Appendix 8

Computation of the inertia matrix of tree structured robots

In this appendix, we develop a method to compute efficiently the inertia matrix of tree structured robots. Note that a serial robot is a special case of the tree structured robot. This method is based on the utilization of a simplified special case of Newton-Euler algorithm and on the concept of composite links [Khalil 90b].

A8.1. Inertial parameters of a composite link

The composite link j^+ is composed of link j and of the links supported by link j (Figure A8.1). The inertial parameters of the composite link j^+ can be calculated in terms of the standard parameters (or base parameters) of its links using the following recursive algorithm:

i) initialization. For j = 1, ..., n:

$$^{j}\mathbf{J}_j^+ = {}^{j}\mathbf{J}_j,\ {}^{j}\mathbf{MS}_j^+ = {}^{j}\mathbf{MS}_j,\ M_j^+ = M_j$$

We recall that a(j) indicates the link that is antecedent to link j;

ii) for j = n, ..., 2 and a(j) ≠ 0:

$$^{a(j)}\mathbf{J}_{a(j)}^+ = {}^{a(j)}\mathbf{J}_{a(j)}^+ + {}^{a(j)}\mathbf{A}_j\, {}^{j}\mathbf{J}_j^+\, {}^{j}\mathbf{A}_{a(j)} - [{}^{a(j)}\hat{\mathbf{P}}_j\, {}^{a(j)}\hat{\mathbf{MS}}_j^+ + ({}^{a(j)}\hat{\mathbf{P}}_j\, {}^{a(j)}\hat{\mathbf{MS}}_j^+)^T] + {}^{a(j)}\hat{\mathbf{P}}_j\, {}^{a(j)}\hat{\mathbf{P}}_j^T\, M_j^+ \qquad [A8.1a]$$

$$^{a(j)}\mathbf{MS}_{a(j)}^+ = {}^{a(j)}\mathbf{MS}_{a(j)}^+ + {}^{a(j)}\mathbf{MS}_j^+ + {}^{a(j)}\mathbf{P}_j\, M_j^+ \qquad [A8.1b]$$

$$M_{a(j)}^+ = M_{a(j)}^+ + M_j^+ \qquad [A8.1c]$$

with:

- ${}^{a(j)}\mathbf{MS}_j^+ = {}^{a(j)}\mathbf{A}_j \, {}^j\mathbf{MS}_j^+$;
- $\hat{\mathbf{v}}$: (3x3) skew-symmetric matrix of the components of the vector $\mathbf{v}$;
- ${}^{a(j)}\mathbf{T}_j = \begin{bmatrix} {}^{a(j)}\mathbf{A}_j & {}^{a(j)}\mathbf{P}_j \\ \mathbf{0}_{3,1} & 1 \end{bmatrix}$;
- ${}^j\mathbf{J}_j^+$: inertia tensor of the composite link j^+ referred to frame R_j;
- ${}^j\mathbf{MS}_j^+$: first moments of the composite link j^+ referred to frame R_j;
- M_j^+: mass of the composite link j^+.

We note that equations [A8.1] are equivalent to the following:

$$ {}^{a(j)}\mathbf{K}_{a(j)}^+ = {}^{a(j)}\mathbf{K}_{a(j)}^* + {}^{a(j)}\boldsymbol{\lambda}_j \, {}^j\mathbf{K}_j^+ \qquad [A8.2] $$

where ${}^{a(j)}\boldsymbol{\lambda}_j$ and ${}^j\mathbf{K}_j$ are defined in Chapters 9 and 10 and Appendix 7.

NOTE.– The relationship between the concept of composite link parameters and base inertial parameters is considered in [Khalil 90a].

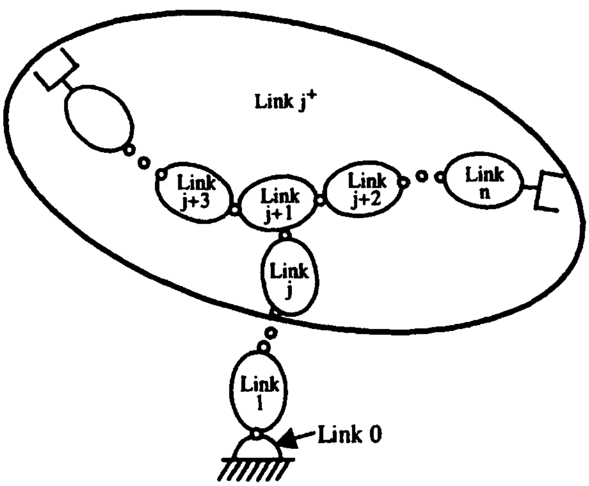

Figure A8.1. *Composite link j^+*

A8.2. Computation of the inertia matrix

We have seen in § 9.7.1 that the j^{th} column of the inertia matrix A can be computed by the Newton-Euler inverse dynamic algorithm by setting:

$$\ddot{\mathbf{q}} = \mathbf{u}_j,\ \dot{\mathbf{q}} = \mathbf{0},\ \mathbf{g} = \mathbf{0},\ \mathbf{F}_c = \mathbf{0}\ (\mathbf{f}_{ei} = \mathbf{0},\ \mathbf{m}_{ei} = \mathbf{0} \qquad \text{for } i = 1,\dots, n)$$

where $\mathbf{u}_j$ is an (nx1) vector with a 1 in the j^{th} row and zeros elsewhere.

Under these conditions, the forward recursive equations of the Newton-Euler inverse dynamic (Chapter 9) are only applied to link j^+:

$$^k\omega_k = \mathbf{0},\ ^k\dot{\omega}_k = \mathbf{0},\ ^k\dot{\mathbf{V}}_k = \mathbf{0},\ ^k\mathbf{F}_k = \mathbf{0},\ ^k\mathbf{M}_k = \mathbf{0} \quad \text{for } k < j \tag{A8.3}$$

$$^j\omega_j = \mathbf{0} \tag{A8.4}$$

$$^j\dot{\omega}_j = \bar{\sigma}_j\, {}^j\mathbf{a}_j \tag{A8.5}$$

$$^j\dot{\mathbf{V}}_j = \sigma_j\, {}^j\mathbf{a}_j \tag{A8.6}$$

$$^j\mathbf{F}_j = M_j^+\, {}^j\dot{\mathbf{V}}_j + {}^j\dot{\omega}_j \times {}^j\mathbf{MS}_j^+ \tag{A8.7}$$

$$^j\mathbf{M}_j = {}^j\mathbf{J}_j^+\, {}^j\dot{\omega}_j + {}^j\mathbf{MS}_j^+ \times {}^j\dot{\mathbf{V}}_j \tag{A8.8}$$

We deduce that:

– if joint j is prismatic ($^j\dot{\omega}_j = 0$, $^j\mathbf{M}_j = 0$ and $^j\dot{\mathbf{V}}_j = [\,0 \quad 0 \quad 1\,]^T$), then:

$$^j\mathbf{F}_j = [0 \quad 0 \quad M_j^+]^T \tag{A8.9}$$

$$^j\mathbf{M}_j = [MY_j^+ \quad -MX_j^+ \quad 0]^T \tag{A8.10}$$

– if joint j is revolute ($^j\dot{\mathbf{V}}_j = 0$ and $^j\dot{\omega}_j = [\,0 \quad 0 \quad 1\,]^T$), then:

$$^j\mathbf{F}_j = [-MY_j^+ \quad MX_j^+ \quad 0]^T \tag{A8.11}$$

$$^j\mathbf{M}_j = [XZ_j^+ \quad YZ_j^+ \quad ZZ_j^+]^T \tag{A8.12}$$

The recursive backward computation starts by link j and ends with link s, where a(s) = 0. The algorithm is given by the following equations:

– if joint j is prismatic, then:

$$^j\mathbf{f}_j = {}^j\mathbf{F}_j = [0 \quad 0 \quad M_j^+]^T \tag{A8.13}$$

$$^{j}\mathbf{m}_j = {}^{j}\mathbf{M}_j = [MY_j^+ \quad -MX_j^+ \quad 0]^T \qquad [A8.14]$$

$$A_{j,j} = M_j^+ + Ia_j \qquad [A8.15]$$

– if joint j is revolute, then:

$$^{j}\mathbf{f}_j = {}^{j}\mathbf{F}_j = [-MY_j^+ \quad MX_j^+ \quad 0]^T \qquad [A8.16]$$

$$^{j}\mathbf{m}_j = {}^{j}\mathbf{M}_j = [XZ_j^+ \quad YZ_j^+ \quad ZZ_j^+]^T \qquad [A8.17]$$

$$A_{j,j} = ZZ_j^+ + Ia_j \qquad [A8.18]$$

Then, the following equations are computed for $k = j, a(j), a(a(j)), ..., s$, where $a(s) = 0$:

$$^{a(k)}\mathbf{f}_{a(k)} = {}^{a(k)}\mathbf{A}_k \, {}^{k}\mathbf{f}_k \qquad [A8.19]$$

$$^{a(k)}\mathbf{m}_{a(k)} = {}^{a(k)}\mathbf{A}_k \, {}^{k}\mathbf{m}_k + ({}^{a(k)}\mathbf{P}_k \times {}^{a(k)}\mathbf{f}_{a(k)}) \qquad [A8.20]$$

$$\Gamma_{a(k)} = A_{a(k),j} = (\sigma_{a(k)} \mathbf{f}_{a(k)} + \bar{\sigma}_{a(k)} \mathbf{m}_{a(k)})^T \, {}^{a(k)}\mathbf{a}_{a(k)} \qquad [A8.21]$$

NOTES.–

- the element $A_{i,j}$ of the inertia matrix is set to zero if link i does not belong to the path between the base and link j;
- this algorithm provides the elements of the lower part of the inertia matrix. The other elements are deduced using the fact that the inertia matrix **A** is symmetric.