, \nu) = l\Big(\dot{m}\nu - \frac{1}{2}\dot{\theta}^* \mathcal{M}_\theta \dot{\theta}\Big), \qquad \varepsilon = l(\theta)\tau \tag{41}$$

where $l(\theta) = m^{-1}(\theta)$ and $\mathcal{M}_\theta$ denotes its derivative with respect to the vector of generalized coordinates. The set of equations given by eqn. (40) is known as the set of normalized equations of motion.

Recall that $\mathcal{M}$ is the positive definite matrix for which factorization given by eqn. (26) exists and therefore $l(\theta)$ exists. Equation (41) defines a new expression for the component $C(\theta, \nu)$, which depends on the generalized positions and the new velocity vector. The term $C(\theta, \nu)$ depends quadratically on the velocities ν_i. Notice also that there are some similarities between the expressions for $C(\theta, \nu)$ and $C(\theta, \dot{\theta})$. In both expressions the term $\frac{1}{2}\dot{\theta}^* \mathcal{M}_\theta \dot{\theta}$ appears, but in eqn. (41) it is multiplied by $m^{-1}(\theta)$. Jain and Rodriguez (1994) proved that the term $C(\theta, \nu)$ is calculated according to the following formula:

$$C(\theta, \nu) = \frac{1}{2} D^{-\frac{1}{2}} H \Psi \left([\mathcal{E}_\Psi, \dot{\lambda}] - \Omega_\delta P - P \Omega_\delta + 2V \times M \right) \Psi^* H^* D^{-\frac{1}{2}} \nu \tag{42}$$

An algorithm to compute $C(\theta,\nu)$ recursively is described below and is based on (Jain and Rodriguez, 1994).

$$\dot{\lambda}(0) = 0$$

for $k = 1, \ldots, N$

$$X(k) = \Omega_\delta(k)P(k)$$

$$\dot{\lambda}(k) = \Psi(k,k-1)\dot{\lambda}(k-1)\Psi^*(k,k-1) + X(k) + X^*(k)$$

$$y(k) = \Psi(k,k-1)y(k-1) + \left[2V(k)\times M(k) - X(k) + X^*(k)\right]V(k) + \Psi(k,k-1)\dot{\lambda}(k-1)V(k-1) - \dot{\lambda}(k)\Psi^*(k+1,k)V(k+1)$$

$$C(k) = \frac{1}{2}D^{-\frac{1}{2}}H(k)y(k)$$

end loop.

In Section 2 we have introduced another set of velocities defined by eqn. (34). By making use of these velocities the kinetic energy can be written as

$$K(\xi,\dot{\xi}) = \frac{1}{2}\dot{\xi}^* D(\theta)\dot{\xi} \tag{43}$$

Consequently, the equations of motion in the new coordinates (θ,ξ) are

$$D\dot{\xi} + C(\theta,\xi) = \kappa \tag{44}$$

where $\kappa = D^{\frac{1}{2}}\varepsilon = [I - H\Psi K]\tau$ and

$$C(\theta,\xi) = H\Psi\left[\dot{\lambda}H^*\xi - (\Omega_\delta P - V\times M)V\right] \tag{45}$$

An alternative set of equations given by eqns. (44) and (45) is known as the unnormalized diagonal equations. Here we make two comments. First notice that a term $(V\times M)V$ which appears in eqns. (42) and (45) represents the spatial bias forces vector b which is defined in Nomenclature. Since a gravity term can be included in the vector b (Kozłowski, 1993), both normalized and unnormalized equations can work in the gravity field. This extends the result originally described by Jain and Rodriguez (1994). Also notice that an alternative formulation for the matrix $C(\theta,\xi)$ can be written as

$$C(\theta,\xi) = D^{\frac{1}{2}}C(\theta,\nu) - D\frac{\mathrm{d}D^{-\frac{1}{2}}}{\mathrm{d}t}\nu \tag{46}$$

Note that $-D\frac{\mathrm{d}D^{-\frac{1}{2}}}{\mathrm{d}t}\nu = \frac{1}{2}H\dot{P}H^*D^{-\frac{1}{2}} = \frac{1}{2}H\dot{P}H^*\xi = \frac{1}{2}H\dot{\lambda}H^*\xi$, which means that ξ is present on the right-hand side of eqn. (46). It is clear from eqn. (46) that in this formulation matrix $C(\theta,\nu)$ is present and a new term $\frac{1}{2}H\dot{\lambda}H^*\xi$ is introduced which results only as the diagonal elements (the matrix $C(\theta,\nu)$ is skew-symmetric).

The elements of the matrix $C(\theta,\xi)$ can be calculated in the following recurrence (Jain and Rodriguez, 1994):

$$\dot{\lambda}(0) = 0$$

for $k = 1,\dots,N$

$$X(k) = \Omega_\delta(k)P(k)$$

$$\dot{\lambda}(k) = \Psi(k,k-1)\dot{\lambda}(k,k-1)\Psi^*(k,k-1) + X(k) + X^*(k)$$

$$\begin{aligned} y(k) = {} & \Psi(k,k-1)y(k-1) + \dot{\lambda}(k)H^*(k)\xi(k) \\ & + \Big[V(k)\times M(k) - X(k)\Big]V(k) \end{aligned}$$

$$C(k) = H(k)y(k)$$

end loop.

Here we make another comment. Substituting eqn. (36) in terms of the derivative with respect to the next coordinate frame and eqn. (41) into eqn. (44) we get, after tedious algebraic calculations, eqn. (38). In a similar fashion, starting from the normalized equations of motion given by eqn. (40) one can prove by performing time differentiation of ν and by rearranging terms that we get the standard equations of motion given by eqn. (38).

In the next section we focus our attention on a physical interpretation of each term of eqn. (42) and eqn. (45) at the component level.

5. Example of the Diagonalized Equations of Motion

In this section we emphasize a physical interpretation of each term of the matrix $C(\theta,\nu)$ at the component level in a similar fashion as in Section 3.

It has been observed that the matrix $C(\theta,\nu)$ is orthogonal to the new velocity vector ν. Therefore, we have

$$\nu^* C(\theta,\nu) = \frac{1}{2}V^*\Big([\mathcal{E}_\Psi,\dot{\lambda}] - \dot{\lambda}\varepsilon^*_\Psi - \Omega_\delta P - P\Omega_\delta\Big)V = 0 \tag{47}$$

From the last expression it is clear that the matrix in the middle is skew-symmetric. Notice that the matrix $C(\theta,\nu)$ has on both sides of the squared brackets elements $D^{-\frac{1}{2}}H\Psi$ and $\Psi^*H^*D^{-\frac{1}{2}}$; therefore, we rewrite the matrix $C(\theta,\nu)$ in the component form assuming that it is not multiplied by the velocity vector ν.

We do not consider the term $(V\times M)V$, because it represents the spatial bias forces vector b from the standard formulation transformed by the articulated quantities. Taking into account the assumptions described above, we have observed that a part of the matrix $C(\theta, \nu)$ (without velocity vector ν) can be considerably simplified due to cancellation of terms when written in the component form. After some algebraic manipulation the elements c_{ki} can be written in the following form:

$$c_{ki} = D^{-\frac{1}{2}}(k) \left[\sum_{m=1}^{i-1} H(k)\Psi(k,m)P(m)\Omega_\delta(m)\Psi^*(i,m)H^*(i) \right] D^{-\frac{1}{2}}(i) \quad (48)$$

and $c_{ki} = -c_{ik}$.

For the unnormalized equations of motion the off-diagonal elements are given by eqn. (48) and for the diagonal elements the upper summation index is i (compare with eqn. (45) in which, as before, the spatial bias forces vector has been omitted). For a better understanding of how to construct the matrix $C(\theta, \xi)$ consider the last row of this matrix and the corresponding Fig. 1 for $N = 5$.

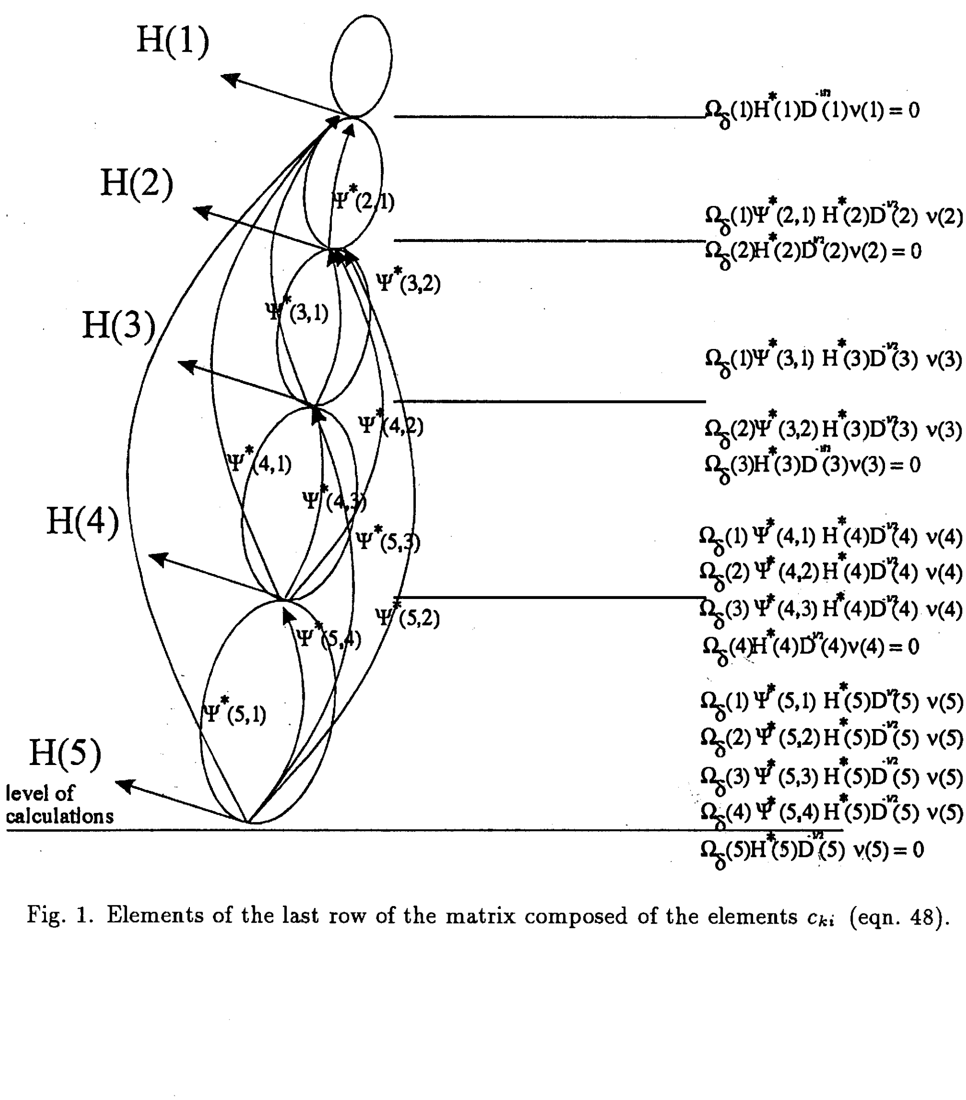

Fig. 1. Elements of the last row of the matrix composed of the elements c_{ki} (eqn. 48).

Recall that the operator expression for the velocity V in terms of the articulated quantities has the following form:

$$V(i) = \sum_{k=i}^{5} \Psi^*(k,i) H^*(k) D^{-\frac{1}{2}}(k) \nu(k) \tag{49}$$

On the right-side of Fig. 1 all the components of the velocities $V(i)$ for $i = 1, 2, \ldots, 5$ are shown, multiplied by the operators $\Omega_\delta(i)$. The new velocities $\nu(k)$ or $\xi(k)$, which can be considered as velocities of each joint, are first projected on the operator $H^*(k)$ and then transformed by the matrix $\Psi^*(k,i)$. The elements $D^{-\frac{1}{2}}(k)$ can be considered as scaling factors at each joint. Taking into account the expression given by eqn. (49) one can conclude that the last row of the particular matrix $C(\theta, \xi)$ shows how the elements of the velocity in terms of the articulated quantities are "distributed" through the links. Notice that $-\Omega_\delta(k)V(k) = V(k)\Omega_\delta^*(k)$, which corresponds to the spatial bias acceleration for the k-th joint expressed in terms of the articulated quantities. Next, these bias accelerations are multiplied by the articulated mass matrix $P(k)$ and transformed by the sequence of the matrices $D^{-\frac{1}{2}} H \Psi$. To understand this, take the standard equations of motion given by eqn. (38) and observe the term $H \Phi M \Phi^* a$, which transforms the bias spatial accelerations by the sequence of the matrices $H \Phi M \Phi^*$. From a detailed analysis one can conclude that the spatial bias accelerations for the diagonalized equations of motion are transformed in a similar manner as in the standard equations of motion. The only difference is that in the diagonalized equations of motion the articulated spatial bias accelerations are transformed by the sequence of operators $H\Psi$ instead of being transformed by the sequence of operators $H\Phi$. This result is not surprising and shows a similarity between the standard formulation and the new diagonalized equations of motion.

From eqn. (42) it is clear that the spatial bias forces are transformed by the sequence of operators $D^{-\frac{1}{2}} H \Psi$, which resembles the transformation of the vector b by the sequence of operators $H\Phi$ in the standard equations of motion.

6. Computational Complexity of the Related Algorithms

In this section we consider the scalar operations (multiplications, additions, subtractions and divisions, FLOPS but not per second) required to implement the algorithms presented in the previous sections. The calculation of the number of arithmetic operations is based on several assumptions. We assume that the link-to-link coordinate orientation transformation is the modified Denavit-Hartenberg orientation matrix. Generally, in calculations of the FLOPS we have considered two situations. In the first one, we have assumed that in the transition matrix, $\Phi(k, k-1)$, the skew symmetric part is full and the orientation cosine matrix is assigned to be arbitrary (according to the modified Denavit–Hartenberg notation). In the other case, it is assumed that the movement in the transition matrix $\Phi(k, k-1)$ is only in one direction, along x-axis, and the direction cosine matrix has been restricted to the twist angle α_i: $0°$, $90°$, and $-90°$.

We have implemented the concept of customizing the dynamic equations to reduce the computational requirements (Khosla, 1986). According to this convention, the non-zero elements of a vector or matrix are denoted by subscript variables, and the zero and unity elements by 0 and 1, respectively. We propagate the non-zero elements as variables and the zero elements as zeros. The customization procedure guarantees that every two mathematically equivalent expressions are denoted by the same variable name. Using the customization procedure results in longer but faster computer programs. The direction cosine matrix can be split up into two planar rotation matrices (Brandl *et al.*, 1986). If we realize that every matrix has an invariant part with respect to planar rotations, we shall see that we significantly reduce the number of operations.

A detailed analysis of the inverse dynamic problem presented in Section 3 has been reported by Kozłowski (1993). Here we only recall the main results. An N degrees-of-freedom general purpose manipulator with rotational joints only (which is the worst case) requires $142N - 161$ multiplications and $109N - 135$ additions/subtractions ($N > 2$). The computational requirements of the general purpose implementation incorporates the savings obtained by zero elements of the orientation matrices, $A(k)$, and the sparse $H(k)$ vector, the zero initial conditions with regard to the force and torque acting at the mass centre of each link, and the gravitational acceleration of the manipulator base.

Most of the existing manipulators have adjacent axes which are either parallel or perpendicular (the second assumption discussed above). Thus, for an N degrees-of-freedom manipulator the computational load is $104N - 117$ multiplications and $87N - 102$ additions/subtraction ($N > 2$). Here we make one comment; the calculation of the two-point boundary-value problem, which essentially is solved by the Kalman filtering and smoothing algorithms, seems to be slightly slower due to the fact that we recognize spatial quantities on both sides of each joint. The fastest algorithm presented by Khosla (1986) does not solve the two-point boundary-value problem and therefore is slighty faster.

Now we extend the analysis of the computational complexity to the diagonalized equations of motion presented in Section 4. The analysis is under the same conditions as presented at the begining of the section. Both normalized and unnormalized equations are considered. This kind of analysis has not been reported in the literature on robotics.

First, we write recursions which transform the generalized velocities to a new set of velocities ν. These are summarized in the following algorithms:

$\nu = D^{\frac{1}{2}}[I + H\Phi K]^*\dot{\theta}$	$\dot{\theta} = l^*\nu = [I - H\Psi K]D^{-\frac{1}{2}}\nu$
initial condition $V(N+1) = 0$	initial condition $V(N+1) = 0$
for $k = N, \ldots, 1$	for $k = N, \ldots, 1$
$V^+(k) = \Phi^*(k+1,k)V(k+1)$	$V^+(k) = \Phi^*(k+1,k)V(k+1)$
$\nu(k) = D^{\frac{1}{2}}(k)\left[\dot{\theta}(k) + G^*(k)A^{-1}(k)V^+(k)\right]$	$\dot{\theta}(k) = D^{-\frac{1}{2}}(k)\nu(k) - G^*(k)A^{-1}(k)V^+(k)$
$V(k) = A^{-1}(k)V^+(k) + H^*(k)\dot{\theta}(k)$	$V(k) = A^{-1}(k)V^+(k) + H^*(k)\dot{\theta}(k)$
end loop	end loop

The generalized forces τ and the new forces ε can be computed from each other as follows:

$\varepsilon = l\tau = D^{-\frac{1}{2}}[I - H\Psi K]\tau$

initial condition $z^+(0) = 0$
for $k = 1, \dots, N$
$z(k) = \Phi(k, k-1)z^+(k-1)$
$\varepsilon(k) = D^{-\frac{1}{2}}(k)\left[\tau(k) - H(k)z(k)\right]$
$z^+(k) = A(k)\left[z(k) + G(k)D^{\frac{1}{2}}(k)\varepsilon(k)\right]$
end loop

$\tau = m\varepsilon = [I + H\Phi K]D^{\frac{1}{2}}\varepsilon$

initial condition $z^+(0) = 0$
for $k = 1, \dots, N$
$z(k) = \Phi(k, k-1)z^+(k-1)$
$\tau(k)\big] = D^{\frac{1}{2}}(k)\varepsilon(k) + H(k)z(k)$
$z^+(k) = A(k)\left[z(k) + G(k)D^{\frac{1}{2}}(k)\varepsilon(k)\right]$
end loop.

In the above equations the transformation matrix $A(k)$ and its inverse $A(k)^{-1}$ are incorporated whenever it is necessary. First two algorithms are calculated outwardly, namely from the base of the manipulator to its tip, whereas the other two are calculated inwardly, in the opposite direction. This is consistent with transformation of the velocities and forces through the links of the manipulator. In Section 4 it has been shown how to calculate rows of the matrix $C(\theta, \nu)$ in a recursive form. It involves the recursion for the time derivative of the articulated spatial matrix $\lambda(k)$, compare eqn. (21). The recursion is similar to that for the Riccati equation. Computational complexity is similar to those equations for the articulated spatial inertia; it requires slighty more computations. This is due to the fact that the time derivative of the inertia update cannot be simplified (compare Section 2). In the expression for the vector $y(k)$ there are also terms which are sequences of matrices multiplied by the spatial vector. In such a situation we suggest to multiply the vector by the adjacent matrix, which results in another vector, multiplied in the sequel by the next matrix. Even if the matrices are the same, they can be stored and called whenever the multiplication is performed, although this leads to more calculations.

With the above assumptions, the resulting numbers of operations for the normalized and unnormalized equations of motion are presented in Tab. 2 ($N > 2$).

In Tab. 2 a row with the label "subtotal" indicates the number of operations which are common for both normalized and unnormalized equations of motion. Both algorithms are of complexity $O(N)$. The left-hand side of each column (indicated by 1) gives the number of operations for the general-purpose manipulator and the right-hand column (indicated by 2) numbers are for the manipulators which have axes either paraller or perpendicular and have displacement along the direction of the x-axis in the matrix $\Phi(k, k-1)$. In the first case the total number of arithmetic operations (FLOPS) is $1135N - 910$ and for the other case $782N - 740$. In Tab. 2 we do not include the calculation required to evaluate the sines and cosines. The above results have been obtained starting from the recurrence form for calculating the elements of the matrix $C(\theta, \nu)$ or $C(\theta, \xi)$. The component form of these matrices (discussed in Section 5) is less efficient due to the fact that each element of the corresponding matrix is calculated separately. In constructing Tab. 2 we have taken

Tab. 2. Number of arithmetic operations for an N-link manipulator required by the normalized and unnormalized equations of motion.

Recursions	Multiplications/divisions		Additions/subtractions	
	1	2	1	2
P	$101N-108$	$51N-58$	$94N-105$	$53N-64$
G	$5N-1$	$5N-1$	0	0
V^+	$6N-12$	$2N-4$	$6N-12$	$2N-4$
V	$16N-29$	$8N-16$	$9N-17$	$5N-9$
z	$6N-12$	$2N-4$	$6N-12$	$2N-4$
ψ	$44N-1$	$12N-1$	$19N-1$	$5N-1$
b	$59N-43$	$55N-53$	$40N-27$	$38N-35$
X	$18N-12$	$18N-12$	0	0
$\dot{P}$	$159N-6$	$95N-6$	$171N-6$	$113N-6$
Subtotal	$414N-224$	$248N-155$	$354N-180$	$218N-123$
$\dot{\theta}$	$7N-10$	$7N-11$	$6N-10$	$6N-11$
$\in$	N	N	$N-1$	$N-1$
z^+	$23N-4$	$14N-3$	$15N-8$	$10N-7$
y	$158N-223$	$134N-211$	$162N-249$	$140N-217$
$C(\theta,\nu)$	$2N-1$	$2N-1$	0	0
$\dot{\nu}$	0	0	N	N
	$191N-238$	$158N-226$	$185N-268$	$158N-236$
Normalized equations of motion	$605N-462$	$406N-381$	$530N-448$	$376N-359$
$\dot{\theta}$	$5N-8$	$5N-9$	$5N-8$	$5N-9$
κ	0	0	$N-1$	$N-1$
z^+	$21N-4$	$43N-3$	$13N-8$	$9N-7$
y	$59N-83$	$47N-71$	$61N-89$	$50N-78$
$C(\theta,\xi)$	0	0	0	0
$\dot{\xi}$	N	N	N	N
	$86N-95$	$66N-83$	$81N-106$	$66N-95$
Unnormalized equations of motion	$500N-319$	$314N-238$	$426N-286$	$284N-218$

into account the initial conditions for the various recursions in order to minimize the number of FLOPS. These conditions are concerned both with the base of the manipulator and its tip. In many recursions it was possible to take into account the next link adjacent to the initial link, which leads to $N>2$ as a condition for Tab. 2.

It is also of interest to compare the above numbers of operations for other forward dynamics algorithms. These results are summarized in Tab. 3. From Tab. 3 it is clear that the most efficient algorithms are those derived from the Kalman filtering and smoothing (Kozłowski, 1993) and one introduced by Brandl *et al.* (1986). In order to compare the results we have calculated FLOPS for the case $N = 6$ and $N = 12$. For a general-purpose manipulator with six degrees of freedom and for the normalized equations of motion we have 5900 FLOPS. This result is better in comparison with two of the methods presented by Walker and Orin (1982), namely the first and the fourth one, which result in 5920 FLOPS and 5967 FLOPS, respectively. One can conclude that the diagonalized equations of motions are comparable to the methods presented by Walker and Orin (1982).

Tab. 3. Number of operations for different forward dynamics algorithms.

Method	Computational complexity	N	
		N=6	N=12
Articulated body (Featherstone, 1987)	$579N - 526$	2948	6422
1-st Walker and Orin (1982)	$\frac{1}{3}N^3 + 130\frac{1}{2}N^2 + 197\frac{1}{6}N - 33$	5920	21701
2-nd Walker and Orin (1982)	$\frac{1}{3}N^3 + 66\frac{1}{2}N^2 + 261\frac{1}{6}N - 33$	4000	13253
3-rd Walker and Orin (1982)	$\frac{1}{6}N^3 + 21\frac{1}{2}N^2 + 358\frac{1}{6}N - 113$	2882	7857
4-th Walker and Orin (1982)	$132\frac{1}{2}N^2 + 207N - 31$	5967	21537
Brandl *et al.* (1986)	$470N - 420$	2400	5220
Numerical solution of the dyn. eqs. (Kozłowski, 1992)	$\frac{1}{3}N^3 + 20\frac{1}{2}N^2 + 341\frac{1}{6}N - 395$	2462	7227
Kalman filtering and smoothing (Kozłowski,1993)	$477N - 503$	2359	5221
Normalized equations of motion	$1135N - 910$	5900	12710
Unnormalized equations of motion	$926N - 605$	4951	10507

Finally we make one comment. In all the algorithms presented in Tab. 3 the number of operations required to perform integration in the forward dynamics problem has not been included.

7. Concluding Remarks

In this paper, we have reviewed the diagonalized equations of motion. We have examined the differentiation operation of such quantities as scalars, spatial vector, and matrices in different coordinate frames. It has been explained why local and global time derivatives of certain matrices are the same. We have discussed the original derivation given by Jain and Rodriguez (1994) and made several comments which give better understanding of the diagonalized equations of motion. In particular, we have discussed the time differentiation of different quantities at each stage of derivation.

Apart from that, we have investigated both the matrices $C(\theta, \nu)$ and $C(\theta, \xi)$ in the component form. An exact closed-form solution for each component has been given with its physical interpretation. It allows the reader to write the equations both in normalized and unnormalized forms. All terms which appear in the final equations of motion have been explained in both graphical and mathematical forms. This provides more insight in the new equations of motion. The same analysis has been done for the standard equations of motion. It has been shown that there exists a relevance between both formulations. This relevance is through the spatial bias acceleration vector and the spatial bias forces vector. These vectors transform in the new formulation through the sequence of operators $H\Psi$, while in the standard formulation through the sequence of operators $H\Phi$. The operator Ψ is associated with the articulated quantities, while the operator Φ is associated with the composite body quantities.

Two examples for the standard and diagonalized equations of motion have been considered. These examples have been completely discussed and give better understanding and more insight into the new algorithms. Some numerical considerations are discussed in (Kozłowski, 1994).

Computational complexity of the related algorithms analysis has been presented. Both standard and new algorithms have been considered. This analysis is relatively wide and relates the results to those existing in the literature on robotics. Much work is required for the analysis of the values $D(k)$ and how to use them for the purpose of control. A more comprehensive application of diagonalized models in robot control will require further investigation. Identification of the values $D(k)$ is another problem which is also of interest and has not been solved so far. Some numerical considerations on the diagonalized equations of motion and their sensitivity to the Riccati equation (which does not change from one step to the next) would be very interesting. These problems require more investigations and will be reported in subsequent papers.

References

Arnold V.I. (1978): *Mathematical Methods of Classical Mechanics.* — Berlin: Springer-Verlag.

Bedrossian N. (1992): *Linearizing coordinate transformations and Riemann curvature.* — Proc. 31st IEEE Conf. *Decision and Control*, Tusson, AZ, USA pp.80–85.

Brandl H., Johanni R. and Otter M. (1986): *A very efficient algorithm for the simulation of robots and similar multibody systems without inversion of the mass matrix.* — Proc. IFAC/IFIP/IMACS Int. Symp. *Theory of Robots*, Wienna, Austria, pp.365–370.

Craig J.J. (1986): *Introduction to Robotics. Mechanics and Control.* — New York: Addison-Wesley Publishing Company, Inc.

Featherstone R. (1987): *Robot Dynamics Algorithms.* — Lancaster: Boston/Dordrecht Kluwer Academic Publishers.

Gutowski R. (1971): *Analytical Mechanics.* — Warsaw: Polish Scientific Publishers (in Polish).

Jain A. and Rodriguez G. (1994): *Diagonalized Lagrangian robot dynamics.* — Proc. IEEE Int. Conf. *Robotics and Automation*, San Diego, USA, v.1, pp.334–339.

Khosla P.K. (1986): *Real–Time Control and Identification of Direct–Drive Manipulators.* — Ph.D. Thesis, Carnegie–Mellon University.

Koditschek D. (1985): *Robot kinematics and coordinate transformations.* — Proc. of the 24-th IEEE Conf. *Decision and Control*, Ft. Landerdale, Florida, USA, pp.1–4.

Kozłowski K. (1992): *Mathematical Dynamic Robot Models and Identification of their Parameters.* — Technical University of Poznań Press, (in Polish).

Kozłowski K. (1993): *Computational requirements for a discrete Kalman filter in robot dynamics algorithms.* — Robotica, v.11, pp.27– 36.

Kozłowski K. (1994): *Computational requirements for a discrete Kalman filter in diagonalized robot dynamics.* — Proc. 1st Int. Symp. *Mathematical Models in Automation and Robotics*, Międzyzdroje, Poland, pp.392–397.

Levi–Civita T. (1950): *The Absolute Differential Calculus.* — London and Glasgow: Blackie and Son Ltd.

Rodriguez G. (1987): *Kalman filtering, smoothing and recursive robot arm forward and inverse dynamics.* — IEEE J. Robotics and Automation, v.3, No.6, pp.624–639.

Rodriguez G. and Kreutz K. (1988): *Recursive Mass Matrix Factorization and Inversion, An Operator Approach to Open - and Closed - Chain Multibody Dynamics.* — JPL Publication 88-11. National Aeronautics and Space Administration, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, USA.

Spong H. (1992): *Remarks on robot dynamics: canonical transformations and Riemannian geometry.* — Proc. 1992 IEEE Conf. *Robotics and Automation*, Nice, France, pp.554–559.

Walker M.W. and Orin D.E. (1982): *Efficient dynamics computer simulation of robotics mechanisms.* — J. Dynamics Systems, Measurement and Control, v.104, pp.205–211.

Wittenburg J. (1977): *Dynamics of Systems of Rigid Bodies.* — Stuttgart: R.G. Teuber.

Received: December 20, 1994
Revised: February 15, 1995