Robotica (2009) volume 27, pp. 689–700. © 2008 Cambridge University Press
doi:10.1017/S0263574708005067 Printed in the United Kingdom

A new factorization of the Coriolis/centripetal matrix

Juan Ignacio Mulero-Martínez*

Departamento de Ingeniería de Sistemas y Automática, Universidad Politécnica de Cartagena, Campus Muralla del Mar, Cartagena 30203, España

(Received in Final Form: July 15, 2008. First published online: September 30, 2008)

SUMMARY

This paper provides a comprehensive description of a new method of factorization for the Coriolis/centripetal matrix. In the past three decades, studies on dynamics have rapidly developed through the efforts of many researchers in the field of mechanics. While direct methods for deriving the Coriolis/centripetal matrix are well known and have been widely used in the last century, the entries of this matrix were always obtained by means of the Christoffel symbols of first kind. Startling techniques for deriving dynamic equations of robot manipulators first appeared about 30 years ago. Since then, much has been done to refine and develop the method, but it is still a highly active field of research, with many outstanding problems, both theoretical and in applications. This work presents, in a unitary frame and from a new perspective, the main concepts and results of one of the most fascinating aspects of mechanics, namely the factorization of structures, and offers the reader another point of view concerning a possible way to approach the Coriolis/centripetal matrix. It aims to study a theory of representation for such a matrix based on an elegant method of fundamental matrices. The paper is intended to be self-contained by presenting complete properties emerging from these novel structures. This work is useful not only to researchers in mechanics, but also to control engineers who are interested in learning some of the mechanical modeling. Toward this end, the paper provides numerical examples, as well as practical adaptive applications for modern designers to use at the system level.

KEYWORDS: Coriolis/centripetal matrix; Factorization; Robot Dynamics.

1. Introduction

In the last decade, motion control of robot manipulators has become an active research field, leading to interesting results connected to the structural properties of mechanics. Two remarkable properties of the dynamics have been analyzed in the literature, turning out to be very useful for dynamic parameter identification as well as for deriving control laws, namely the linearity in the parameters and the passivity property of robot manipulators.[1] Actually, these two properties have been extensively applied in the adaptive control of robot manipulators. An interesting study of the properties of the terms arising in the standard model for the dynamics of an n-link rigid robot and their functional dependence can be found in refs. [2, 3]. Furthermore, the analysis of the structural properties of robot manipulators has been explored in the literature in terms of the Coriolis/centripetal matrix via several factorizations.[4–9] However, this does not mean that the topic is completely closed and that there are no other more appropriate formulations to represent this matrix.

The first contribution of this paper is the analysis of the structural properties of the Coriolis/centripetal matrix. To this purpose, two fundamental matrices are introduced in order to represent this matrix, namely the inertia derivative matrix and the inertia velocity matrix. These components exhibit interesting *properties of commutation and representation*, allowing one to describe the Coriolis/centripetal matrix in a useful way. Remarkably, in this paper a new factorization is proposed with the benefit of providing a compact description in matrix form for the Coriolis/centripetal effects as compared to other factorizations appearing in the literature up to now. As it is well known in mathematics, whenever a new structure is introduced, it is important to provide a description, as complete as possible, concerning the properties inferred from it. The classification of the *structural properties* is novel in terms of these matrices, which hereafter are referred to as *fundamental matrices*, since they have not been considered in the literature, and constitute a fundamental tool for the implementation of both controllers and nonlinear emulators. The breakthrough of this new approach stands in its applicability, since it is inherently based on a matrix formulation.[10–12] The main focus of this paper is to provide the reader with a deep understanding of modeling and design strategies based on the approximation of nonlinear terms. Moreover, to further link the theoretical aspects with practical issues, and to provide the reader with an idea of the advantages of this method, a numerical example and an adaptive application are included in the paper.

After an introduction covering background of the Coriolis/centripetal effects, the reader learns more about the factorizations in Section 2. Since this is an interesting research field, a comprehensive state-of-the-art overview is very valuable, introducing the main problems as well as the ways of solution that seem promising and emphasizing their basis, strengths, and weaknesses. In Section 3, fundamental matrices as well as their properties are described and analyzed in detail. The paper presents the theory behind Coriolis/centripetal effects together with the practical aspects related to put it to work on the real examples. Throughout all parts of Sections 4 and 5, the author has taken a practical and

*Corresponding author. E-mail: juan.mulero@upct.es

https://doi.org/10.1017/S0263574708005067 Published online by Cambridge University Press

application-driven point of view, providing a comprehensive and easily understood approach to all the concepts discussed. Specifically, Section 4 is devoted to putting forward a simple example which serves to illustrate the basic ideas. Section 5 provides a real-world application along with practical rules-of-thumb for modern designers to use at the system level as well as the architectural level. Finally, concluding remarks are discussed in Section 6.

2. Representations of the Coriolis/Centripetal Matrix: Factorizations

The equations of the dynamics of a robot manipulator involve nonlinear terms such as inertial, Coriolis/centripetal, frictional, and gravitational forces that depend on trigonometric functions in a complex way, see ref. [13]. The dynamics of an n-link robot manipulator may be derived from the Lagrange or Hamilton formulation, see refs. [5, 6], using the position of the joints as generalized coordinates. The standard form of the dynamics equations is as follows:

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

where $q(t), \dot{q}(t)$, and $\ddot{q}(t) \in \mathbb{R}^n$ represent the position, velocity, and acceleration of the joints, respectively and $M(q) \in \mathbb{R}^{n\times n}$ stands for the generalized inertia matrix, $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$ the Coriolis/centripetal matrix, $F(q,\dot{q}) \in \mathbb{R}^n$ the friction forces, $G(q) \in \mathbb{R}^n$ the gravitational vector, and $\tau(t) \in \mathbb{R}^n$ the joint torques. A deep analysis of the dynamics reveals the complexity of the inertial and Coriolis/centrifugal terms, see refs. [4, 5]. The Coriolis/centripetal forces $C(q,\dot{q})\dot{q}$ are always present in every articulated mechanism with revolute degrees of freedom. In this sense, centripetal forces appear when links or bodies are rotating with respect to reference systems whereas Coriolis forces do so when a link develops a motion with respect to a rotating reference system. Centripetal forces are proportional to the centripetal acceleration, directed to the center of the trajectory, with a magnitude directly proportional to the square of the generalized joint velocity, $\dot{q}_i^2$. In turn, Coriolis forces are proportional to the Coriolis acceleration $2(\omega \times v)$ and as a result to $\dot{q}_i\dot{q}_j$. Indeed, if r_{i,c_j} represents the position of the center of mass attached to the link j relative to the frame i, the acceleration of the center of mass of the jth link with respect to the base frame is

$$\ddot{r}_{0,c_j} = \omega_0^i \times \left(\omega_0^i \times r_{i,c_j}\right) + 2\left(\omega_0^i \times \dot{r}_{i,c_j}\right) + \dot{\omega}_0^i \times r_{i,c_j} + R_0^i \ddot{r}_{i,c_j} + \ddot{d}_0^i$$

where the term $\omega_0^i \times (\omega_0^i \times r_{i,c_j})$ is the centripetal acceleration caused by the revolute motion of the vector r_{i,c_j} around the frame i with an angular velocity ω_0^i. In contrast, the term $2(\omega_0^i \times \dot{r}_{i,c_j})$ is called the Coriolis acceleration and is originated when the motion of the center of mass of the link j is expressed regarded with the frame i which is rotating with an angular velocity ω_0^i. The last terms correspond to the transversal acceleration and the drag acceleration. The literature has presented some ways to derive the Coriolis/centripetal matrix from the generalized inertia matrix leading to two major factorizations of this matrix: (1) factorizations based on the Kronecker product and (2) factorizations using tensor analysis. The former were introduced by Lewis, see ref. [6], allowing to split up the Coriolis/centripetal matrix into a static component $C_p^T(q) = \left(\frac{\partial M(q)}{\partial q}\right)^T = \left[\frac{\partial M(q)}{\partial q_1}, \ldots, \frac{\partial M(q)}{\partial q_n}\right] \in \mathbb{R}^{n\times n^2}$ depending only on the position vector $q(t)$ and a dynamic term $C_{\mathrm{v}}(\dot{q}) = (\dot{q}^T \otimes I_n) - \frac{1}{2}(I_n \otimes \dot{q}^T) \in \mathbb{R}^{n\times n^2}$ related to the generalized velocity vector $\dot{q}(t)$

$$C(q,\dot{q}) = C_{\mathrm{v}}(\dot{q})C_p(q).$$

In the second class of factorizations, the definition of the Coriolis/centripetal forces from the Lagrangian or Hamiltonian formulation is considered

$$\begin{aligned} C(q,\dot{q})\dot{q} &= \dot{M}(q)\dot{q} - \frac{1}{2}\frac{\partial}{\partial q}(\dot{q}^T M(q)\dot{q}) \\ &= \dot{M}(q)\dot{q} - \frac{\partial K(q,\dot{q})}{\partial q} \end{aligned} \quad (2)$$

where $K(q,\dot{q})$ is the kinetic energy of the whole manipulator. As a result, the partial derivatives of the matrix $M(q)$ with respect to a vector $q(t)$ of dimension n is carried out, which causes the emergence of tensors of order three. A widely extended tensor representation is based on Christoffel symbols of the first kind, see refs. [5, 6], which are defined as

$$c_{ijk}(q) = \frac{1}{2}\left(\frac{\partial M_{ik}(q)}{\partial q_j} + \frac{\partial M_{jk}(q)}{\partial q_i} - \frac{\partial M_{ij}(q)}{\partial q_k}\right) \quad (3)$$

where $M_{ij}(q)$ represents the entry (i, j) of the generalized inertia matrix. In this sense, the element (k, j) from the Coriolis/centripetal matrix can be represented as

$$c_{kj}(q,\dot{q}) = \sum_{i=1}^{n} c_{ijk}(q)\dot{q}_i \quad (4)$$

and the component k of the generalized Coriolis/centripetal force is

$$e_k^T C(q,\dot{q})\dot{q} = \sum_{j=1}^{n}\sum_{i=1}^{n} c_{ijk}(q)\dot{q}_i\dot{q}_j \quad (5)$$

where the first indices of the Christoffel symbols are related to the velocities of the joints i and j that induce a generalized reaction torque over the joint k, see refs. [1, 8]. Therefore, $c_{iik}(q)\dot{q}_i^2$ stands for the centripetal effect induced over the joint k when the joint i turns at velocity $\dot{q}_i$. In a similar way, $c_{ijk}(q)\dot{q}_i\dot{q}_j$ represents the Coriolis effect induced over the joint k when the joints i and j turn at velocities $\dot{q}_i$ and $\dot{q}_j$, respectively. Using the definition of Christoffel symbol of the first kind, the element (k, j) of $C(q,\dot{q})$ can be written as

$$c_{kj}(q,\dot{q}) = \frac{1}{2}\left(\left(\frac{\partial M_k(q)}{\partial q_j}\right)^T\dot{q} + \dot{M}_{jk}(q) - \left(\frac{\partial M_j(q)}{\partial q_k}\right)^T\dot{q}\right) \quad (6)$$

where $M_j(q)$ represents the column j of the inertia matrix $M(q)$. Then, the column vectors of C are affected by the

https://doi.org/10.1017/S0263574708005067 Published online by Cambridge University Press

transformation

$$c_j(q) = \frac{1}{2}\left(\dot{M}_j(q) + \frac{\partial M(q)}{\partial q_j}\dot{q} - \left(\frac{\partial M_j(q)}{\partial q}\right)^T \dot{q}\right). \quad (7)$$

Therefore, the Coriolis/centripetal matrix can be expressed as the sum of three matrix terms

$$C(q,\dot{q}) = \frac{1}{2}\left(\dot{M}(q) + \sum_{j=1}^{n}\frac{\partial M(q)}{\partial q_j}\dot{q}e_j^T - \sum_{j=1}^{n}\left(\frac{\partial M_j}{\partial q}\right)^T \dot{q}e_j^T\right) \quad (8)$$

where $e_j \in \mathbb{R}^n$ is a canonical vector with an entry "1" at the jth position and the rest of the entries are "0." Another philosophy to express the Coriolis/centripetal matrix consists of separating the terms including the quadratic velocity $\dot{q}_k^2$ from those cross terms. Thus, the element (k, j) from this matrix can be expressed as

$$\begin{aligned} c_{kj}(q,\dot{q}) &= \sum_{i=1}^{n}\left(\frac{\partial M_{kj}(q)}{\partial q_i} - \frac{1}{2}\frac{\partial M_{ij}(q)}{\partial q_k}\right)\dot{q}_i \\ &= \dot{M}_{kj}(q) - \frac{1}{2}\left(\frac{\partial M_j(q)}{\partial q_k}\right)^T \dot{q}. \end{aligned} \quad (9)$$

Schilling has introduced a tensor $c_{ij}^k(q)$, see ref. [7], defined as

$$c_{ij}^k(q) = \frac{\partial M_{kj}(q)}{\partial q_i} - \frac{1}{2}\frac{\partial M_{ij}(q)}{\partial q_k} \quad \text{for } 1 \le i, j, k \le n$$

so that the Coriolis/centripetal force can be written as $C(q,\dot{q}) = \sum_{i=1}^{n}(\dot{q}^T C_i(q)\dot{q})e_i$ where the $n \times n$ matrix $C_k(q) = \sum_{j=1}^{n}\sum_{i=1}^{n} e_i c_{ij}^k(q) e_j^T$ is referred to the velocity coupling matrix for the joint k. Thus, the component i of the Coriolis/centripetal force can be split up into two terms

$$\dot{q}^T C_k(q)\dot{q} = \sum_{i=1}^{n} c_{ii}^k(q)\dot{q}_i^2 + \sum_{i=1}^{n}\sum_{j\neq i}^{n} c_{ij}^k(q)\dot{q}_i\dot{q}_j. \quad (10)$$

Hence, it is possible to interpret the terms of the last expression: the first term represents the centripetal force acting on the joint i due to the motion of the joint k while the second one is the Coriolis force acting on the joint i caused by the combined motion of the joints i and j where $i \neq j$. Craig has also developed the same factorization as in (10) separating the centripetal matrix $C_{\text{cen}}(q) \in \mathbb{R}^{n\times n}$ from the Coriolis matrix $C_{\text{cor}}(q) \in \mathbb{R}^{\frac{n(n-1)}{2}\times\frac{n(n-1)}{2}}$.

$$C(q,\dot{q}) = C_{\text{cor}}(q)\langle\dot{q},\dot{q}\rangle + C_{\text{cen}}(q)\langle\dot{q}^2\rangle$$

where $\langle\dot{q},\dot{q}\rangle = (\dot{q}_1\dot{q}_2, \dot{q}_1\dot{q}_3, \ldots, \dot{q}_{n-1}\dot{q}_n)^T \in \mathbb{R}^{\frac{n(n-1)}{2}}$ and $\langle\dot{q}^2\rangle = (\dot{q}_1^2, \dot{q}_2^2, \ldots, \dot{q}_n^2) \in R^{n\times 1}$.

As it is discussed earlier, the computations of the Coriolis/centripetal matrix depend on how the generalized inertia matrix is expressed. A method to obtain this inertia matrix consists of obtaining an explicit form for the kinetic energy $K(q,\dot{q})$ as the sum of a rotation energy $K_{\text{rot}}(q,\dot{q}) = \frac{1}{2}\sum_{i=1}^{n}\omega_i^T I_0^i(q)\omega_i$ and a translation energy $K_{\text{trans}}(q,\dot{q}) = \frac{1}{2}\sum_{i=1}^{n} m_i v_{c_i}^T v_{c_i}$. Then the velocities are transformed from the Cartesian space to the joint space using the Jacobian transformation $J_{v_{c_i}}$ and J_{w_i} and the inertia matrix of a link I_i^i is projected from the frame to which it is rigidly attached, to the base frame via a congruent transformation (this can be derived using the concept of the angular momentum, see ref. [14])

$$\begin{aligned} M(q) = \sum_{i=1}^{n} & J_{v_{c_i}}^T(q) m_i J_{v_{ci}}(q) \\ & + J_{w_i}^T(q) R_0^i(q) I_i^i \left(R_0^i(q)\right)^T J_{w_i}(q). \end{aligned} \quad (11)$$

However, the kinetic energy of the link i can also be written using the trace of matrices, see refs. [8, 9]

$$K_i(q,\dot{q}) = \frac{1}{2}\sum_{i=1}^{n}\sum_{p=1}^{i}\sum_{r=1}^{i}\text{Tr}\left(U_{ip}(q) J_i U_{ir}^T(q)\right)\dot{q}_p\dot{q}_r$$

where $J_i \in \mathbb{R}^{4\times 4}$ is the pseudo-inertia matrix of the link, and $U_{ij}(q) = \frac{\partial A_0^i(q)}{\partial q_j}$ is the variation of the matrix of rigid motion (homogenous transformation) from the frame i to the base frame. In this case, a new tensor $h_{kij}(q)$ emerges from this formulation

$$h_{kij}(q) = \sum_{m=\max(k,i,j)} \text{Tr}\left(U_{mij}(q) J_m U_{mk}^T(q)\right)$$

where $U_{ijk}(q) = \frac{\partial^2 A_0^i(q)}{\partial q_k \partial q_j}$. In this case, the component k of the generalized Coriolis/centripetal force is

$$e_k^T C(q,\dot{q})\dot{q} = \sum_{i=1}^{n}\sum_{j=1}^{n} h_{kij}(q)\dot{q}_i\dot{q}_j. \quad (12)$$

Comparing Eqs. (5) and (12), it is concluded that $h_{kij}(q) = c_{ijk}(q)$. Another factorization emerges if the column j of the Coriolis/centripetal matrix is calculated from the generalized inertia matrix using the expression of $c_{kj}(q)$ from (9)

$$c_j(q,\dot{q}) = \sum_{k=1}^{n} c_{kj}(q)e_k = \dot{M}_j(q) - \frac{1}{2}\left(\frac{\partial M_j(q)}{\partial q}\right)^T \dot{q} \quad (13)$$

where e_k is a unitary vector of dimension n with a value 1 at the position k. As a consequence, the Coriolis/centripetal matrix is expressed from the representation in column vectors $c_j(q,\dot{q})$ as defined in Eq. (13).

$$\begin{aligned} C(q,\dot{q}) &= \sum_{i=1}^{n} c_i(q,\dot{q})e_i^T \\ &= \dot{M}(q) - \frac{1}{2}\left(\left(\frac{\partial M_1(q)}{\partial q}\right)^T \dot{q}, \ldots, \left(\frac{\partial M_n(q)}{\partial q}\right)^T \dot{q}\right) \\ &= \dot{M}(q) - \frac{1}{2}\sum_{i=1}^{n}\left(\frac{\partial M_i(q)}{\partial q}\right)^T \dot{q}e_i^T. \end{aligned} \quad (14)$$

https://doi.org/10.1017/S0263574708005067 Published online by Cambridge University Press

The two classes of factorization presented earlier are connected so that the Coriolis/centripetal matrix is written in terms of the Kronecker product and a matrix $\bar{C}(q) = (C_1(q)^T, \ldots, C_n(q)^T)^T$,

$$C(q, \dot{q}) = (I_n \otimes \dot{q}^T)\bar{C}(q). \tag{15}$$

Equation (13) can be factorized using the Kronecker product, see ref. [6],

$$C(q, \dot{q}) = \left(U^T(q, \dot{q}) - \frac{1}{2}U(q, \dot{q})\right) \tag{16}$$

where $U(q, \dot{q}) = \left(\frac{\partial M(q)}{\partial q}\right)^T(\dot{q} \otimes I_n)$ or

$$C(q, \dot{q}) = \frac{1}{2}(\dot{M}(q) + S(q, \dot{q})) \tag{17}$$

where $S(q, \dot{q})$ is a skew-symmetric matrix defined as $S(q, \dot{q}) = U - U^T$. The last equation can be identified with (8). Furthermore, it is observed that $S(q, \dot{q}) = \sum_{j=1}^{n} \frac{\partial M(q)}{\partial q_j}\dot{q}e_j^T - \sum_{j=1}^{n} \frac{\partial M_j^T}{\partial q}\dot{q}e_j^T$ which implies that $U = \sum_{j=1}^{n} \frac{\partial M(q)}{\partial q_j}\dot{q}e_j^T$. Finally, Eqs. (2) and (14) can be written down using the matrix U,

$$C(q, \dot{q}) = \dot{M}(q) - \frac{1}{2}U(q, \dot{q}). \tag{18}$$

3. Fundamental Matrices

In this section some notions regarding the representation of the Coriolis/centripetal matrices are introduced. All the ideas presented here constitute an original contribution and have many interesting implications in the field of robotics. To this end, fundamental matrices are introduced and described in terms of their structure. Moreover, some emerging properties are analyzed, allowing one to build the Coriolis/centripetal matrix in a simple way. Lets start with the definition of the matrix M_{D} which from now onward will be called the inertia derivative matrix.

Definition 1 (Inertia derivative matrix).

$$M_{\mathrm{D}}(q, x) = \sum_{j=1}^{n} \frac{\partial M(q)}{\partial q_j} x e_j^T \tag{19}$$

where $M(q)$ is a generalized inertia matrix of dimension $n \times n$, e_j a unitary vector of dimension n with the value 1 *in the position j, and x is an arbitrary vector of dimension n.*

It is noted that if x represents the generalized velocity vector, the matrix M_{D} will stand for the gradient of the generalized momentum with respect to the position coordinates q. This means that the gradient of the kinetic energy as a quadratic form $\frac{1}{2}x^T M(q)x$ relative to the joint position can be written down as $\frac{1}{2}M_{\mathrm{D}}^T(q, x)x$. Another representation of M_{D} is the following.

Property 2. *The matrix $M_{\mathrm{D}}(q, x)$ can be expressed as*

$$M_{\mathrm{D}}(q, x) = \sum_{i=1}^{n} \frac{\partial M_i}{\partial q} x_i. \tag{20}$$

Proof. It is very easy to show that $\frac{\partial M(q)}{\partial q_j} x e_j^T = \sum_{i=1}^{n} \frac{\partial M_i}{\partial q_j} x_i e_j^T$ since the following intermediate equation is obtained

$$\frac{\partial M(q)}{\partial q_j} x e_j^T = \left(\sum_{i=1}^{n} \frac{\partial M_i}{\partial q_j} e_i^T\right) x e_j^T = \sum_{i=1}^{n} \frac{\partial M_i}{\partial q_j} x_i e_j^T. \tag{21}$$

Using definition 1 and identity 21 the hypothesis of the property is concluded

$$M_{\mathrm{D}}(q, x) = \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial M_i}{\partial q_j} x_i e_j^T = \sum_{i=1}^{n} \frac{\partial M_i}{\partial q} x_i. \tag{22}$$

Q.E.D.

Now a new matrix will be introduced and from now onward will be called as inertia velocity matrix, playing a central role in the representation theory.[15]

Definition 3 (Inertia velocity matrix). *Let us define $M_{\mathrm{v}}(q, x)$ in the following way*

$$M_{\mathrm{v}}(q, x) = \left(\frac{\partial M_1}{\partial q} x, \ldots, \frac{\partial M_n}{\partial q} x\right) = \sum_{i=1}^{n} \frac{\partial M_i}{\partial q} x e_i^T. \tag{23}$$

The inertia velocity matrix $M_{\mathrm{v}}(q, x)$ gets its name from the fact that when $x = \dot{q}$, the term $M_{\mathrm{v}}(q, \dot{q})$ will be the time differentiation of the generalized inertia matrix, i.e. $\dot{M}(q)$. The following property provides an alternative way to write the matrix M_{v}.

Property 4. *The inertia velocity matrix can also be expressed as*

$$M_{\mathrm{v}}(q, x) = \sum_{i=1}^{n} \frac{\partial M(q)}{\partial q_i} x_i. \tag{24}$$

Proof. It is known that $\frac{\partial M_i}{\partial q} x e_i^T = \sum_{j=1}^{n} \frac{\partial M_i}{\partial q_j} x_j e_i^T$ and using the definition 3 the property is proved as follows

$$\begin{aligned} M_{\mathrm{v}}(q, x) &= \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial M_i}{\partial q_j} x_j e_i^T \\ &= \sum_{j=1}^{n}\left(\sum_{i=1}^{n} \frac{\partial M_i}{\partial q_j} e_i^T\right) x_j \\ &= \sum_{j=1}^{n} \frac{\partial M(q)}{\partial q_j} x_j. \end{aligned} \tag{25}$$

Q.E.D.

https://doi.org/10.1017/S0263574708005067 Published online by Cambridge University Press

3.1. Properties of the fundamental matrices

Subsequently, some properties related to the fundamental matrices are analyzed. Following a systematic methodology, the properties have been classified into two groups: commutation properties and representation properties.

3.1.1. Commutation properties. Commutation properties permit interchange of an external arbitrary vector y and a vector x passed to a fundamental matrix as an argument. The following property makes possible the commutation while keeping the type of the fundamental matrices. This means that the transpose of the inertia derivative matrix can be transformed into the same structure by simply interchanging the roles of x and y.

Property 5. $M_{\mathrm{D}}^T(q,x)y = M_{\mathrm{D}}^T(q,y)x$.

The proof of this property follows directly from the definition of M_{D}. The following property allows to pass from a type of fundamental matrix to another commuting the vectors x and y.

Property 6. $M_{\mathrm{v}}(q,x)y = M_{\mathrm{D}}(q,y)x$.

Proof. $M_{\mathrm{v}}(q,x)y = \sum_{i=1}^{n} \frac{\partial M_i}{\partial q} x e_i^T y = \sum_{i=1}^{n} \frac{\partial M_i}{\partial q} x y_i = M_{\mathrm{D}}(q,y)x$.

Q.E.D.

3.1.2. Properties of representation of the Coriolis/centripetal matrix. These properties are very important to describe the Coriolis/centripetal matrix from the fundamental structures.

Property 7. $M_{\mathrm{D}}^T(q,\dot{q}) = \sum_{i=1}^{n} \frac{\partial M_i^T}{\partial q} \dot{q} e_i^T$.

Proof. First of all, the transpose of the inertia derivative matrix can be represented as $M_{\mathrm{D}}^T(q,\dot{q}) = \sum_{j=1}^{n} e_j \dot{q}^T \sum_{i=1}^{n} \frac{\partial M_i}{\partial q_j} e_i^T$ using the definition 1 and the fact that the differentiation of the inertia matrix with respect to the generalized coordinate q_j is $\frac{\partial M(q)}{\partial q_j} = \sum_{i=1}^{n} \frac{\partial M_i}{\partial q_j} e_i^T$. Since scalar product is commutative the following expression is derived, $\dot{q}^T \frac{\partial M_i}{\partial q_j} = \frac{\partial M_i^T}{\partial q_j} \dot{q}$, and as a result $M_{\mathrm{D}}^T(q,\dot{q}) = \sum_{j=1}^{n} e_j \sum_{i=1}^{n} \frac{\partial M_i^T}{\partial q_j} \dot{q} e_i^T$. The order of summation is needed to be permutated to get the proposed identity.

$$M_{\mathrm{D}}^T(q,\dot{q}) = \sum_{i=1}^{n}\sum_{j=1}^{n} \left(\frac{\partial M_i}{\partial q_j} e_j^T\right)^T \dot{q} e_i^T = \sum_{i=1}^{n} \frac{\partial M_i^T}{\partial q} \dot{q} e_i^T$$

Q.E.D.

The following representations of the Coriolis/centripetal matrix are introduced as properties derived from M_{D} and M_{v}.

Property 8. *The matrix of Coriolis/centripetal effects can be expressed as*

$$C(q,\dot{q}) = \frac{1}{2}\left(M_{\mathrm{v}}(q,\dot{q}) + M_{\mathrm{D}}(q,\dot{q}) - M_{\mathrm{D}}^T(q,\dot{q})\right) = \frac{1}{2}(M_{\mathrm{v}}(q,\dot{q}) + J(q,\dot{q})) \tag{26}$$

where $J(q,\dot{q}) = M_{\mathrm{D}}(q,\dot{q}) - M_{\mathrm{D}}^T(q,\dot{q})$ is a skew-symmetric matrix, i.e. $J(q,\dot{q}) = -J^T(q,\dot{q})$.

Proof. This is an immediate consequence of the representation of $C(q,\dot{q})$ by means of Eq. (8), property 7, and the fact that the inertia velocity matrix is $M_{\mathrm{v}}(q,\dot{q}) = \dot{M}(q)$.

Q.E.D.

In a general way, the following representation can be derived

$$C(q,x) = \frac{1}{2}(M_{\mathrm{v}}(q,x) + J(q,x)) \tag{27}$$

where x is a vector of dimension n and $J(q,x) = M_{\mathrm{D}}(q,x) - M_{\mathrm{D}}^T(q,x)$ is a skew-symmetric matrix. It is remarkable that the definition of the Coriolis/centripetal matrix presented earlier is different from the definition given in ref. 15 in the identity 2, $C(q,z)z = \frac{1}{2}(M_{\mathrm{D}}(q,z) - J(q,z)z)$. An interesting property which is a direct implication of the property 8 is that, by setting $x = y$ in $C(q,x)y$.

Property 9. *The Coriolis/centripetal force can be represented as*

$$C(q,\dot{q})\dot{q} = \left(\dot{M}(q) - \frac{1}{2}M_{\mathrm{D}}^T(q,\dot{q})\right)\dot{q} \tag{28}$$

or in a general form as

$$C(q,x)x = M_{\mathrm{v}}(q,x)x - \frac{1}{2}M_{\mathrm{D}}^T(q,x)x \tag{29}$$

where x is an arbitrary vector of dimension n.

Property 10. *The Coriolis/centripetal matrix commutes with external vectors*

$$C(q,x)y = C(q,y)x. \tag{30}$$

Proof. In order to prove this point the representation of the Coriolis/centripetal matrix will be used as a sum of the inertia velocity matrix, $M_{\mathrm{v}}(q,x)$ and the skew-symmetric matrix $J(q,x)$ given by Eq. (27).

$$C(q,x)y = \frac{1}{2}(M_{\mathrm{v}}(q,x)y + J(q,x)y) \tag{31}$$

On one hand, it is known that $J(q,x) = M_{\mathrm{D}}(q,x) - M_{\mathrm{D}}^T(q,x)$. Using the commutation properties (5) and (6) the following expression is derived

$$C(q,x)y = \frac{1}{2}(M_{\mathrm{D}}(q,y)x - M_{\mathrm{D}}^T(q,y)x + M_{\mathrm{D}}(q,x)y). \tag{32}$$

On the other hand, $J(q,y)x = M_{\mathrm{D}}(q,y)x - M_{\mathrm{D}}^T(q,y)x$ and then applying the commutation property 6 to $M_{\mathrm{D}}(q,x)y$ the following result is achieved as claimed

$$C(q,x)y = \frac{1}{2}(J(q,y)x + M_{\mathrm{v}}(q,y)x) = C(q,y)x \tag{33}$$

Q.E.D.

https://doi.org/10.1017/S0263574708005067 Published online by Cambridge University Press

3.2. Fundamental matrices revisited

This section pays due attention to how a tensor product can be constructed from the definition of M_{D}. Similar to the previous step the bilinear nature of fundamental matrices is analyzed.

Lemma 11. *Let $\Omega_x, \Omega_y, \Omega_q \subseteq \mathbb{R}^n$ be n-dimensional subspaces over the field $\mathbb{R}$, and $W \subseteq \mathbb{R}^n$ an $\mathbb{R}$-vector subspace, then the maps $\sigma_{\mathrm{D}}, \sigma_{\mathrm{v}} : \Omega_x \times \Omega_y \to W$ defined as $\sigma_{\mathrm{D}}(x, y) = M_{\mathrm{D}}(q, x)y$ and $\sigma_{\mathrm{v}}(x, y) = M_{\mathrm{v}}(q, x)y$, for a given $q \in \Omega_q$, are bilinear maps.*

Proof. For all $x, x' \in \Omega_x$, $y, y' \in \Omega_y$, $q \in \Omega_q$, $\alpha \in \mathbb{R}$, the following properties hold:

(i) $\sigma_{\mathrm{D}}(x + x', y) = M_{\mathrm{D}}(q, x + x')y = \sum_{j=1}^{n} \frac{\partial M(q)}{\partial q_j}(x + x')y_j = \sum_{j=1}^{n} \frac{\partial M(q)}{\partial q_j} x y_j + \sum_{j=1}^{n} \frac{\partial M(q)}{\partial q_j} x' y_j = \sigma_{\mathrm{D}}(x, y) + \sigma_{\mathrm{D}}(x', y)$

(ii) $\sigma_{\mathrm{D}}(x, y + y') = M_{\mathrm{D}}(q, x)(y + y') = M_{\mathrm{D}}(q, x)y + M_{\mathrm{D}}(q, x)y' = \sigma_{\mathrm{D}}(x, y) + \sigma_{\mathrm{D}}(x, y')$

(iii) $\sigma_{\mathrm{D}}(\alpha x, y) = \sigma_{\mathrm{D}}(x, \alpha y) = \alpha M_{\mathrm{D}}(x, y) = \alpha \sigma_{\mathrm{D}}(x, y)$ Analogously it is proved that σ_{v} is a bilinear map. Q.E.D.

From now onward I shall write σ to refer to either σ_{D} or σ_{v}. Can σ be encoded in a natural way using just a few real numbers? Pick a canonical base of Ω_x, Ω_y, and W, for instance $\mathcal{B} = \{e_1, e_2, \ldots, e_n\}$. If one knows $\sigma(e_i, e_j)$ whenever $e_i \in \mathbb{R}^n$ is a basis vector then by bilinearity one knows $\sigma(x, y)$ for all pairs (x, y) in $\Omega_x \times \Omega_y$. But $\sigma(e_i, e_j)$ is a vector in W and therefore can be written in terms of the basis of W. Thus, it is enough to specify n^3 numbers in order to fully specify σ. As a matter of fact, this is the cause that Coriolis/centripetal matrix requires a complexity $O(n^3)$ to be computed. Furthermore, it is not hard to see that every n^3 grid of numbers specifies the bilinear map σ: the number in position (i, j, k) gives us the kth coordinate of $\sigma(e_i, e_j)$.

For the bilinear maps σ_{D} and σ_{v} we have the following expressions:

$$\sigma_{\mathrm{D}}(x, y) = M_{\mathrm{D}}(q, x)y = \sum_{j=1}^{n} \frac{\partial M(q)}{\partial q_j} x y_j = \sum_{j=1}^{n} \sum_{i=1}^{n} \frac{\partial M_i(q)}{\partial q_j} x_i y_j$$

$$\sigma_{\mathrm{v}}(x, y) = M_{\mathrm{v}}(q, x)y = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial M_j(q)}{\partial q_i} x_i y_j.$$

Henceforth, $\sigma_{\mathrm{D}}(e_i, e_j) = \frac{\partial M_i(q)}{\partial q_j} = \sigma_{\mathrm{v}}(e_j, e_i)$. This is not surprising since M_{D} and M_{v} commutes.

This formulation defines a tensor product $\Omega_x \otimes_\sigma \Omega_y$. This space with associated bilinear map $\theta : \Omega_x \times \Omega_y \to \Omega_x \otimes_\sigma \Omega_y$ is the unique (up to isomorphism) vector space such that uniquely factorizes the bilinear maps σ_{D} and σ_{v}. There exists a bilinear map $t : \Omega_x \times \Omega_y \to \Omega_x \otimes_\sigma \Omega_y$ such that $t(x, y) = x \otimes_\sigma y$. Suppose that Ω_x and Ω_y are finite-dimensional vector spaces with canonical bases, so that we can think of a typical vector in Ω_x in coordinate form as $x = (x_1, x_2, \ldots, x_n)$, and a typical vector in Ω_y as $y = (y_1, y_2, \ldots, y_n)$. Then $x \otimes_\sigma y$ can be thought of as the $n \times n$ matrix xy^T of rank 1. For this reason, an element of $\Omega_x \otimes_\sigma \Omega_y$ of the form $x \otimes_\sigma y$ is called a rank-1 tensor. A typical element of $\Omega_x \otimes_\sigma \Omega_y$ will be a linear combination of elements $x \otimes_\sigma y$, whose rank is defined to be the smallest number of rank-1 tensors to add together to make it.[1] Vector spaces $\mathbb{R}^n \times \mathbb{R}^n$ and $\mathbb{R}^{n^2}$ are isomorphic, so that matrix xy^T is isomorphic to the Kronecker product $y \otimes x$. As a result, the bilinear map θ can be defined as $\theta(x, y) = y \otimes x$.

Any bilinear map $\sigma : \Omega_x \times \Omega_y \to W$ factorizes uniquely through $\Omega_x \otimes_\sigma \Omega_y$, i.e. there exists a unique homomorphism $\varsigma : \Omega_x \otimes_\sigma \Omega_y \to W$ such that $\sigma = \varsigma \circ \theta$. The bilinear map σ_{D} can be written as follows:

$$\begin{aligned}
\sigma_{\mathrm{D}}(x, y) &= M_{\mathrm{D}}(q, x)y \\
&= \sum_{i=1}^{n} \frac{\partial M(q)}{\partial q_i} x y_i \\
&= (y^T \otimes I_n) \sum_{i=1}^{n} \left(e_i \otimes \frac{\partial M(q)}{\partial q_i} x \right) \\
&= \sum_{i=1}^{n} \left(e_i^T \otimes \frac{\partial M(q)}{\partial q_i} \right)(y \otimes x) \\
&= \left(\frac{\partial M(q)}{\partial q} \right)^T (y \otimes x)
\end{aligned}$$

so that the homomorphism ς is expressed as $\varsigma(z) = (\frac{\partial M(q)}{\partial q})^T z$ for all $z \in \Omega_x \otimes_\sigma \Omega_y$. The gradient of $M(q)$ with respect to q is written in terms of the basic elements $\sigma(e_k, e_i)$ as follows:

$$\begin{aligned}
\frac{\partial M(q)}{\partial q} &= \sum_{i=1}^{n} \left(e_i \otimes \sum_{k=1}^{n} \frac{\partial M_k(q)}{\partial q_i} e_k^T \right) \\
&= \sum_{i=1}^{n} \left(e_i \otimes \sum_{k=1}^{n} \sigma_{\mathrm{D}}(e_k, e_i) e_k^T \right) \\
&= \sum_{i=1}^{n} \sum_{k=1}^{n} e_i \otimes e_k^T \otimes \sigma_{\mathrm{D}}(e_k, e_i) \\
&= \sum_{i=1}^{n} \sum_{k=1}^{n} E_{ik} \otimes \sigma_{\mathrm{D}}(e_k, e_i).
\end{aligned}$$

Matrix M_{D} has a meaningful physical interpretation: the term $\frac{1}{2} M_{\mathrm{D}}^T(q, \dot{q})\dot{q}$ represents a generalized force caused by variations of the kinetic energy $K(q, \dot{q})$ with respect to the generalized position q, where $K(q, \dot{q})$ is a quadratic form in the generalized velocities $\dot{q}$. In fact, the application of the Lagrangian formulation implies the appearance of such a term.[16] This confirms the transformative nature of the fundamental matrices.

[1] This coincides with the usual definition of the rank of a matrix.

https://doi.org/10.1017/S0263574708005067 Published online by Cambridge University Press

4. Example

This section is devoted to provide additional insight in understanding the meaning of the fundamental matrices. Previously, it is worth emphasizing that the factorization is innovative for the reasons immediately mentioned. Note that the definition given in this paper is general allowing one to treat with the computation of $C(q, x)y$ where $x \neq y$. In this sense, the formula proposed by Professor Wen can be seen as a particular case when $x = y$. By restricting our attention to the problem of designing inertia-related adaptive controllers, it is required to cope with the case $C(q, \dot{q})\dot{q}_r$ where $\dot{q}_r(t) \triangleq \dot{q}_d(t) + \Lambda e(t)$, so the identity 2 in ref. [15] is not sufficient for solve the problem. Moreover, in the proof of the identity 2 (ref. [15], p. 1364) it is claimed that $C(q, z) = \frac{1}{2}(M_{\mathrm{D}}(q, z) - J(q, z))$, however this expression does not work in general. The following example serves to illustrate the basic ideas presented earlier. Let us consider the following generalized inertia matrix extracted from ref. [17] (p. 123):

$$M(q) = \begin{pmatrix} m_1 l_{c_1}^2 + m_2\left(l_1^2 + l_{c_2}^2 + 2l_1 l_{c_2}\cos(q_2)\right) + I_1 + I_2 & m_2\left(l_{c_2}^2 + l_1 l_{c_2}\cos(q_2)\right) + I_2 \\ m_2\left(l_{c_2}^2 + l_1 l_{c_2}\cos(q_2)\right) + I_2 & m_2 l_{c_2}^2 + I_2 \end{pmatrix}. \tag{34}$$

Inertia derivative matrix is directly derived from property 2:

$$\begin{aligned} M_{\mathrm{D}}(q, x) &= \sum_{i=1}^{n} \frac{\partial M_i(q)}{\partial q} x_i \\ &= -m_2 l_1 l_{c_2}\sin(q_2)\left[\begin{pmatrix} 0 & 2 \\ 0 & 1 \end{pmatrix} x_1 + \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} x_2\right] \\ &= -m_2 l_1 l_{c_2}\sin(q_2)\begin{pmatrix} 0 & 2x_1 + x_2 \\ 0 & x_1 \end{pmatrix} \end{aligned}$$

where $M_i(q)$ is the ith column of the generalized inertia matrix. Wen and Bayard have defined $J(q, x) = M_{\mathrm{D}}^T(q, x) - M_{\mathrm{D}}(q, x)$ (cf. Eq. 2.6 in ref. [15])

$$\begin{aligned} J(q, x) &= M_{\mathrm{D}}^T(q, x) - M_{\mathrm{D}}(q, x) \\ &= -m_2 l_1 l_{c_2}\sin(q_2)\begin{pmatrix} 0 & 2x_1 + x_2 \\ -(2x_1 + x_2) & 0 \end{pmatrix}. \end{aligned}$$

Henceforth,

$$\begin{aligned} C(q, x) &= \frac{1}{2}(M_{\mathrm{D}}(q, x) - J(q, x)) \\ &= -\frac{1}{2} m_2 l_1 l_2 \sin(q_2)\begin{pmatrix} 0 & 2(2x_1 + x_2) \\ -(2x_1 + x_2) & x_1 \end{pmatrix}. \end{aligned} \tag{35}$$

The Coriolis/centripetal matrix form the Lagrangian dynamic model (cf. ref. [17], p. 123) is

$$C(q, x) = -m_2 l_1 l_{c_2}\sin(q_2)\begin{pmatrix} x_2 & x_1 + x_2 \\ -x_1 & 0 \end{pmatrix} \tag{36}$$

which is quite different from (35). Now, we will try our hand at using the formulation given by the new matrix $M_{\mathrm{v}}(q, x)$

$$\begin{aligned} M_{\mathrm{v}}(q, x) &= \sum_{i=1}^{n} \frac{\partial M(q)}{\partial q_i} x_i \\ &= -m_2 l_1 l_{c_2}\sin(q_2)\begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix} x_2. \end{aligned}$$

According to the new formulation

$$C(q, x) = \frac{1}{2}(M_{\mathrm{v}}(q, x) + J(q, x)) \tag{37}$$

where x is a vector of dimension n and $J(q, x) = M_{\mathrm{D}}(q, x) - M_{\mathrm{D}}^T(q, x)$ is a skew-symmetric matrix, i.e. $J(q, x) = -J^T(q, x)$. Then,

$$\begin{aligned} J(q, x) &= M_{\mathrm{D}}(q, x) - M_{\mathrm{D}}^T(q, x) \\ &= -m_2 l_1 l_{c_2}\sin(q_2)\begin{pmatrix} 0 & 2x_1 + x_2 \\ -(2x_1 + x_2) & 0 \end{pmatrix}. \end{aligned}$$

Henceforth,

$$\begin{aligned} C(q, x) &= \frac{1}{2}(M_{\mathrm{v}}(q, x) + J(q, x)) \\ &= -\frac{1}{2} m_2 l_1 l_{c_2}\sin(q_2) \\ &\quad \times\left[\begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix} x_2 + \begin{pmatrix} 0 & 2x_1 + x_2 \\ -(2x_1 + x_2) & 0 \end{pmatrix}\right] \\ &= -m_2 l_1 l_{c_2}\sin(q_2)\begin{pmatrix} x_2 & x_1 + x_2 \\ -x_1 & 0 \end{pmatrix} \end{aligned}$$

which agrees with (36). Fortunately, the identity 2 in ref. [15] remains valid because of the commutation property $M_{\mathrm{D}}(q, x)y = M_{\mathrm{v}}(q, y)x$. With this simple example we put forward the necessity of introducing the matrix M_{v}. Due to the discrepancy between the Wen's formulation and the one obtained by traditional methods, we have plumped for a new factorization of the Coriolis/centripetal matrix.

5. Applications to Control

In this section, we try to reduce the gap between theory and practice by transferring the latest theoretical insights into practical applications. In particular, it is devoted to illustrate how the earlier relationships can be used to build adaptive control laws. From Section 2, it is clear that the nonlinear Coriolis/centripetal matrix $C(q, \dot{q})$ in the robot dynamics is not unique. In approximation-based control, it is necessary to select the correct version of $C(q, \dot{q})$; specifically one must use the version of $C(q, \dot{q})$ so that the skew-symmetric property holds. In this context, the new factorization can be applied to a large class of controllers based on approximating unknown robot functions; this class is commonly referred to as *filtered-error approximation-based control* and is characterized by including various forms of adaptive and robust control techniques, as well as learning control and neural network

https://doi.org/10.1017/S0263574708005067 Published online by Cambridge University Press

(NN) techniques. In the following subsections, a particular emphasis is placed on the adaptive control and NN control.

5.1. Adaptive control by an inertia-related approach

The starting point of our investigations in this section is the *linearity-in-the-parameter property* (LIP property). Then, an *inertia-related linearization approach* is developed in order to derive adaptive update rules.

5.1.1. Linearity in the parameters. The new factorization of the Coriolis/centripetal matrix described in the preceding section possesses a number of properties that are very useful in reducing the complexity of dynamics equations in many practical applications. In this section by restricting our attention to the LIP property we try to illustrate how this information may be effectively used in building appropriate models for adaptive control. As a matter of fact a large class of adaptive controllers for robot manipulators assume linearity with respect to the dynamic parameters characterizing the manipulator links. This property is relevant since some or all of the parameters may be unknown; thus the dynamics are linear in the unknown terms. All the constant parameters of interest, such as link lengths, masses, and moments of inertia, appear as coefficients of known functions of the generalized coordinates.

Now that it is known how fundamental matrices M_{D} and M_{v} enable the representation of C, the underlying reason for some of the special relationships developed in Section 2 between them will be clarified. Assume that the columns $M_i(q)$ of $M(q)$ can be expressed as

$$M_i(q) = \Psi_{m_i}(q)\xi_{m_i}$$

where $\Psi_{m_i}(q) \in \mathcal{M}_{n\times l_i}(\mathbb{R})$ is a regression matrix of known robot functions and $\xi_{m_i} \in \mathbb{R}^{l_i}$ is a vector of unknown parameters (e.g. masses, inertia moments, and link parameters). Then, with an arbitrary vector $x \in \mathbb{R}^n$, there follows the decomposition

$$M(q)x = \sum_{i=1}^{n} M_i(q)x_i = \left(\sum_{i=1}^{n} \Psi_{m_i}(q)x_i\xi_{m_i}\right). \tag{38}$$

On the assumption that the same regression matrix serves for each column, i.e. $\Psi_{m_1}(q) = \Psi_{m_2}(q) = \cdots = \Psi_{m_n}(q) = \Psi_m(q) \in \mathcal{M}_{n\times l}(\mathbb{R})$, one may rewrite Eq. (38) by resorting to the Kronecker product as

$$M(q)x = \Phi_m(q, x)\xi_m$$

where $\xi_m \in \mathbb{R}^l$ is an l-dimensional vector of parameters and $\Phi_m(q, x) = (x^T \otimes \Psi_m(q)) \in \mathcal{M}_{n\times l}(\mathbb{R})$ is the regression matrix of the generalized inertia matrix. For the sake of simplicity in the following, the matrix derivative of $\Psi_m(q)$ with $q \in \mathbb{R}^n$ will be denoted as

$$\frac{\partial \Psi_m(q)}{\partial q} \triangleq \begin{pmatrix} \frac{\partial \Psi_m(q)}{\partial q_1} \\ \vdots \\ \frac{\partial \Psi_m(q)}{\partial q_n} \end{pmatrix} \in \mathcal{M}_{n^2\times l}(\mathbb{R}).$$

The following lemmas are helpful for characterizing Coriolis/centripetal matrix in LIP form.

Lemma 12 (LIP form of M_{D}). *For arbitrary vectors $x, y \in \mathbb{R}^n$, $q \in \Omega_q \subseteq \mathbb{R}^n$, the inertia derivative matrix can be written in the LIP property as follows*

$$M_{\mathrm{D}}(q, x)y = \Phi_{\mathrm{D}}(q, x, y)\xi_m$$

where $\xi_m \in \mathbb{R}^l$ is the parameter vector of the generalized inertia matrix and $\Phi_{\mathrm{D}} \in \mathcal{M}_{n\times l}(\mathbb{R})$ a regression matrix

$$\Phi_{\mathrm{D}}(q, x, y) = (y^T \otimes I_n)\psi(q, x)$$
$$\psi(q, x) = \left(x^T \otimes \frac{\partial \Psi_m(q)}{\partial q}\right).$$

Proof. This fact can be straightforward proved by resorting to the definition of M_{D} given earlier

$$\begin{aligned} M_{\mathrm{D}}(q, x)y &= \sum_{i=1}^{n} \frac{\partial M(q)}{\partial q_i} x y_i = \left(\sum_{i=1}^{n} \frac{\partial \Phi_m(q, x)}{\partial q_i} y_i\right)\xi_m \\ &= \left(x^T \otimes \sum_{i=1}^{n} \frac{\partial \Psi_m(q)}{\partial q_i} y_i\right)\xi_m \\ &= \Phi_{\mathrm{D}}(q, x, y)\xi_m. \end{aligned} \tag{39}$$

By closely examining the structure of Φ_{D} and using the property

$$\sum_{i=1}^{n} \frac{\partial \Psi_m(q)}{\partial q_i} y_i = (y^T \otimes I_n)\frac{\partial \Psi_m(q)}{\partial q}$$

it is easy to conclude that

$$\begin{aligned} \Phi_{\mathrm{D}}(q, x, y) &= x^T \otimes \left((y^T \otimes I_n)\frac{\partial \Psi_m(q)}{\partial q}\right) \\ &= \left[x \otimes \left(\frac{\partial \Psi_m^T(q)}{\partial q}(y \otimes I_n)\right)\right]^T \\ &= \left[\left(x \otimes \frac{\partial \Psi_m^T(q)}{\partial q}\right)(y \otimes I_n)\right]^T \\ &= (y^T \otimes I_n)\psi(q, x). \end{aligned}$$

Q.E.D.

Now that it is known how M_{D} can be written in LIP form, the main challenge is to extend this result to its transpose. This is more difficult since it involves a permutation matrix as stated in the following lemma.

Lemma 13 (LIP form of M_D^T). *Let $x, y \in \mathbb{R}^n$, $q \in \Omega_q \subseteq \mathbb{R}^n$ arbitrary vectors with appropriate units and P a permutation*

https://doi.org/10.1017/S0263574708005067 Published online by Cambridge University Press

matrix[2] *defined as follows*

$$P(n,m) = \sum_{i=1}^{n}\sum_{j=1}^{n} E_{ij} \otimes E_{ji}$$

where $E_{ij} = e_i e_j^T$ *is a unit matrix having one in position* (i, j) *and all other entries are zero. Under these conditions, the term* $M_{\mathrm{D}}^T(q,x)$ *can be written in LIP form as follows*

$$M_{\mathrm{D}}^T(q,x)y = (x^T \otimes I_n)P(n,n)\psi(q,y)\xi_m.$$

Proof. By virtue of definition 1 and lemma 12, the generalized force $M_{\mathrm{D}}^T(q,x)y$ can be directly derived in the LIP form which can be detailed into

$$\begin{aligned} M_{\mathrm{D}}^T(q,x)y &= \sum_{i=1}^{n} e_i \left(x^T \frac{\partial M(q)}{\partial q_i} y \right) \\ &= (I_n \otimes x^T)\frac{\partial M(q)}{\partial q} y \qquad (40) \\ &= (I_n \otimes x^T)\frac{\partial \Phi_m(q,y)}{\partial q}\xi_m \\ &= (I_n \otimes x^T)\left(y^T \otimes \frac{\partial \Psi_m(q)}{\partial q} \right)\xi_m. \qquad (41) \end{aligned}$$

With reference to the property $B \otimes A = P(m,p)^T(A \otimes B)P(n,q)$ for all $A \in \mathcal{M}_{m\times n}(\mathbb{R})$ and $B \in \mathcal{M}_{p\times q}(\mathbb{R})$ (see Corollary 4.3.10 in ref. [18], p. 260), the term $(x^T \otimes I_n)$ can be commutated as

$$(x^T \otimes I_n) = P(n,1)^T(I_n \otimes x^T)P(n,n).$$

The last identity can be simplified further by exploiting the structure of the permutation matrices. In particular, it is easy to show that $P(n,1) = P(1,n) = I_n$ (see problem 18 of ref. [18], Section 4.3, p. 265), which leads to

$$(I_n \otimes x^T) = (x^T \otimes I_n)P(n,n).$$

Therefore,

$$\begin{aligned} M_{\mathrm{D}}^T(q,x)y &= (x^T \otimes I_n)P(n,n)\left(y^T \otimes \frac{\partial \Psi_m(q)}{\partial q} \right)\xi_m \\ &= (x^T \otimes I_n)P(n,n)\psi(q,y)\xi_m. \end{aligned}$$

Q.E.D.

Remark 14. *The permutation matrix in the last lemma can also be written as*

$$P(n,n) = \sum_{i=1}^{n} e_i \otimes I_n \otimes e_i^T = \sum_{i=1}^{n} e_i^T \otimes I_n \otimes e_i.$$

[2] $P(m,n)$ is a matrix of zeroes and ones for which $P(m,n)^{-1} = P(m,n)^T = P(m,n)$.

For more details the reader is referred to the problem 21 of ref. [18], Section 4.3, p. 286.

On the basis of the description of M_{D} and M_{D}^T in LIP form, the skew-symmetric matrix J can also be represented as LIP.

Lemma 15 (Matrix J in LIP form). *Let* $x, y \in \mathbb{R}^n$, $q \in \Omega_q \subseteq \mathbb{R}^n$ *be arbitrary vectors with appropriate units. The skew-symmetric matrix* $J(q,x)$ *can be expressed as linear in the parameters,*

$$J(q,x)y = \Phi_J(q,x,y)\xi_m$$

where $\Phi_J(q,x,y) = (y^T \otimes I_n)(I_{n^2} - P(n,n))\psi(q,x)$.

Proof. It is straightforward that

$$\begin{aligned} J(q,x)y &= M_{\mathrm{D}}(q,x)y - M_{\mathrm{D}}^T(q,x)y \\ &= (y^T \otimes I_n)(I_{n^2} - P(n,n))\psi(q,x)\xi_m. \end{aligned}$$

Q.E.D.

Lemma 16 (LIP form of M_{v}). *For arbitrary vectors* $x, y \in \mathbb{R}^n$, $q \in \Omega_q \subseteq \mathbb{R}^n$, *the inertia velocity matrix can be written as*

$$M_{\mathrm{v}}(q,x)y = \Phi_{\mathrm{v}}(q,x,y)\xi_m$$

where $\Phi_{\mathrm{v}}(q,x,y) = \sum_{i=1}^{n} \frac{\partial \Phi_m(q,y)}{\partial q_i} x_i$.

Proof. The proof is direct.

Q.E.D.

Remark 17. *It is observed that* $\Phi_D(q,x,y) = \Phi_{\mathrm{v}}(q,y,x)$, *which is consistent with the commutation property.*

Remark 18. *An alternative way for* J *is obtained by resorting to the commutation property*

$$\begin{aligned} M_{\mathrm{v}}(q,x)y - M_{\mathrm{D}}^T(q,x)y &= M_{\mathrm{D}}(q,y)x - M_{\mathrm{D}}^T(q,y)x \\ &= (x^T \otimes I_n)(I_{n^2} - P(n,n)) \\ &\quad \times \psi(q,y)\xi_m \\ &= J(q,y)x. \end{aligned}$$

Remarkably, the above lemmas can be conveniently used to write the Coriolis/centripetal matrix in LIP form.

Proposition 19 (Coriolis/centripetal matrix in LIP form). *Let* $x, y \in \mathbb{R}^n$, $q \in \Omega_q \subseteq \mathbb{R}^n$ *be the arbitrary vectors, the Coriolis/centripetal effect* $C(q,x)y$ *can be linearly factorized as a regression matrix* $\Phi_C(q,x,y)$ *and a parameter vector* ξ_m, *i.e.* $C(q,x)y = \Phi_C(q,x,y)\xi_m$, *where* Φ_C *is given by*

$$\begin{aligned} \Phi_C(q,x,y) = \frac{1}{2}[&(x^T \otimes I_n)\psi(q,y) \\ &+ (y^T \otimes I_n)(I_{n^2} - P(n,n))\psi(q,x)]. \end{aligned}$$

https://doi.org/10.1017/S0263574708005067 Published online by Cambridge University Press

Proof. By restoring to the LIP form of M_v and J, matrix C can be written as

$$C(q,x)y = \frac{1}{2}(M_v(q,x)y + J(q,x)y) \tag{42}$$

$$= \frac{1}{2}[(x^T \otimes I_n)\psi(q,y) + (y^T \otimes I_n) \times (I_{n^2} - P(n,n))\psi(q,x)]\xi_m. \tag{43}$$

Q.E.D.

5.1.2. Inertia-related linearization. Up to this point, LIP property has been analyzed via fundamental matrices. The control approach employs an *inertia-related linearization approach*, i.e. a conservation of energy formulation, as an attempt to derive update laws and control laws. To be specific, it is required to define an inertia-related Lyapunov function in the stability analysis which utilizes physical properties inherent to a mechanical manipulator (such as those presented earlier). Thus, the stability of the tracking error system is ensured by formulating the adaptive update rule and by analyzing the stability of the tracking error system at the same time. It is well known that dynamic models even though quite complex are anyhow an idealization of reality. Specifically, robots show uncertainties that are mainly found from two sources: variability of parameters and nonlinearity terms in the system. One way of dealing with parametric uncertainties would be to use the inertia-related approach, see ref. [19]. The benefit of this approach as compared to others is that it avoids a direct measurement of acceleration and the invertibility of the generalized inertia matrix, which are restrictions of some controllers such as those inspired in the adaptive computed-torque approaches.

The general form of approximation-based controller can be derived by setting

$$\tau = K_v r + \hat{f} \tag{44}$$

where r is a filtered error signal defining a derivative filter or PD,[20] i.e. $r = \dot{e} + \Lambda e$, Λ being a diagonal matrix of design parameters with big positive elements so that the system is BIBO stable. This matrix allows to filter errors so that no acceleration $\ddot{e}(t)$ will appear in the error dynamic equation. As shown in (44), the control law consists of two terms: (i) a PD-controller to guarantee a good tracking performance, $\tau_{pd} = K_v r$, where $K_v = K_v^T > 0$ is the gain matrix and (ii) a compensator of nonlinearities, $\tau_{nl} = \hat{f}$.

An important feature of this class of controllers is that of being based on the all-important *closed-loop error dynamics*, which results from the substitution of the filtered tracking error into the robot dynamics

$$M(q)\dot{r} = -C(q,\dot{q})r + f(x) - \tau$$

$$= -C(q,\dot{q})r - K_v r + \tilde{f}(x) \tag{45}$$

where $\tilde{f}(x) = f(x) - \hat{f}(x)$ stands for the functional estimation error with x being a vector of appropriate variables as will be shown later. The nonlinearity f can be computed as

$$f(q,\dot{q},\dot{q}_r,\ddot{q}_r) = M(q)\ddot{q}_r + C(q,\dot{q})\dot{q}_r + G(q)$$

where $\ddot{q}_r$ is the reference acceleration, $\ddot{q}_r(t) \triangleq \ddot{q}_d(t) + \Lambda\dot{e}(t)$, and $\dot{q}_r$ is the reference velocity, $\dot{q}_r(t) \triangleq \dot{q}_d(t) + \Lambda e(t)$. The nonlinearity f can be conveniently partitioned into several smaller terms, resulting in an added controller structure

$$f(x) = f_m(x) + f_c(x) + f_g(x) \tag{46}$$

where $f_g(x) = G(q)$. The terms f_m and f_c can be defined in terms of $\ddot{q}_r$ and $\dot{q}_r$ as follows

$$f_m(q,\ddot{q}_r) = M(q)\ddot{q}_r = \Phi_m(q,\ddot{q}_r)\xi_m$$

$$f_c(q,\dot{q},\ddot{q}_r) = C(q,\dot{q})\dot{q}_r = \Phi_c(q,\dot{q},\ddot{q}_r)\xi_m$$

$$f_g(q) = G(q) = \Phi_g(q)\xi_g$$

where $\xi_m \in \mathbb{R}^{l_m}$, $\xi_g \in \mathbb{R}^{l_g}$ are parameter vectors and Φ_m, Φ_c, Φ_g are regression matrices. Select the following inertia-related Lyapunov-like function

$$V = \frac{1}{2}r^T M(q)r + \frac{1}{2}\tilde{\xi}_m^T \Gamma_m^{-1}\tilde{\xi}_m + \frac{1}{2}\tilde{\xi}_g^T \Gamma_g^{-1}\tilde{\xi}_g \tag{47}$$

where $\tilde{\xi}_m \in \mathbb{R}^{l_m}$, $\tilde{\xi}_g \in \mathbb{R}^{l_g}$ are the parameter error vectors, $\Gamma_m \in M_{l_m \times l_m}(\mathbb{R})$, $\Gamma_g \in M_{l_g \times l_g}(\mathbb{R})$ diagonal, positive-definite matrices. Differentiating (47) with respect to time gives

$$\dot{V} = r^T M(q)\dot{r} + \frac{1}{2}r^T \dot{M}(q)r + \tilde{\xi}_m^T \Gamma_m^{-1}\dot{\tilde{\xi}}_m + \tilde{\xi}_g^T \Gamma_g^{-1}\dot{\tilde{\xi}}_g. \tag{48}$$

From (48) it is clear that we must substitute for the variable $\dot{r}$; therefore, we must write the robot equation in terms of the variable r. Using (45) the robot dynamics can be rewritten as

$$M(q)\dot{r} = -C(q,\dot{q})r - K_v r + [\Phi_m(q,\ddot{q}_r) + \Phi_c(q,\dot{q},\dot{q}_r)]\tilde{\xi}_m + \Phi_g(q)\tilde{\xi}_g. \tag{49}$$

This is the same type of parametric separation that appears in the formulation of the adaptive computed-torque controller; however, note that regression matrices are not a function of joint acceleration $\ddot{q}$. Substituting (49) into (48) gives

$$\dot{V} = -r^T K_v r + \left[r^T(\Phi_m(q,\ddot{q}_r) + \Phi_c(q,\dot{q},\dot{q}_r)) + \dot{\tilde{\xi}}_m^T \Gamma_m^{-1}\right]\tilde{\xi}_m + \left[r^T \Phi_g(q) + \dot{\tilde{\xi}}_g^T \Gamma_g^{-1}\right]\tilde{\xi}_g.$$

Now, the strategy consists of making the terms in brackets zero, which results in the following adaptive update rules:

$$\dot{\tilde{\xi}}_m = \Gamma_m\left(\Phi_m^T(q,\ddot{q}_r) + \Phi_c^T(q,\dot{q},\dot{q}_r)\right)r$$

$$= \Gamma_m\left(\left(\ddot{q}_r \otimes \Psi_m^T(q)\right)\frac{1}{2}[\psi^T(q,\dot{q}_r)(\dot{q} \otimes I_n) + \psi^T(q,\dot{q})(I_{n^2} - P(n,n))(\dot{q}_r \otimes I_n)]\right)r$$

$$\dot{\tilde{\xi}}_g = \Gamma_g \Phi_g^T(q)r.$$

https://doi.org/10.1017/S0263574708005067 Published online by Cambridge University Press

Under these adaptive laws tracking errors e and $\dot{e}$ are asymptotically stable and parameter estimates $\hat{\xi}_m$ and $\hat{\xi}_g$ are bounded.

5.2. Neural network control

Now we turn our attention to the problem of applying fundamental matrices in the design of NN controllers. The implementation is streamlined by partitioning the global scheme into several smaller subnets to obtain more efficient computation.

It is fairly easy to carry over the above discussion to the framework of NNs. The first step consists of analyzing the *universal function approximation* (UFA) property. The next step is to resort to the *filtered-error approximation-based control* in order to derive weight updating laws.

5.2.1. Universal approximation property.. Of relevant importance in NN closed-loop control applications is the UFA of NN having at least two layers. On a whole, one-layer NN does not generally have a universal approximation capability, however linear-in-the-parameter neural networks may exhibit this property under certain conditions. These networks are called as *Functional-Link Neural Networks* (FLNN) and describe a nonlinear map given by

$$y(x) = W^T \phi(x) \tag{50}$$

with $W \in \mathcal{M}_{n\times L}(\mathbb{R})$ the NN output weights and $\phi(\cdot)$ a general function from $\mathbb{R}^n$ to $\mathbb{R}^L$. Many FLNN schemes have been proposed in the literature depending on the set of activation functions, such as Gaussian or Radial Basis Functions (RBF),[21, 22] Cerebellar-Model Articulation Controller (CMAC),[23] and fuzzy logic nets.

Fortunately, FLNN can approximate functions so long as the activation functions are selected to form a *basis* satisfying two requirements on a compact simply connected set $\mathcal{S} \subseteq \mathbb{R}^n$[24]: (i) a constant function on $\mathcal{S}$ can be expressed as (50) for a finite number L of hidden-layer neurons, (ii) the functional range of (50) is dense in the space of continuous functions from $\mathcal{S}$ to $\mathbb{R}^m$ for countable L.

Define $C^m(\mathcal{S})$ as the space of continuous functions from $\mathcal{S}$ to $\mathbb{R}^m$, the UFA property states that for all $f \in C^m(\mathcal{S})$, there exist weights W such that $f(x) = W^T\phi(x) + \varepsilon$ with the *functional reconstruction error* bounded by $\|\varepsilon\| < \varepsilon_N$, where ε_N is a positive constant.

By an argument similar to that presented for LIP representation it is not difficult to show that nonlinearity $f(x)$ can be conveniently partitioned into several smaller terms, resulting in an added controller structure

$$f(x) = f_m(x) + f_c(x) + f_g(x) + f_f(x) \tag{51}$$

where

$$\begin{aligned} f_m(x) &= W_m^T \gamma_m(q, \ddot{q}_r) + E_m(q, \ddot{q}_r) \\ f_g(x) &= W_g^T \phi_g(q) + \varepsilon_g(q) \\ f_f(x) &= W_f^T \phi_f(q, \dot{q}) + \varepsilon_f(q, \dot{q}) \end{aligned}$$

with $W_m \in \mathbb{R}^{L_m\times n}$, $W_g \in \mathbb{R}^{L_g\times n}$, $W_f \in \mathbb{R}^{L_f\times n}$, matrices of unknown weights, $\gamma_m \in \mathbb{R}^{L_m}$, $\phi_g \in \mathbb{R}^{L_g}$, and $\phi_f \in \mathbb{R}^{L_f}$, vectors of nonlinear terms (called activation functions), and $E_m, \varepsilon_g, \varepsilon_f \in \mathbb{R}^n$, vectors of functional reconstruction errors.

The Coriolis/centripetal term deserves careful attention in the derivation of a linear approximator. Before providing detailed insight into the representation of $f_c(x)$, it may be observed that the term $M_{\mathrm{D}}(q, \dot{q})\dot{q}_r$ can be linearly parameterized in the following way:

$$M_{\mathrm{D}}(q, \dot{q})\dot{q}_r = W_m^T \varphi(q, \dot{q})\dot{q}_r + E_{\mathrm{D}}(q, \dot{q})\dot{q}_r$$

where $\varphi \in \mathbb{R}^{L_m\times n}$ is a matrix of nonlinearities and $E_{\mathrm{D}} \in \mathbb{R}^{n\times n}$ is the matrix of reconstruction errors. This is an immediate consequence of the description in columns of the inertia derivative matrix from definition 1. Now, notice that the representation property 8 along with the commutation property 6 permits one to write the estimation of the Coriolis/centripetal effects in LIP form as follows[12]:

$$\begin{aligned} f_c(x) = \frac{1}{2}\big(W_m^T \varphi(q, \dot{q})\dot{q}_r + W_m^T \varphi(q, \dot{q}_r)\dot{q} - \varphi^T(q, \dot{q}) W_m \dot{q}_r\big) \\ + \frac{1}{2}\big(E_D(q, \dot{q})\dot{q}_r + E_{\mathrm{V}}(q, \dot{q})\dot{q}_r - E_D^T(q, \dot{q})\dot{q}_r\big). \end{aligned}$$

There are a number of interesting differences between LIP property and UFA property:

(i) Both are linear in the tunable parameters, but the former is linear in the unknown system parameters while the latter is linear in the tunable NN weights.

(ii) The former holds for a specific function f while the latter holds for all functions $f \in C^m(\mathcal{S})$.

(iii) In the LIP property the regression matrix Φ depends on f and must be recomputed for each different f, while in the NN assumption the same basis set ϕ suffices for all $f \in C^m(\mathcal{S})$.

Note that for each different type of robot manipulator, one must recompute regression matrices Φ. As a consequence, one-layer NN controller is significantly more powerful than adaptive controllers.

5.2.2. Filtered-error approximation-based control.. With the aim of making clear the implications of the new factorization introduced in previous sections, this part will concentrate on the neural net control. Many forms of NN controllers for robot manipulators are included in the FLNN class. Most of these controllers share the characteristic of utilizing standard, gradient-type adaptation schemes and rely on the UFA assumption allowing one to write nonlinear functions as linear in the unknown parameters. Following an inertia-related approach similar to that presented in the previous subsection, the weight updating laws can be derived as

$$\begin{aligned} \hat{W}_m &= \Gamma_m \left(\left(\gamma_m(q, \ddot{q}_r) + \frac{1}{2}\varphi(q, \dot{q}_r)\dot{q} \right) r^T \right. \\ &\quad \left. + \frac{1}{2}\varphi(q, \dot{q})\big(\dot{q}_r r^T - r\dot{q}_r^T\big) \right) \\ \hat{W}_g &= \Gamma_g \phi_g(q) r^T \\ \hat{W}_f &= \Gamma_f \phi_f(q, \dot{q}) r^T \end{aligned}$$

https://doi.org/10.1017/S0263574708005067 Published online by Cambridge University Press

with $\Gamma_m = \Gamma_m^T > 0$, $\Gamma_g = \Gamma_g^T > 0$, and $\Gamma_f = \Gamma_f^T > 0$ symmetric positive-definite constant matrices (for more details the reader is referred to ref. [11]).

6. Discussion

The paper deeply investigates the two most effective fundamental matrices in the representation of the Coriolis/centripetal matrix. This approach is innovative (with respect to those proposed in the past) since it is of higher applicability and generality, being capable of coping with many different control schemes based on the approximation of nonlinear terms. The work focuses on the properties associated with these matrices, leading to a wider and deeper understanding of the Coriolis/centripetal effects, such as those on the representation and commutation. Also the practical aspects of the new representation are explained in some detail.

The paper stresses the author's view that the formulation in mechanics (in terms of factorizations) is not a closed topic and may be determined very precisely and explicitly by new structures referred to as fundamental matrices. It is well known that when a new mathematical structure is introduced, it is necessary to provide a number of properties that are inferred from it. In this sense, a complete treatment of the properties connecting M_{D} and M_{v} has been carried out, offering the reader another point of view concerning a possible way to approach the modeling. These properties are very important for the representation when designing emulators. As a consequence, the analysis presented here represents a specific topic which is very close to mechanics and deserves a special treatment.

In designing a control law a question should arise spontaneously: Are all the properties inherent to a mechanical manipulator being exploited? Until now, factorizations of the Coriolis/centripetal matrix have not been addressed as a means to implement efficient control laws. The advantages of this approach are (i) it simplifies the design, (ii) it gives added controller structure, and (iii) it makes for faster coefficient tuning algorithms. Finally, the new approach provides a guide to developers willing to adopt a matrix formulation in their applications.

References

1. L. Sciavicco and B. Siciliano, *Modelling and Control of Robot Manipulators* (Springer-Verlag, London, 2002).
2. V. D. Tourassis and C. P. Neuman, "The inertial characteristics of dynamic robot models," *Mech. Mach. Theory* **20**(1), 41–52 (1985).
3. V. D. Tourassis and C. P. Neuman, "Properties and structure of dynamic robot models for control engineering applications," *Mech. Mach. Theory* **20**(1), 27–40 (1985).
4. J. Craig, *Introduction to Robotics* (Addison-Wesley, Reading, MA, 1989).
5. M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control* (John Wiley and Sons, New York, 1989).
6. F. L. Lewis, C. T. Abdallah and D. M. Dawson, *Control of Robot Manipulators* (Macmillan Co., New York, 1993).
7. R. J. Schilling, *Fundamentals of Robotics: Analisis and Control* (Prentice Hall, Englewood Cliffs, NJ, 1990).
8. K. S. Fu, R. C. Gonzalez and C. S. G. Lee, *Robotics: Control, Sensing, Vision and Intelligence* (McGraw-Hill Education (ISE Editions), International Ed edition, June 1, 1987).
9. R. P. Paul, *Robot Manipulators: Mathematics, Programming, and Control* (The MIT Press, Cambridge, MA, Nov. 1981).
10. J. I. Mulero-Martínez, "Uniform bounds of the coriolis/centripetal matrix of serial robot manipulators," *IEEE Trans. Rob.* **23**(5), 1083–1089 (2007).
11. J. I. Mulero-Martínez, "An improved dynamic neurocontroller based on christoffel symbols," *IEEE Trans. Neural Networks* **18**(3), 865–879 (2007).
12. J. I. Mulero-Martínez, "Bandwidth of mechanical systems and design of emulators with rbf," *Neurocomputing* **70**(7–9), 1453–1465 (2007).
13. H. Asada and J.-J. Slotine, *Robot Analysis and Control* (John Wiley & Sons, New York, 1986).
14. L.-W. Tsai, *Robot Analysis: The Mechanics of Serial and Parallel Manipulators* (Interscience, New York, Feb. 1999).
15. J. T. Wen and D. S. Bayard, "New class of control laws for robotic manipulators. Part 1. Non-adaptive case," *Int. J. Control* **47**(5), 1361–1385 (1988).
16. J. I. Mulero-Martínez, "Canonical transformations used to derive robot control laws from a port-controlled hamiltonian system perspective," *Automatica* **44**(9), 2435–2440 (Sept. 2008).
17. R. Kelly, V. Santibañez and A. Loria, *Control of Robot Manipulators in Joint Space* (Springer-Verlag, London, 2005).
18. R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis* (Cambridge University Press, Cambridge, UK, 1999).
19. F. L. Lewis, D. M. Dawson and C. T. Abdallah, *Robot Manipulator Control. Theory and Practice*, 2nd ed. (Control Engineering Series, Marcel Dekker, New York, 2004).
20. J.-J. E. Slotine and H. Asada, *Robot Analysis and Control* (John Wiley and Sons, New York, 1986).
21. R. M. Sanner and J. E. Slotine, "Gaussian networks for direct adaptive control," *IEEE Trans. Neural Networks* **3**, 837–863 (1992).
22. M. M. Polycarpou and P. A. Ioannou, *Identification and Control Using Neural Network Models: Design and Stability Analysis*, Technical Report 91-09-01, Department of Electrical Engineering Systems, University of Southern California (1991).
23. J. S. Albus, "Data storage in the cerebellar model articulation controller (CMAC)," *Trans. ASME. J. Dyn. Sys. Meas. Control* **63**(3), 228–233 (1975).
24. N. Sadegh, "A perceptron network for functional identification and control of nonlinear systems," *IEEE Trans. Neural Networks* **4**, 982–988 (Nov. 1993).

https://doi.org/10.1017/S0263574708005067 Published online by Cambridge University Press