

RSD-TR-4-82

*CONNECTION BETWEEN FORMULATIONS  
OF ROBOT ARM DYNAMICS WITH  
APPLICATIONS TO SIMULATION  
AND CONTROL*

**J. L. Turney**

**T. N. Mudge**

**C. S. G. Lee**

**November 1981**

**Center for Robotics and Integrated Manufacturing**

**Robot System Division**

**COLLEGE OF ENGINEERING**

**THE UNIVERSITY OF MICHIGAN**

**ANN ARBOR, MICHIGAN 48109**



## TABLE OF CONTENTS

1. INTRODUCTION .....	2
2. NOTATION .....	5
3. EQUATION SETS .....	16
3.1. The Lagrange equations .....	16
3.2. The Recursive Lagrange .....	16
3.3. The Newton-Euler equations .....	18
3.4. The Gibbs-Appell set of equations .....	19
4. CONNECTION BETWEEN THE LAGRANGE AND NEWTON-EULER .....	21
5. CONNECTION BETWEEN RECURSIVE LAGRANGE AND NEWTON-EULER .....	27
6. CONNECTION BETWEEN GIBBS-APPELL AND NEWTON-EULER .....	29
7. IMPROVEMENTS TO NEWTON-EULER .....	31
8. COMPUTATIONAL COMPLEXITY COMPARISON .....	32
9. APPLICATION TO SIMULATION .....	40
10. APPLICATION TO CONTROL .....	42
10.1. Determining joint velocities and accelerations .....	42

10.2. Load adjustment trajectory planning .....	43
11.	
CONCLUSION .....	44
12.	
BIBLIOGRAPHY .....	45
APPENDIX .....	47

## ABSTRACT

In the following, the mathematical connection between several formulations for robot arm dynamics are exhibited. This provides one with a check of their consistency. The computational complexity of the formulations are then compared, followed by a study of the inherent parallelism of each formulation. Finally, a discussion of the application of the formulations to control and simulation is presented with an eye to an optimum approach for each of these applications.

## 1. INTRODUCTION

With the plethora of manipulator dynamics formulations now in publication [Pau72, Bej74, Lew74, LWP80, Hol80, Hot80], we felt it a necessary and useful exercise to exhibit the underlying mathematical connection between those of the most interest. We began this study with an exploration into the parallelism of the then existent dynamics formulations for the purpose of designing special purpose hardware to relieve the computational congestion in gross motion control [ISA paper]. In pursuing this goal we discovered inconsistencies in the Newton-Euler formulation [Wal78], the formulation of most interest. These inconsistencies were later removed in [LWP80]. Nevertheless, we achieved some insight into the optimality of a dynamics formulation by understanding the complete connection between formulations. We present here our original study along with comparisons with more recent formulations [Hol80, Hot80], together with our study of parallelism and a discussion of applications in control and simulation.

To begin with, an "arm" is defined to be an open chain of links driven at each joint by an actuator in a coordinated fashion to move the end-effector or "hand" link with multiple degrees of freedom along a prescribed trajectory.

Uicker [Uic67] used Lagrange formalism to derive a compact but complex set of equations for open link chains. Lewis [Lew74] (Note: The notation of Lewis is used here.) and others then applied these equations to robot arm dynamics. This compact equation set will be referred to as the "Lagrange" set in order to reflect the physical formalism involved in deriving this set of equations. These equations were the first to appear and although they are quite compact they are too complex for practical applications, being of the order  $n^3$ . (See Appendix A for a computational breakdown.)

Luh, Walker and Paul [LWP80] employ Newton's laws applied to rotating systems and obtain a computationally less complex set of equations than the Lagrange set, but at the sacrifice of losing the compact and well defined structure present in the Lagrange set. This equation set will be referred to as the "Newton-Euler" set.

Intermediate in complexity between the preceding two equation set is the equation set of Hollerbach. Hollerbach [Hol80] derived a recursive form of the Lagrange set which has roughly the efficiency of the Newton-Euler set but not the compactness of the Lagrange set of Lewis. This set will be referred to as the "recursive Lagrange."

Finally, Horowitz and Tomizuka [HoT80] use Gibbs Appell formalism to derive a set of equations whose complexity falls between the Lagrange and Newton-Euler set. They, however, did not propose to perform the actual computations. In their case the structure of the equations was obtained in order to parameterize the computation and allow adjustment of parameters by adaptive control. An examination will be made of this set, in any case, and it will be shown that it is a very close to the Newton-Euler set. This set will be referred to as the "Gibbs-Appell" set.

In the following we:

- (1) Introduce a consistent set of notation.
- (2) Display the form of all the equations of interest in this notation.
- (3) Develop the underlying connection between all the formulations mentioned above.
- (4) Discuss computational complexity and inherent parallelism of the above formulations.

- (5) Discuss applications of an "optimal" formulation to simulation and control.
- (6) Conclude with a summary of results.



## 1. NOTATION

The following set of notation is adopted to provide consistency between the formulations.

Matrices, and tensors will be represented in upper case type, while vectors will be in **boldface** type.

Greek indices will sometimes be used to denote components of a vector or matrix and the "summation convention" is employed, i.e. repeated indices are assumed summed over all three coordinates. For example the inner product between vectors **a** and **b** can be written in two ways, in matrix notation,  $\mathbf{a}^t \mathbf{b}$  and in terms of the vector using the summation convention,  $a_\gamma b_\gamma$ . The product of two matrices A and B, C, is  $C = AB$  in matrix notation and  $C_{\alpha\beta} = A_{\alpha\gamma} B_{\gamma\beta}$  in component notation with the summation convention.

$R_j^i$  represents a three by three rotation matrix which maps a vector from its representation in the  $i^{\text{th}}$  link coordinate frame to its equivalent in the  $j^{\text{th}}$  coordinate frame. Some well known properties of rotation matrices represented in this notation are:

$$(R_j^i)^t = (R_j^i)^{-1} = R_j^i \quad 1.1$$

A superscripted t, i.e. notation  $( )^t$ , denotes a transpose.

A rotation between coordinate frames i and j can be written as a chain product of rotations between successive frames:

$$R_j^i = R_j^{j+1} R_{j+1}^{j+2} \cdots R_{i-1}^i \quad 1.2$$

In general, with the inverse defined by Eqn. 1.1, one obtains the relation  $R_j^k R_k^i = R_j^i$  for all integer values of k. Further define  $R_i^i = E$ , the identity, for consistency. Notice E has been used as the identity matrix rather than the more usual "I". "I" is used later to denote the inertial tensor.

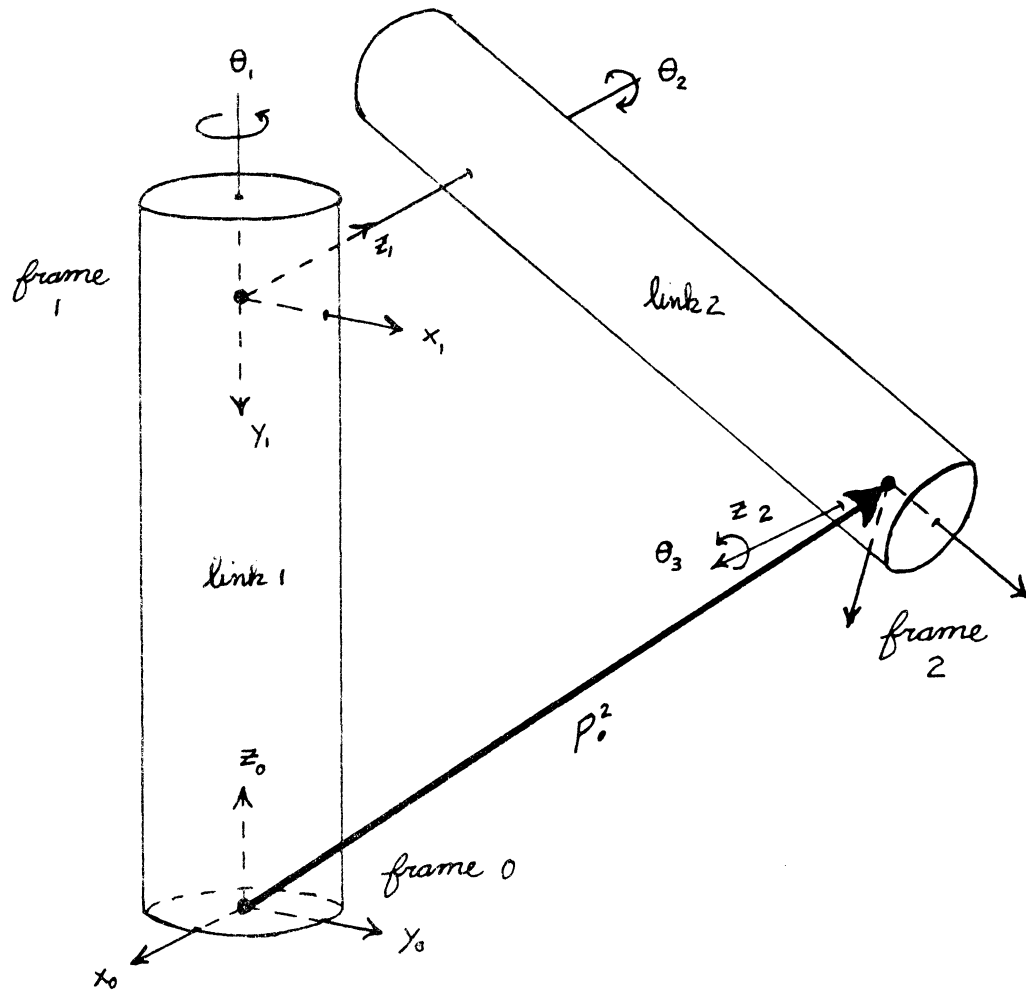


Figure 2.1

Each link,  $i$ , of the arm will have its own coordinate frame fixed in the  $i^{\text{th}}$  link and referred to as the  $i^{\text{th}}$  frame as pictured in Fig. 2.1.

A unit vector along the  $z$  axis of the  $i^{\text{th}}$  frame and represented in the  $i^{\text{th}}$  frame will be denoted by  $z_i$ . The same unit vector may be represented with respect to the base ( $0^{\text{th}}$ ) frame by applying a rotation, i.e.  $R_0^i z_i$ , but to simplify notation, such vectors are starred to indicate that they have been rotated into a base frame representation, i.e.  $R_0^i z_i^*$ . The lower index indicates the fixed

frame to which the vector originally belongs.

Rotations operate on a vector product in the following fashion:

$$R(\mathbf{b} \times \mathbf{c}) = R\mathbf{b} \times R\mathbf{c}, \quad 2.3$$

where  $\mathbf{b}$  and  $\mathbf{c}$  are any vectors, since a vector product must itself transform as a vector under rotation.

One often encounters expressions of the form:

$$R_k^i(\mathbf{z}_i \times \mathbf{c}_i) = R_k^i \mathbf{z}_i \times R_k^i \mathbf{c}_i, \quad 2.4$$

where  $\mathbf{z}_i$ , as before, is a unit vector in the  $z$  direction of the  $i^{\text{th}}$  frame, and  $\mathbf{c}_i$  is also a vector in the  $i^{\text{th}}$  frame. In order to simplify this above frequently occurring expression, define a matrix:

$$Q^z = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad 2.5$$

which is the matrix representation of the above cross product, i.e.:

$$Q^z \mathbf{c}_i = \mathbf{z}_i \times \mathbf{c}_i, \quad 2.6$$

when  $\mathbf{c}_i$  is represented as a column vector with a basis in the  $i^{\text{th}}$  frame.  $Q^z$  is actually a matrix often used in mechanics when dealing with angular momentum. The  $Q^\beta$  matrices have the property that  $Q^x \mathbf{c}_i = \mathbf{x}_i \times \mathbf{c}_i$ ,  $Q^y \mathbf{c}_i = \mathbf{y}_i \times \mathbf{c}_i$ , and  $Q^z \mathbf{c}_i = \mathbf{z}_i \times \mathbf{c}_i$  as shown above in Eqn. 2.6. One can capture the action of a vector cross product in a matrix operation. The  $Q^\beta$  matrices are listed below:

$$Q^x = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \quad Q^y = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \quad Q^z = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad 2.7$$

The  $Q^\beta$  are just components of the Levi Cevita tensor. Using this notation the cross product between two vectors can be written in matrix notation as:

$$(\mathbf{b} \times \mathbf{c}) = [(b_x Q^x + b_y Q^y + b_z Q^z) \mathbf{c}] = b_\beta Q^\beta \mathbf{c} \quad 2.8$$

Also notice another property of the Q matrices:

$$Q^x Q^y = Q^y Q^x + Q^z, \text{ and cyclic,} \quad 2.9$$

or in component notation.

$$Q_{\alpha\gamma}^\beta Q_{\gamma\chi}^\eta = \delta_{\alpha\eta} \delta_{\beta\chi} - \delta_{\alpha\chi} \delta_{\beta\eta}, \quad 2.10$$

where  $\delta$  is the Kronecker delta. This equation can be shown by explicit Q multiplication.

From Eqns. 2.3 and 2.8 there is a useful way of commuting rotation and Q' operator:

$$R_k^i(\mathbf{b}_i)_\beta Q^\beta \mathbf{c}_i = R_k^i(\mathbf{b}_i \times \mathbf{c}_i) = R_k^i \mathbf{b}_i \times R_k^i \mathbf{c}_i = (R_k^i \mathbf{b}_i)_\beta Q^\beta (R_k^i \mathbf{c}_i) \quad 2.11$$

Confining ourselves to arms with links connected in the fashion of Denavit and Hartenberg [DeH55], where all relative joint rotations of the  $i^{\text{th}}$  link occur about the  $\mathbf{z}_{i-1}$  axis, Fig. 2.1. In this case, matrices,  $R_{i-1}^i$ , have the form:

$$R_{i-1}^i = \begin{bmatrix} \cos\vartheta_i & -\cos\varphi_i \sin\vartheta_i & \sin\varphi_i \sin\vartheta_i \\ \sin\vartheta_i & \cos\varphi_i \cos\vartheta_i & -\sin\varphi_i \cos\vartheta_i \\ 0 & \sin\varphi_i & \cos\varphi_i \end{bmatrix}, \quad 2.12$$

where  $\vartheta_i$  is the relative joint angle between links  $i$  and  $i-1$  (Fig. 1) and  $\varphi_i$  is a fixed structural angle which allows successive coordinate frames to be set up so that joint rotations always occur about the  $\mathbf{z}$  axis of the previous link. For example, in Fig. 2.1, a  $\vartheta_1$  rotation about the  $\mathbf{z}_0$  axis aligns the  $\mathbf{x}_1$  and  $\mathbf{x}_0$  axis while the fixed rotation of  $\varphi_1 = \frac{\pi}{2}$  about the  $\mathbf{x}_1$  axis brings the  $\mathbf{z}_1$  axis into coincidence with the  $\mathbf{z}_0$  axis. ( Note that the  $\mathbf{x}_i$  axis is always chosen perpendicular to both the  $\mathbf{z}_{i-1}$  and the  $\mathbf{z}_i$  axes.)

It can be shown that:

$$\frac{\partial R_{i-1}^i}{\partial \vartheta_j} \mathbf{c}_i = Q^z R_{i-1}^i = \mathbf{z}_{i-1} \times R_{i-1}^i \mathbf{c}_i, \quad 2.13$$

by comparing the derivative to the matrix product, and hence from Eqn. 2.2:

$$\frac{\partial R_0^i}{\partial \vartheta_j} \mathbf{c}_1 = \frac{\partial R_0^{i-1}}{\partial \vartheta_j} R_{j-1}^i = R_0^{i-1} Q^z R_{j-1}^i \mathbf{c}_1 = R_0^{i-1} (\mathbf{z}_{j-1} \times R_{j-1}^i \mathbf{c}_1) \quad 2.14$$

where  $j \leq k$ , and hence also:

$$\frac{\partial^2 R_0^i}{\partial \vartheta_j \partial \vartheta_k} \mathbf{c}_1 = R_0^{u-1} Q^z R_{u-1}^{v-1} Q^z R_{v-1}^i \mathbf{c}_1 = R_0^{u-1} (\mathbf{z}_{u-1} \times (R_{u-1}^{v-1} \mathbf{z}_{v-1} \times R_{v-1}^i \mathbf{c}_1)), \quad 2.15$$

where  $u = \min(j, k)$  and  $v = \max(j, k)$ . Thus, differentiation is reduced to matrix multiplication or a cross product.

Denavit and Hartenberg [DeH55] introduced a matrix,  $T_{i-1}^i$ , which expresses both the rotation and translation necessary to map a position vector in the  $i^{\text{th}}$  frame to its equivalent in a displaced  $i-1^{\text{th}}$  frame. The notation of [Lew74] is used.  $T_{i-1}^i$  operates on an augmented form of a vector  $\mathbf{d}_i$  in the  $i^{\text{th}}$  frame given by:

$$\mathbf{d}_i^a = \begin{bmatrix} \mathbf{d}_i \\ 1 \end{bmatrix}$$

and the matrix  $T_{i-1}^i$  is given by:

$$T_{i-1}^i = \begin{bmatrix} R_{i-1}^i & \mathbf{p}_{i-1}^i \\ 0 & 1 \end{bmatrix} \quad 2.16$$

The position pointed to by  $\mathbf{d}_i^a$  in the displaced  $i-1^{\text{th}}$  frame is given by:

$$T_{i-1}^i \mathbf{d}_i^a$$

The submatrix,  $R_{i-1}^i$ , is just the rotation matrix discussed above, and  $\mathbf{p}_{i-1}^i$  is the displacement of the  $i^{\text{th}}$  origin from the  $i-1^{\text{th}}$  origin viewed in the  $i-1^{\text{th}}$  frame. A similar vector describing the same displacement, but viewed in the  $i^{\text{th}}$  frame would be  $R_{i-1}^{i-1} \mathbf{p}_{i-1}^i$ . To be consistent with the notation of [LWP80] define this displacement between the  $i^{\text{th}}$  and  $i-1^{\text{th}}$  frames as viewed in the  $i^{\text{th}}$  frame as  $\mathbf{r}_i$  (Fig. 2.2), hence one has:

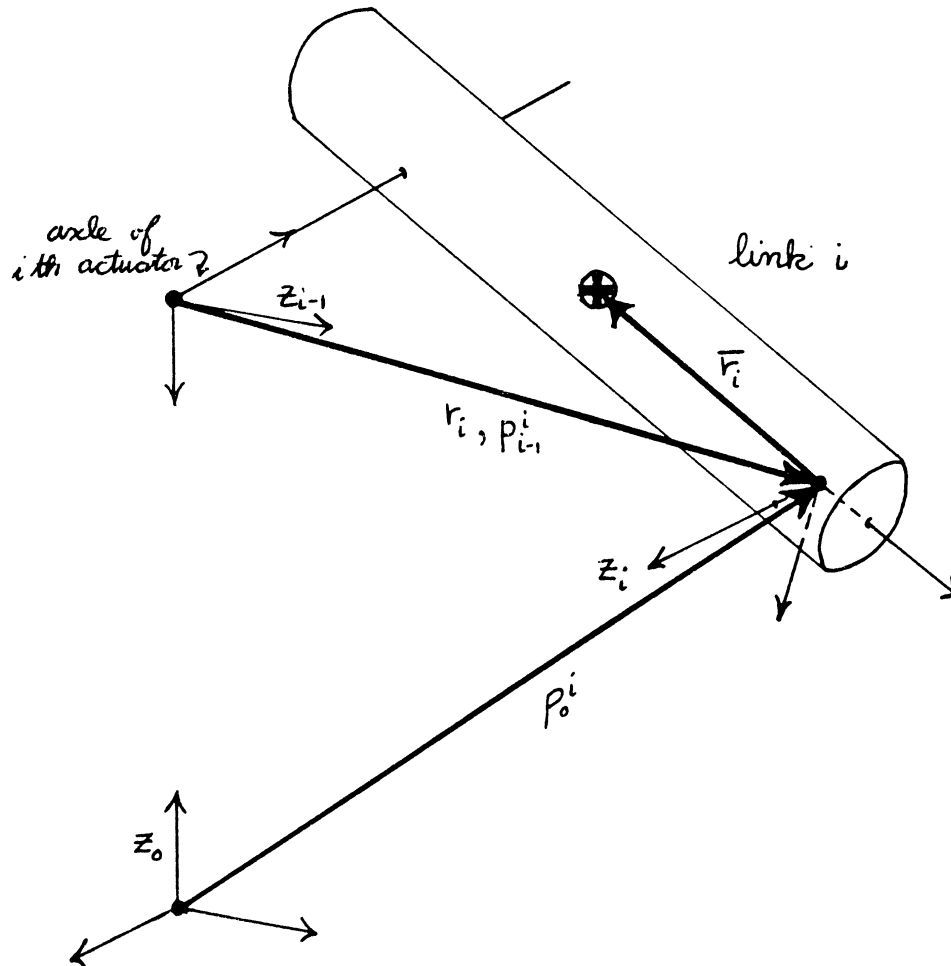


Figure 2.2

$$R_i^{i-1} \mathbf{p}_{i-1}^i = \mathbf{r}_i \quad 2.17$$

The  $T_{i-1}^i$  matrices can be chained in the manner of Eqn. 2.1 to obtain:

$$\begin{aligned} T_j^i &= T_j^{j+1} T_{j+1}^{j+2} \cdots T_{i-1}^i \\ &= \begin{bmatrix} R_j^i & \mathbf{p}_j^i \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_j^{j+1} & \mathbf{p}_j^{j+1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_{j+1}^{j+2} & \mathbf{p}_{j+1}^{j+2} \\ 0 & 1 \end{bmatrix} \cdots \begin{bmatrix} R_{i-1}^i & \mathbf{p}_{i-1}^i \\ 0 & 1 \end{bmatrix} \end{aligned} \quad 2.18$$

in which case it can be shown by multiplication of the R and  $\mathbf{p}$  submatrices of T

together with Eqns. 2.2 and

that the submatrix  $\mathbf{p}_j^i$  of  $T_j^i$  can be written:

$$\mathbf{p}_j^i = \sum_{m=j+1}^i R_j^{m-1} \mathbf{p}_{m-1}^m = \sum_{m=j+1}^i R_j^m R_m^{m-1} \mathbf{p}_{m-1}^m = \sum_{m=j+1}^i R_j^m \mathbf{r}_m. \quad 2.19$$

where  $j < i$ . Thus, the position vector  $\mathbf{p}_j^i$  from the  $j^{\text{th}}$  origin to the  $i^{\text{th}}$  origin is composed of a chain of vectors fixed in intermediate links. Furthermore, it can be seen that:

$$T_j^i \mathbf{d}_i^a = T_j^i \begin{bmatrix} \mathbf{d}_i \\ 1 \end{bmatrix} = \begin{bmatrix} R_j^i & \mathbf{p}_j^i \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{d}_i \\ 1 \end{bmatrix} = \begin{bmatrix} R_j^i \mathbf{d}_i + \mathbf{p}_j^i \\ 1 \end{bmatrix}, \quad 2.20$$

i.e. the position pointed to by vector  $\mathbf{d}_i$  in the  $i^{\text{th}}$  frame can be determined in the  $j^{\text{th}}$  frame by rotating the  $\mathbf{d}_i$  position vector into the  $j^{\text{th}}$  frame ( $R_j^i \mathbf{d}_i$ ) and adding a frame displacement,  $\mathbf{p}_j^i$ . For convenience define a vector from the  $j^{\text{th}}$  frame to the  $i^{\text{th}}$  center of mass denoted,  $\bar{\mathbf{p}}_j^i$ .

From Eqn. 2.19 one obtains:

$$\frac{\partial \mathbf{p}_0^q}{\partial \vartheta_1} = \frac{\partial R_0^i}{\partial \vartheta_1} \mathbf{p}_1^q = \sum_{m=1}^i \frac{\partial R_0^m}{\partial \vartheta_1} \mathbf{r}_m \quad 2.21$$

Replacing the partial derivatives using Eqn. 2.14 and Eqn. 2.15 one obtains:

$$\frac{\partial \mathbf{p}_0^q}{\partial \vartheta_1} = \sum_{m=j}^i R_0^{j-1} Q^z R_{j-1}^m \mathbf{r}_m \quad 2.22$$

$$= \sum_{m=j}^i R_0^{j-1} \mathbf{z}_{j-1} \times R_{j-1}^m \mathbf{r}_m = R_0^{j-1} \mathbf{z}_{j-1} \times \bar{\mathbf{p}}_1^q \quad 2.23$$

and similarly:

$$\begin{aligned} \frac{\partial^2 \mathbf{p}_0^i}{\partial \vartheta_j \partial \vartheta_k} &= \sum_{m=k}^i R_0^{u-1} Q^z R_{u-1}^{v-1} Q^z R_{v-1}^m \mathbf{r}_m \\ &= \sum_{m=k}^i R_0^{u-1} \mathbf{z}_{u-1} \times (R_{u-1}^{v-1} (\mathbf{z}_{v-1} \times R_{v-1}^m \mathbf{r}_m)), \end{aligned} \quad 2.24$$

where  $u = \min(j, k)$  and  $v = \max(j, k)$ .

The notation,  $\mathbf{a}^t\mathbf{b}$ , will be used to denote the dot product between vectors  $\mathbf{a}$  and  $\mathbf{b}$ , while the notation,  $\mathbf{ab}^t$ , the outer product, is equivalent to a vector dyadic, in particular,  $(\mathbf{ab}^t)\mathbf{c}=\mathbf{a}(\mathbf{b}^t\mathbf{c})$ . The following vector and matrix identities are used:

$$(\mathbf{a} \times \mathbf{b})^t\mathbf{c}=\mathbf{a}^t(\mathbf{b} \times \mathbf{c}), \quad 2.25$$

whereas the dot and cross product can be interchanged.

$$\text{Tr} \{\mathbf{ab}^t\}=\mathbf{b}^t\mathbf{a} \quad 2.26$$

The trace of an outer product of two vectors is the inner product of the two vectors.

$$(\mathbf{a} \times \mathbf{b}) \times \mathbf{c}=\mathbf{a} \times (\mathbf{b} \times \mathbf{c})-\mathbf{b} \times (\mathbf{a} \times \mathbf{c}), \quad 2.27$$

a simple cross product relation. Another relation which appears several times is:

$$\sum_{j=1}^q \sum_{k=1}^q \mathbf{b}_u \times (\mathbf{b}_v \times \mathbf{c}_w)=\sum_{j=1}^q \sum_{k=1}^{j-1} \mathbf{b}_k \times (\mathbf{b}_j \times \mathbf{c}_w)+\sum_{j=1}^q \sum_{k=j}^q \mathbf{b}_j \times (\mathbf{b}_k \times \mathbf{c}_w), \quad 2.27$$

where  $u=\min(j,k)$  and  $v=\max(j,k)$ , and from this implies:

$$\sum_{j=1}^q \sum_{k=1}^q \mathbf{b}_u \times (\mathbf{b}_v \times \mathbf{c}_w)=\sum_{j=1}^q \sum_{k=1}^{j-1} (\mathbf{b}_k \times \mathbf{b}_j) \times \mathbf{c}_w+\sum_{j=1}^q \sum_{k=1}^q \mathbf{b}_j \times (\mathbf{b}_k \times \mathbf{c}_w) \quad 2.28$$

$$\text{Tr} \{ABC\}=\text{Tr} \{(ABC)^t\}=\text{Tr} \{C^tB^tA^t\} \quad 2.29$$

Also,

$$\text{Tr} \{ABC\}=\text{Tr} \{CAB\} \quad 2.30$$

Finally, some notation for the Lagrangian formulation concerning the link inertial tensors is introduced. Consider link  $q$  in Fig. 2.3. If one integrates the infinitesimal mass  $dm$  times the outer product  $\mathbf{d}_q\mathbf{d}_q^t$  over the entire link mass, one obtains an inertial type matrix,  $J_q$  defined by:



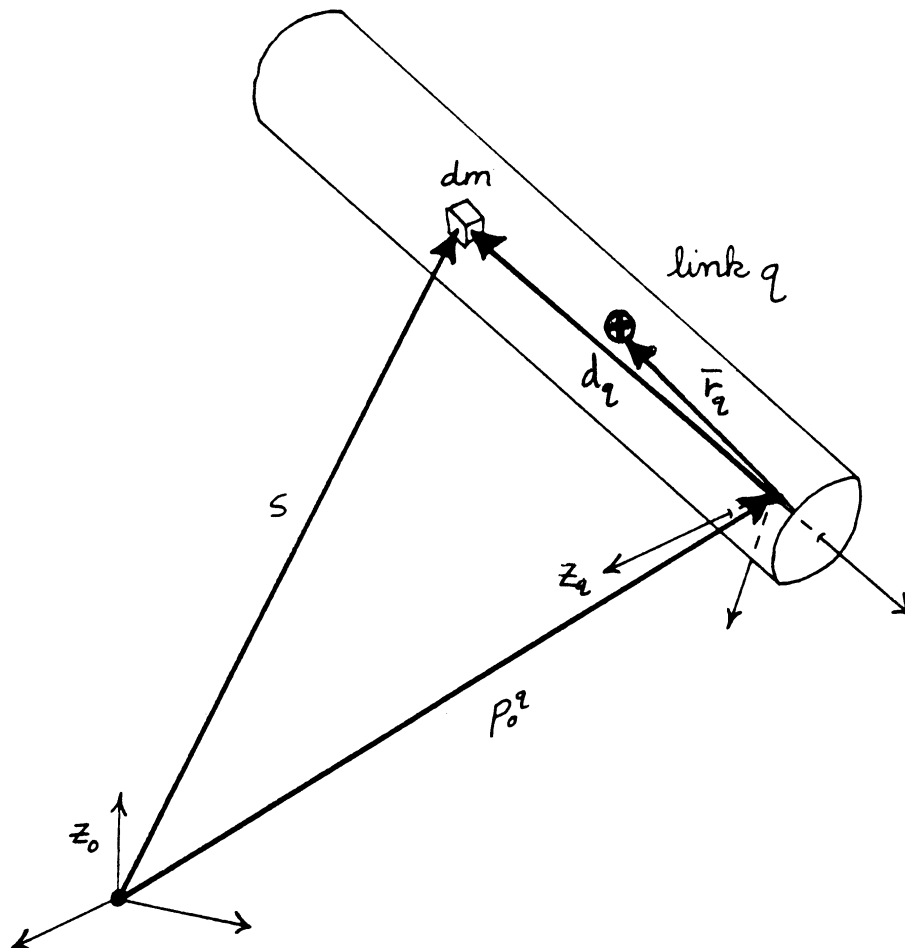


Figure 2.3

$$J_q = \int \mathbf{d}_q \mathbf{d}_q^T dm, \quad 2.31$$

or

$$J_q = \begin{bmatrix} \int d_{q_x}^2 dm & \int d_{q_x} d_{q_y} dm & \int d_{q_x} d_{q_z} dm \\ \int d_{q_y} d_{q_x} dm & \int d_{q_y}^2 dm & \int d_{q_y} d_{q_z} dm \\ \int d_{q_z} d_{q_x} dm & \int d_{q_z} d_{q_y} dm & \int d_{q_z}^2 dm \end{bmatrix} \quad 2.32$$

The integrals above are taken about the upper end of the link. Normally

inertias are not specified in this fashion but are taken about the center of mass. Using the parallel axis theorem [Sym71],  $J_q$  can be rewritten in terms of the link center of mass inertial matrix,  $I_q$ , and the center of mass vector,  $\bar{r}_q$ , shown in Fig. 2.3, as below:

$$J_q = \begin{bmatrix} \frac{-I_{q_{xx}} + I_{q_{yy}} + I_{q_{zz}}}{2} + m_q \bar{r}_{q_z}^2 & m_q \bar{r}_{q_x} \bar{r}_{q_y} & m_q \bar{r}_{q_x} \bar{r}_{q_z} \\ m_q \bar{r}_{q_y} \bar{r}_{q_x} & \frac{I_{q_{xx}} - I_{q_{yy}} + I_{q_{zz}}}{2} + m_q \bar{r}_{q_y}^2 & m_q \bar{r}_{q_y} \bar{r}_{q_z} \\ m_q \bar{r}_{q_z} \bar{r}_{q_x} & m_q \bar{r}_{q_z} \bar{r}_{q_y} & \frac{I_{q_{xx}} + I_{q_{yy}} - I_{q_{zz}}}{2} + m_q \bar{r}_{q_z}^2 \end{bmatrix}, \quad 2.33$$

where it is assumed that a principle set of inertial axis can be found by a simple translation from the  $q^{\text{th}}$  origin to the  $q^{\text{th}}$  center of mass. Note that if the inertial is integrated about the center of mass one has a simpler inertial matrix,  $K_q$ :

$$K_q = \begin{bmatrix} \frac{I_{q_{xx}} + I_{q_{yy}} + I_{q_{zz}}}{2} & 0 & 0 \\ 0 & \frac{I_{q_{xx}} - I_{q_{yy}} + I_{q_{zz}}}{2} & 0 \\ 0 & 0 & \frac{I_{q_{xx}} + I_{q_{yy}} - I_{q_{zz}}}{2} \end{bmatrix}, \quad 2.34$$

where  $K_q$  and  $J_q$  are related by:

$$K_q = J_q - m_q \bar{r}_q^t \bar{r}_q \quad 2.35$$

Notice that:

$$I_q = K_q - \text{Tr } K_q \quad 2.36$$

To be consistent with [LWP80] define an augmented matrix,  $J_q^a$ , which has the form:

$$J_q^a = \begin{bmatrix} J_q & m_q \bar{r}_q \\ m_q \bar{r}_q^t & m_q \end{bmatrix},$$

2.37

where  $m_q$  is the  $q^{\text{th}}$  link mass.

### 3. EQUATION SETS

The four equation sets under discussion will be introduced below.

#### 3.1. The Lagrange equations

The first formulation for an arm chronologically was the Lagrange equation set. The torque,  $\tau_i$ , is related to the relative joint angular accelerations,  $\ddot{\vartheta}_j$ , and velocities,  $\dot{\vartheta}_j$ , through a compact  $\vartheta$  dependent relation. The equation for  $\tau_i$  is:

$$\tau_i = \sum_{q=1}^n \sum_{j=1}^q \left[ \text{Tr} \left\{ \frac{\partial T_\sigma^q}{\partial \vartheta_j} J_q^a \left( \frac{\partial T_\sigma^q}{\partial \vartheta_i} \right)^t \right\} \ddot{\vartheta}_j + \sum_{k=1}^q \text{Tr} \left\{ \frac{\partial^2 T_\sigma^q}{\partial \vartheta_j \partial \vartheta_k} J_q^a \left( \frac{\partial T_\sigma^q}{\partial \vartheta_i} \right)^t \right\} \dot{\vartheta}_j \dot{\vartheta}_k \right] + \sum_{q=1}^n g(\mathbf{z}_\sigma^q)^t \frac{\partial T_\sigma^q}{\partial \vartheta_i} m_q \bar{\mathbf{r}}_q^a \quad 3.1$$

All of the notation has already been introduced in Sec. 2, with possibly the exception of the  $g$  term, which is just the gravitational constant 9.8 m/s. A development of these equations may be found in [Lew74] [TML80].

The equations are compact but are computationally of the order,  $n^3$  (See Appendix A), where  $n$  is the number of links.

#### 3.2. The Recursive Lagrange

Hollerbach [Hol80] has derived a iterative form of the Lagrange equations stated below:

$$\tau_i = \text{Tr} \left\{ \frac{\partial R_\sigma^i}{\partial \vartheta_i} D_i \right\}, \quad 3.2$$

where:

$$D_i = R_i^{i+1} D_{i+1} + r_i e_{i+1} + J_i \left( \frac{d^2 R_0^i}{dt^2} \right)^t + m_i \bar{r}_i \left( \frac{d^2 R_0^i}{dt^2} \bar{r}_i \right)^t + m_i (\bar{r}_i + r_i) \left( \frac{d^2 p_0^i}{dt^2} \right)^t + m_i r_i \left( \frac{d^2 R_0^i}{dt^2} \bar{r}_i \right)^t, \quad 3.3$$

and where:

$$e_i = \sum_{q=i}^n \left( m_q \frac{d^2 p_0^q}{dt^2} + m_i \frac{d^2 R_0^q}{dt^2} \bar{r}_q \right) = e_{i+1} + m_i \frac{d^2 p_0^i}{dt^2} + m_i \frac{d^2 R_0^i}{dt^2} \bar{r}_i, \quad 3.4$$

and where the derivatives of  $R_0^i$  and  $p_0^i$  are found iteratively as below:

$$\frac{d^2 R_0^i}{dt^2} = \left( \frac{d^2 R_0^{i-1}}{dt^2} + 2 \frac{d R_0^{i-1}}{dt} Q^z \dot{\vartheta}_j + R_0^{i-1} \left( (Q^z)^2 \dot{\vartheta}_j^2 + Q^z \ddot{\vartheta}_j \right) \right) R_{j-1}^i, \quad 3.5$$

and:

$$\frac{d R_0^i}{dt} = \left( \frac{d R_0^{i-1}}{dt} + R_0^{i-1} Q^z \dot{\vartheta}_j \right) R_{j-1}^i \quad 3.6$$

(Note: There were some errors in the original formulation of  $D_i$  which have been corrected here.)

These equations are linear in  $n$  but take more computations than the following Newton-Euler set, chiefly because they are still formulated in terms of matrices, instead of vectors, and transforming vectors between coordinate frames is computationally less expensive than transforming matrices. In the Newton-Euler case below one can take advantage of this fact.

**3.3. The Newton-Euler equations** The Newton-Euler equations are less compact, but are computationally linear in the number of links,  $n$ . (See Sec. 6 for a comparison).

The Lagrangian approach allows the formulation of the solution to problems in dynamics in an "automatic" way. However, this ease of formulation is obtained at the expense of physical insight into the problem. In particular, it is often not possible to identify calculations that have little contribution to the value of the solution. With the Newton-Euler formulation these two terms can be identified and eliminated.

Now for an overview. In the Newton-Euler formulation one works from the base to the hand determining kinematic terms of the links and passing them up in a causal fashion. Then one works from the hand to the base determining dynamic terms and passing them down in a causal fashion. One would assume this technique might be the most efficient and this assumption appears to be the case.

$$\omega_0=0$$

$$\alpha_0=0$$

$$\mathbf{a}_0=9.8 \text{ m/s}^2$$

$$\omega_j=R_j^{j-1}(\omega_{j-1}+\dot{\vartheta}_j \mathbf{z}_{j-1}) \quad 3.7$$

$$\alpha_j=R_j^{j-1}(\alpha_{j-1}+\omega_j \times \mathbf{z}_{j-1} \dot{\vartheta}_j+\ddot{\vartheta}_j \mathbf{z}_{j-1}) \quad 3.8$$

$$\mathbf{a}_j=\omega_j \times (\omega_j \times \mathbf{r}_j)+\alpha_j \times \mathbf{r}_j+R_j^{j-1} \mathbf{a}_{j-1} \quad 3.9$$

$$\bar{\mathbf{a}}_j=\omega_j \times (\omega_j \times \bar{\mathbf{r}}_j)+\alpha_j \times \bar{\mathbf{r}}_j+\mathbf{a}_j \quad 3.10$$

$$\mathbf{f}_j=m_j \bar{\mathbf{a}}_j+R_j^{j+1} \mathbf{f}_{j+1} \quad 3.11$$

$$\mathbf{n}_i = I_i \boldsymbol{\alpha}_i + \boldsymbol{\omega}_i \times (I_i \boldsymbol{\omega}_i) + m_i (\bar{\mathbf{r}}_i + \mathbf{r}_i) \times \bar{\mathbf{a}}_i + \mathbf{r}_i \times R_i^{i+1} \mathbf{f}_{i+1} + R_i^{i+1} \mathbf{n}_{i+1} \quad 3.12$$

$$\tau_i = (R_i^{i-1} \mathbf{z}_{i-1})^t \mathbf{n}_i, \quad 3.13$$

$$\mathbf{f}_{n+1} = 0$$

$$\mathbf{n}_{n+1} = 0$$

All the vectors above are represented in the coordinate frame of their lower index. Notice the  $I_i$ 's,  $\mathbf{r}_i$ 's, and  $\bar{\mathbf{r}}_i$ 's are represented in the  $i^{\text{th}}$  frame and hence constant. There was some confusion caused by the notation of the paper by Luh, Walker, and Paul [LWP80]. In their paper Luh et al represented all vectors and inertias with respect to the base, and then rotated the final inertias and vectors into the link systems. The  $I$ 's etc. looked as if they had to be rotated into the base system before being used. That, however, is not the case.

### 3.4. The Gibbs-Appell set of equations

The Gibbs-Appell set will have roughly the same computational complexity as the Lagrange set. It is presented for completeness.

$$\tau_i = \sum_{q=i}^n \left[ \sum_{j=1}^q \left[ \mathbf{z}_{i-1}^{\dot{t}} I_q^{\dot{t}} \mathbf{z}_{j-1}^{\dot{t}} + [\mathbf{z}_{i-1}^{\dot{t}} \times \bar{\mathbf{p}}_i^{\dot{q}}]^t [\mathbf{z}_{j-1}^{\dot{t}} \times \bar{\mathbf{p}}_j^{\dot{q}}] \ddot{\vartheta}_j \right. \right. \quad 3.14$$

$$\left. \left. \sum_{k=1}^q \sum_{j=1}^q \mathbf{z}_{i-1}^{\dot{t}} \left[ \mathbf{z}_{u-1}^{\dot{t}} \times I_q^{\dot{t}} \mathbf{z}_{v-1}^{\dot{t}} + \mathbf{z}_{v-1}^{\dot{t}} \times \mathbf{z}_{u-1}^{\dot{t}} \frac{1}{2} \text{Tr}(I_q^{\dot{t}}) \right. \right. \right. \\ \left. \left. \left. + (\mathbf{z}_{i-1}^{\dot{t}} \times \bar{\mathbf{p}}_i^{\dot{q}})^t \mathbf{z}_{u-1}^{\dot{t}} \times (\mathbf{z}_{v-1}^{\dot{t}} \times \bar{\mathbf{p}}_j^{\dot{q}}) \right] \right] \ddot{\vartheta}_j \ddot{\vartheta}_k ,$$

$u = \min(j, k)$  and  $v = \max(j, k)$ .

As mentioned before the Gibbs-Appell set was derived to determine the structure of the equation for  $\tau_i$ . It was not intended to compete computationally with the Newton-Euler set.



#### 4. CONNECTION BETWEEN THE LAGRANGE AND NEWTON-EULER

$$\begin{aligned} \tau_i = & \sum_{q=1}^n \sum_{j=1}^a \left[ \text{Tr} \left\{ \frac{\partial T_q^a}{\partial \vartheta_j} J_q^a \left( \frac{\partial T_q^a}{\partial \vartheta_1} \right)^t \right\} \ddot{\vartheta}_j + \sum_{k=1}^a \text{Tr} \left\{ \frac{\partial^2 T_q^a}{\partial \vartheta_j \partial \vartheta_k} J_q^a \left( \frac{\partial T_q^a}{\partial \vartheta_1} \right)^t \right\} \dot{\vartheta}_j \dot{\vartheta}_k \right] \\ & + \sum_{q=1}^n g(\mathbf{z}_0^a)^t \frac{\partial T_q^a}{\partial \vartheta_1} m_q \bar{\mathbf{r}}_q^a \end{aligned} \quad 4.1$$

The Lagrange formulation when expanded by submatrix multiplication of the  $T_q^a$ 's and  $J_q^a$  and can be written as below:

$$\begin{aligned} \tau_i = & \sum_{q=1}^n \left[ \sum_{j=1}^a \text{Tr} \left\{ \left( \frac{\partial R_q^a}{\partial \vartheta_j} J_q + m_q \frac{\partial \mathbf{p}_q^a}{\partial \vartheta_j} \bar{\mathbf{r}}_q^t \right) \left( \frac{\partial R_q^a}{\partial \vartheta_1} \right)^t \right. \right. \\ & \left. \left. + \left( m_q \frac{\partial R_q^a}{\partial \vartheta_j} \bar{\mathbf{r}}_q + m_q \frac{\partial \mathbf{p}_q^a}{\partial \vartheta_j} \right) \left( \frac{\partial \mathbf{p}_q^a}{\partial \vartheta_1} \right)^t \right\} \ddot{\vartheta}_j \right. \\ & \left. + \sum_{k=1}^a \text{Tr} \left\{ \left( \frac{\partial^2 R_q^a}{\partial \vartheta_j \partial \vartheta_k} J_q + m_q \frac{\partial^2 \bar{\mathbf{p}}_q^a}{\partial \vartheta_j \partial \vartheta_k} \bar{\mathbf{r}}_q^t \right) \left( \frac{\partial R_q^a}{\partial \vartheta_1} \right)^t \right. \right. \\ & \left. \left. + \left( \frac{\partial^2 R_q^a}{\partial \vartheta_j \partial \vartheta_k} m_q \bar{\mathbf{r}}_q + \frac{\partial^2 \bar{\mathbf{p}}_q^a}{\partial \vartheta_j \partial \vartheta_k} m_q \right) \left( \frac{\partial \mathbf{p}_q^a}{\partial \vartheta_1} \right)^t \right\} \dot{\vartheta}_j \dot{\vartheta}_k \right. \\ & \left. + (m_i g \mathbf{z}_0)^t \left( \frac{\partial R_q^a}{\partial \vartheta_1} \bar{\mathbf{r}}_q + \frac{\partial \mathbf{p}_q^a}{\partial \vartheta_1} \right) \right] \end{aligned} \quad 4.2$$

Defining a inertial tensor according to Eqn. 2.35, one can rewrite this as:

$$\begin{aligned} \tau_i = & \sum_{q=1}^n \sum_{j=1}^a \left[ \text{Tr} \left\{ \left( \frac{\partial R_q^a}{\partial \vartheta_j} K_q \right) \left( \frac{\partial R_q^a}{\partial \vartheta_1} \right)^t + \left( m_q \frac{\partial \bar{\mathbf{p}}_q^a}{\partial \vartheta_j} \right) \left( \frac{\partial \bar{\mathbf{p}}_q^a}{\partial \vartheta_1} \right)^t \right\} \ddot{\vartheta}_j \right. \\ & \left. + \sum_{k=1}^a \text{Tr} \left\{ \left( \frac{\partial^2 R_q^a}{\partial \vartheta_j \partial \vartheta_k} K_q \right) \left( \frac{\partial R_q^a}{\partial \vartheta_1} \right)^t \right. \right. \\ & \left. \left. + \left( \frac{\partial^2 \bar{\mathbf{p}}_q^a}{\partial \vartheta_j \partial \vartheta_k} m_q \right) \left( \frac{\partial \bar{\mathbf{p}}_q^a}{\partial \vartheta_1} \right)^t \right\} \dot{\vartheta}_j \dot{\vartheta}_k \right. \\ & \left. + \text{Tr} \left\{ (m_i g \mathbf{z}_0) (R_0^{i-1} R_i^{i-1} \mathbf{z}_{i-1} \times \bar{\mathbf{p}}_{i-1}^a) \right\} \right] . \end{aligned} \quad 4.3$$

(In dealing with the gravity term one can use Eqn. 2.23 and 2.14 to show that:

$$(m_i g z_0)^t \frac{\partial R_0^q}{\partial \vartheta_i} \bar{r}_1 + \frac{\partial \mathbf{p}_0^q}{\partial \vartheta_i} = (m_i g z_0)^t R_0^{i-1} (z_{i-1} \times \bar{\mathbf{p}}_{i-1}^q) \quad 4.4$$

and then Eqn. 2.26 to obtain:

$$= \text{Tr} \{ (m_i g z_0) R_0^i (R_i^{i-1} z_{i-1} \times R_i^{i-1} \bar{\mathbf{p}}_{i-1}^q)^t \} \quad 4.5$$

One can contract the time derivatives to obtain:

$$\sum_{q=i}^n \text{Tr} \left\{ \frac{d^2 R_0^q}{dt^2} K_q \left( \frac{\partial R_0^q}{\partial \vartheta_i} \right)^t + m_q \frac{d^2 \bar{\mathbf{p}}_0^q}{dt^2} \left( \frac{\partial \bar{\mathbf{p}}_0^q}{\partial \vartheta_i} \right)^t \right\} \quad 4.6$$

$$+ \text{Tr} \{ (m_i g z_0) R_0^i (R_i^{i-1} z_{i-1} \times R_i^{i-1} \bar{\mathbf{p}}_{i-1}^q)^t \}$$

Now define an acceleration matrix,  $A_q$ :

$$A_q = R_q^0 \frac{d^2 R_0^q}{dt^2} \quad 4.7$$

By Eqns. 2.14 and 2.15 another relation for  $A_q$  is:

$$A_q = R_q^0 \left[ \sum_{j=1}^q R_0^{j-1} Q^z R_{j-1}^q \ddot{\vartheta}_j + \sum_{j=1}^q \sum_{k=1}^q R_0^{u-1} Q^z R_{u-1}^v Q^z R_{v-1}^q \dot{\vartheta}_j \dot{\vartheta}_k \right] \quad 4.8$$

where  $u = \min(j, k)$  and  $v = \max(j, k)$ .

And now an aside to determine the form of the  $A_q$  term in Eqn. Define a quantity by the iterative relation:

$$\omega_q = R_q^{q-1} (\omega_{q-1} + z_{q-1} \dot{\vartheta}_q), \quad \omega_0 = 0 \quad 4.9$$

Upon expanding this becomes:

$$\omega_q = \sum_{j=1}^q R_j^{j-1} z_{j-1} \dot{\vartheta}_j \quad 4.10$$

Define a second quantity by the relation:

$$\alpha_q = R_q^{q-1} (\alpha_{q-1} + z_{q-1} \ddot{\vartheta}_q + \omega_{q-1} \times z_{q-1} \dot{\vartheta}_q), \quad 4.11$$

iterating this becomes:

$$\begin{aligned}
\alpha_q &= \sum_{j=1}^q R_q^{j-1} z_{j-1} \ddot{\vartheta}_j + \sum_{j=1}^q R_q^{j-1} (\omega_{j-1} \times z_{j-1}) \dot{\vartheta}_j \\
&= \sum_{j=1}^q R_q^{j-1} z_{j-1} \ddot{\vartheta}_j + \sum_{j=1}^q \sum_{k=1}^{j-1} (R_q^{k-1} z_{k-1} \times R_q^{j-1} z_{j-1}) \dot{\vartheta}_j \dot{\vartheta}_k
\end{aligned} \tag{4.12}$$

( $\alpha_q$  and  $\omega_q$  are, of course by Eqns. 3.7 and 3.8 just the angular acceleration and velocity expressed in the  $i^{\text{th}}$  frame.)

Consider an arbitrary vector in the  $q^{\text{th}}$  frame  $c_q$  then:

$$\alpha_q \times c_q + \omega_q \times (\omega_q \times c_q), \tag{4.13}$$

can be written as:

$$\begin{aligned}
&\sum_{j=1}^q R_q^{j-1} z_{j-1} \ddot{\vartheta}_j \times c_q + \sum_{j=1}^q \sum_{k=1}^{j-1} (R_q^{k-1} z_{k-1} \dot{\vartheta}_k \times R_q^{j-1} z_{j-1} \dot{\vartheta}_j) \times c_q \\
&+ \sum_{j=1}^q \sum_{k=1}^q R_q^{j-1} z_{j-1} \dot{\vartheta}_j \times (R_q^{k-1} z_{k-1} \overset{n}{t}_k \times c_q).
\end{aligned} \tag{4.14}$$

Employing Eqn 2.27 on the second term and third terms and moving a rotation  $R_q^0$  out using Eqn 2.3 one obtains:

$$R_q^0 \left[ \sum_{j=1}^q R_0^{j-1} z_{j-1} \times c_q \ddot{\vartheta}_j + \sum_{j=1}^q \sum_{k=1}^q R_0^{u-1} z_{u-1} \dot{\vartheta}_u \times (R_0^{v-1} z_{v-1} \dot{\vartheta}_v \times R_{v-1}^q c_q) \right] \tag{4.16}$$

Since a cross product can be replaced by a  $Q^\beta$  matrix operation from Eqn. 2.8 one obtains:

$$= R_q^0 \sum_{j=1}^q Q^z c_q R_{j-1}^q \ddot{\vartheta}_j + \sum_{j=1}^q \sum_{k=1}^q R_0^{u-1} Q^z R_{u-1}^{v-1} Q^z R_{v-1}^q c_q \dot{\vartheta}_j \dot{\vartheta}_k. \tag{4.17}$$

Thus from Eqns. 4.8 and 4.13 one can see that  $A_q$  can be written as:

$$A_q = (\alpha_q)_\beta Q^\beta + (\omega_q)_\beta Q^\beta (\omega_q)_\chi Q^\chi \tag{4.18}$$

One recognizes the acceleration  $\bar{a}_q$  represented in the  $q^{\text{th}}$  system:

$$m_q \frac{d^2 \bar{p}_0^q}{dt^2} = R_0^q m_q \bar{a}_q \tag{4.19}$$

Note that:

$$\frac{d^2 \bar{\mathbf{p}}_0^q}{dt^2} = \sum_{j=1}^q \frac{d^2 R_0^j}{dt^2} \bar{\mathbf{p}}_j^q = \sum_{q=i}^n R_0^j A_j \bar{\mathbf{p}}_j^q = R_0^q \sum_{j=1}^q R_q^j A_j \sum_{s=j+1}^q R_j^s \mathbf{r}_s + A_q \bar{\mathbf{r}}_q, \quad 4.20$$

hence:

$$R_0^q \bar{\mathbf{a}}_q = R_0^q A_q \bar{\mathbf{r}}_q + R_q^{q-1} \mathbf{a}_{q-1}, \quad 4.21$$

where  $\mathbf{a}_q$  has been defined by the iterative relation:

$$\mathbf{a}_q = A_q \mathbf{r}_q + R_q^{q-1} \mathbf{a}_{q-1} \quad 4.22$$

These are just equations 3.9 and 3.10. Using Eqns. 4.7, 4.21, and 4.22, the equation for tau, Eqn. 4.6 becomes:

$$\tau_i = \sum_{q=i}^n \text{Tr} \left\{ R_0^q (A_q K_q (R_0^q Q^\gamma (R_q^{i-1} \mathbf{z}_{i-1}))^\gamma)^\dagger + m_q \bar{\mathbf{a}}_q (R_0^q R_q^i R_q^{i-1} \mathbf{z}_{i-1} \times \bar{\mathbf{p}}_{i-1}^q)^\dagger \right\} \quad 4.23$$

The gravity term has been absorbed into the acceleration iterative relation Eqn. 4.22, whereas, now  $\mathbf{a}_0 = m_i g \mathbf{z}_0$ .

Consider the second term. The trace of an outer product can be changed into an inner product by Eqn. 2.26:

$$\begin{aligned} & \sum_{q=i}^n \text{Tr} \left\{ R_i^q m_q \bar{\mathbf{a}}_q (R_q^{i-1} \mathbf{z}_{i-1} \times R_i^{i-1} \bar{\mathbf{p}}_{i-1}^q)^\dagger \right\} \\ &= \sum_{q=i}^n (R_q^{i-1} \mathbf{z}_{i-1} \times R_i^{i-1} \bar{\mathbf{p}}_{i-1}^q)^\dagger R_i^q m_q \bar{\mathbf{a}}_q \end{aligned} \quad 4.24$$

Now interchanging the inner and cross product one obtains:

$$(R_q^{i-1} \mathbf{z}_{i-1})^\dagger \sum_{q=i}^n R_i^{i-1} \bar{\mathbf{p}}_{i-1}^q \times R_i^q m_q \bar{\mathbf{a}}_q \quad 4.25$$

The equation for  $\tau_i$  is:

$$\begin{aligned} \tau_i &= \sum_{q=i}^n \text{Tr} \left\{ A_q K_q Q^\gamma (R_q^{i-1} \mathbf{z}_{i-1})^\gamma \right\} \\ &+ (R_q^{i-1} \mathbf{z}_{i-1})^\dagger \sum_{q=i}^n R_i^{i-1} \bar{\mathbf{p}}_{i-1}^q \times R_i^q m_q \bar{\mathbf{a}}_q \end{aligned} \quad 4.26$$

one can see that this can be rewritten:

$$\tau_i = \sum_{q=1}^n (R_q^{i-1} \mathbf{z}_{i-1})_\gamma \left[ \text{Tr} \left\{ A_q K_q (Q^\gamma)^t \right\} + \left( \sum_{q=1}^n R_q^{i-1} \bar{\mathbf{p}}_{i-1}^q \times R_q^q m_q \bar{\mathbf{a}}_q \right)_\gamma \right] \quad 4.27$$

The first term in Eqn. 4.27 becomes:

$$(R_q^{i-1} \mathbf{z}_{i-1})_\gamma \sum_{q=1}^n \text{Tr} \left\{ \left[ (\alpha_q)_\beta Q^\beta + (\omega_q)_\beta Q^\beta (\omega_q)_\chi Q^\chi K_q \right] (Q^\gamma)^t \right\} \quad 4.28$$

Examine the first term in terms of its components:

$$(R_q^{i-1} \mathbf{z}_{i-1})_\gamma (\alpha_q)_\beta Q_{\chi\epsilon}^\gamma Q_{\epsilon\lambda}^\beta (K_q)_{\lambda\chi} \quad 4.29$$

employing Eqn. 2.10

$$= (R_q^{i-1} \mathbf{z}_{i-1})_\gamma (\alpha_q)_\beta (\delta_{\chi\beta} \delta_{\gamma\lambda} - \delta_{\chi\lambda} \delta_{\gamma\beta}) (K_q)_{\lambda\chi}$$

$$(R_q^{i-1} \mathbf{z}_{i-1})_\gamma (K_q - E \text{Tr} \{ K_q \})_{\gamma\beta} (\alpha_q)_\beta = (R_q^{i-1} \mathbf{z}_{i-1})^t I_q \alpha_q .$$

K is related to I through Eqn. 2.36. The second term of Eqn. 4.28, in terms of its components is just:

$$(R_q^{i-1} \mathbf{z}_{i-1})_\gamma (\omega_q)_\beta (\omega_q)_\chi \text{Tr} \left\{ (Q^\gamma)^t Q^\beta Q^\chi K_q \right\} \quad 4.30$$

Consider the inside of the trace written in component form:

$$Q_{\epsilon\alpha}^\gamma Q_{\epsilon\rho}^\beta Q_{\rho\varphi}^\chi (K_q)_{\varphi\alpha} , \quad 4.31$$

using 2.10 this is:

$$Q_{\epsilon\alpha}^\gamma \delta_{\epsilon\chi} \delta_{\beta\varphi} - \delta_{\epsilon\varphi} \delta_{\beta\chi} (K_q)_{\alpha\varphi} , \quad 4.32$$

where the symmetry of  $K_q$  has been taken into account to allow  $(K_q)_{\varphi\alpha}$  to be rewritten as  $(K_q)_{\alpha\varphi}$ . Contracting the Kronecker delta's and including the terms outside the trace, one obtains:

$$(R_q^{i-1} \mathbf{z}_{i-1})_\gamma (\omega_q)_\chi Q_{\chi\alpha}^\gamma (K_q)_{\alpha\beta} (\omega_q)_\beta - (R_q^{i-1} \mathbf{z}_{i-1})_\gamma Q_{\varphi\alpha}^\gamma (K_q)_{\alpha\varphi} (\omega_q)_\beta (\omega_q)_\beta \quad 4.33$$

This is:

$$(R_q^{i-1} \mathbf{z}_{i-1})^t [\boldsymbol{\omega}_q \times K_q \boldsymbol{\omega}_q + 0] \quad 4.34$$

Since  $\boldsymbol{\omega}_q \times (-E \text{Tr} \{K_q\}) \boldsymbol{\omega}_q = 0$  one can add this in and obtain an expression with the inertia I matrix:

$$(R_q^{i-1} \mathbf{z}_{i-1})^t [\boldsymbol{\omega}_q \times I_q \boldsymbol{\omega}_q] \quad 4.35$$

Adding 4.35 to the previous result of Eqn. 4.29 one has:

$$\text{Tr} \left\{ \mathbf{A}_q K_q (Q^\gamma (R_q^{i-1} \mathbf{z}_{i-1}))^\gamma \right\} = (R_q^{i-1} \mathbf{z}_{i-1})^t [I_q \boldsymbol{\alpha}_q + \boldsymbol{\omega}_q \times (I_q \boldsymbol{\omega}_q)] \quad 4.36$$

Thus one can see that the equation for  $\tau_i$ , Eqn 4.27 can be rewritten as:

$$\begin{aligned} \tau_i &= \sum_{q=i}^n (R_q^{i-1} \mathbf{z}_{i-1})^t [I_q \boldsymbol{\alpha}_q + \boldsymbol{\omega}_q \times (I_q \boldsymbol{\omega}_q) + R_i^{i-1} \bar{\mathbf{p}}_{i-1}^q \times R_i^q m_q \bar{\mathbf{a}}_q] \\ &= (R_i^{i-1} \mathbf{z}_{i-1})^t \sum_{q=i}^n R_i^q [I_q \boldsymbol{\alpha}_q + \boldsymbol{\omega}_q \times (I_q \boldsymbol{\omega}_q) + R_i^{i-1} \bar{\mathbf{p}}_{i-1}^q \times R_i^q m_q \bar{\mathbf{a}}_q] \end{aligned} \quad 4.37$$

$\bar{\mathbf{p}}_{i-1}^q$  by definition Eqn. is:

$$\sum_{s=i}^{q-1} R_{i-1}^s \mathbf{r}_s + R_{i-1}^q (\mathbf{r}_q + \bar{\mathbf{r}}_q) \quad 4.38$$

Combining this with  $R_i^q m_q \bar{\mathbf{a}}_q$  one obtains:

$$\begin{aligned} \sum_{q=i}^n \bar{\mathbf{p}}_{i-1}^q \times R_i^q m_q \bar{\mathbf{a}}_q &= \sum_{q=i}^n R_i^q (\mathbf{r}_q + \bar{\mathbf{r}}_q) \times R_i^q m_q \bar{\mathbf{a}}_q + \sum_{s=i}^n R_i^s (\mathbf{r}_s \times \sum_{q=s+1}^n R_s^q m_q \bar{\mathbf{a}}_q) \\ &= \sum_{q=i}^n R_i^q [(\mathbf{r}_q + \bar{\mathbf{r}}_q) \times m_q \bar{\mathbf{a}}_q + \sum_{s=i}^n R_i^s (\mathbf{r}_s \times R_s^{s+1} \mathbf{f}_{s+1})], \end{aligned} \quad 4.39$$

where  $\mathbf{f}_{s+1}$  is defined as  $\sum_{q=s+1}^n R_{s+1}^q m_q \bar{\mathbf{a}}_q$ . (Notice that this is just Eqn. 3.11). If

one now defines  $\mathbf{n}_i$  by the recursion (same as Eqn. 3.12):

$$\mathbf{n}_i = I_i \boldsymbol{\alpha}_i + \boldsymbol{\omega} \times I_i \boldsymbol{\omega}_i + (\mathbf{r}_i + \bar{\mathbf{r}}_i) \times \bar{\mathbf{a}}_i + \mathbf{r}_i \times R_i^{i+1} \mathbf{f}_{i+1} + R_i^{i+1} \mathbf{n}_{i+1}. \quad 4.40$$

one can see that  $\tau_i$  can be written as:

$$\tau_i = (R_i^{i-1} \mathbf{z}_{i-1})^t \mathbf{n}_i$$

which is Eqn. 3.13.

## 5. CONNECTION BETWEEN RECURSIVE LAGRANGE AND NEWTON-EULER

The first item to note is that the recursive vector  $\mathbf{e}_i$  defined in Eqn. 3.3 is just the vector  $R_0^i \mathbf{f}_i$  defined in Eqn. 3.11.

The second substitution to be performed is to replace  $J_i$  by  $K_i + m_i \bar{\mathbf{r}}_i \bar{\mathbf{r}}_i^t$  using Eqn 2.35. Then the defining equation for  $D_i$  becomes:

$$\begin{aligned} D_i = & R_i^{i+1} D_{i+1} + \mathbf{r}_i (R_0^{i+1} \mathbf{f}_{i+1})^t + K_i \left( \frac{d^2 R_0^i}{dt^2} \right)^t + m_i \bar{\mathbf{r}}_i \bar{\mathbf{r}}_i^t \left( \frac{d^2 R_0^i}{dt^2} \right)^t \\ & + m_i \bar{\mathbf{r}}_i \left( \frac{d^2 R_0^i}{dt^2} \bar{\mathbf{r}}_i \right)^t + m_i (\bar{\mathbf{r}}_i + \mathbf{r}_i) \left( \frac{d^2 \mathbf{p}_0^i}{dt^2} \right)^t + m_i \mathbf{r}_i \left( \frac{d^2 R_0^i}{dt^2} \bar{\mathbf{r}}_i \right)^t \end{aligned} \quad 5.1$$

The last four terms can be combined into:

$$m_i (\mathbf{r}_i + \bar{\mathbf{r}}_i) \left( \frac{d^2 \mathbf{p}_0^i}{dt^2} \right)^t = (\mathbf{r}_i + \bar{\mathbf{r}}_i) (R_0^i m_i \bar{\mathbf{a}}_i) ,$$

so the equation for  $\tau_i$ , Eqn. 3.2 can be written:

$$\tau_i = \text{Tr} \left\{ \frac{\partial R_0^i}{\partial \mathcal{V}_i} \left[ R_i^{i+1} D_{i+1} + \mathbf{r}_i R_0^{i+1} \mathbf{f}_{i+1} + K_i \left( \frac{d^2 R_0^i}{dt^2} \right)^t + (\mathbf{r}_i + \bar{\mathbf{r}}_i) (R_0^i m_i \bar{\mathbf{a}}_i)^t \right] \right\} \quad 5.2$$

The term  $\frac{\partial R_0^i}{\partial \mathcal{V}_i}$  can be rewritten as  $R_0^i (R_i^{i-1} \mathbf{z}_{i-1})_\gamma Q^\gamma$ . With this the third term can be rearranged into:

$$\text{Tr} \{ R_0^i (R_i^{i-1} \mathbf{z}_{i-1})_\gamma Q^\gamma K_i (R_0^i \mathbf{A}_i)^t \} = \text{Tr} \{ (R_i^{i-1} \mathbf{z}_{i-1})_\gamma Q^\gamma K_i \mathbf{A}_i^t \} \quad 5.3$$

Taking the transpose of the contents of the trace one obtains ( $K_i$  is symmetric):

$$\text{Tr} \{ \mathbf{A}_i K_i Q^\gamma (R_i^{i-1} \mathbf{z}_{i-1})_\gamma \} = (R_i^{i-1} \mathbf{z}_{i-1})^t I_i \boldsymbol{\alpha}_i + \boldsymbol{\omega}_i \times (I_i \boldsymbol{\omega}_i) \} , \quad 5.4$$

using Eqn. 4.36. The second term in Eqn. 5.2 can be treated straightforwardly.

After substituting in for  $\frac{\partial R_0^i}{\partial \mathcal{V}_i}$  one obtains:

$$\begin{aligned} \text{Tr} \{ R_0^i R_i^{i-1} \mathbf{z}_{i-1} \times \mathbf{r}_i (R_0^i R_i^{i+1} \mathbf{f}_{i+1})^t \} &= \text{Tr} \{ (R_i^{i-1} \mathbf{z}_{i-1} \times \mathbf{r}_i) (R_i^{i+1} \mathbf{f}_{i+1})^t \} \\ &= (R_i^{i+1} \mathbf{f}_{i+1})^t (R_i^{i-1} \mathbf{z}_{i-1} \times \mathbf{r}_i) = (R_i^{i-1} \mathbf{z}_{i-1})^t \mathbf{r}_i \times R_i^{i+1} \mathbf{f}_{i+1} , \end{aligned} \quad 5.5$$

upon exchanging a dot and cross product. In a similar fashion one can show that the third term of Eqn. 4.23 can be written as:

$$(\mathbf{R}_i^{i-1} \mathbf{z}_{i-1})^t \left[ (\mathbf{r}_i + \bar{\mathbf{r}}_i) \times m_i \bar{\mathbf{a}}_i \right] \quad 5.6$$

The terms found in Eqns. 5.4, 5.5, and 5.6 are just the terms of Eqn. 3.12 for  $\mathbf{n}_i$ . With a little more effort one can show that the  $\mathbf{R}_i^{i+1} D_i$  term gives a  $\mathbf{R}_i^{i+1} \mathbf{n}_{i+1}$  contribution.

The Recursive and Newton-Euler formulations are very close. As mentioned before the only reason that the Recursive-Lagrangian is computationally more complex than the Newton-Euler is that the Recursive Lagrangian passes 3 by 3 matrices between link frames. These matrices are then combined with the  $\mathbf{l}$ ,  $\mathbf{r}$ , and,  $\bar{\mathbf{r}}$  fixed objects. The Newton-Euler, however, passes all quantities as vectors between frames and saves considerable computation.



## 6. CONNECTION BETWEEN GIBBS-APPELL AND NEWTON-EULER

The Gibbs-Appell set of Horowitz and Tomizuka [HoT80] is yet another formulation that falls between the Lagrange and the Newton-Euler. We show in the following that it is quite close to the Newton-Euler.

Examine Eqn. 3.14. If all cross products with  $\mathbf{z}_{i-1}^{\bullet}$  are exchanged with dot products evoking Eqn. 2.25 the  $\mathbf{z}_{i-1}^{\bullet}$  terms may be pulled to the front:

$$\begin{aligned} \tau_i = & (\mathbf{z}_{i-1}^{\bullet})^t \sum_{q=1}^n \left[ \sum_{j=1}^q \left[ I_q^{\bullet} \mathbf{z}_{j-1}^{\bullet} + \bar{\mathbf{p}}_i^{\bullet q} \times (\mathbf{z}_{j-1}^{\bullet} \times \bar{\mathbf{p}}_j^{\bullet q}) \right] \ddot{\vartheta}_j \right. \\ & + \sum_{j=1}^q \sum_{k=1}^q \left[ \mathbf{z}_{u-1}^{\bullet} \times \left[ I_q^{\bullet} \mathbf{z}_{v-1}^{\bullet} - \frac{1}{2} \text{Tr} \{ I_q^{\bullet} \} \right. \right. \\ & \left. \left. + \bar{\mathbf{p}}_i^{\bullet q} \times (\mathbf{z}_{u-1}^{\bullet} \times (\mathbf{z}_{v-1}^{\bullet} \times \bar{\mathbf{p}}_v^{\bullet q})) \right] \dot{\vartheta}_j \dot{\vartheta}_k \right] \end{aligned} \quad 6.1$$

The second and fifth term are from Eqn. 4.16 just:

$$\sum_{q=1}^n \bar{\mathbf{p}}_i^{\bullet q} \times \sum_{j=1}^q A_j^{\bullet} \bar{\mathbf{p}}_j^{\bullet q} = \sum_{q=1}^n \bar{\mathbf{p}}_i^{\bullet q} \times \bar{\mathbf{a}}_q^{\bullet} \quad 6.2$$

In order to combine the first, third and fourth terms, they are pulled into a trace. The first term is:

$$\mathbf{z}_{i-1}^{\bullet t} I_q^{\bullet} \mathbf{z}_{j-1}^{\bullet} \ddot{\vartheta}_j = (\mathbf{z}_{i-1}^{\bullet})_{\gamma} [(K_q^{\bullet})_{\gamma\beta} - \delta_{\gamma\beta} (K_q^{\bullet})_{\lambda\lambda}] (\mathbf{z}_{j-1}^{\bullet})_{\beta} \ddot{\vartheta}_j \quad 6.3$$

from Eqn. 2.36:

$$\begin{aligned} & = (\mathbf{z}_{i-1}^{\bullet})_{\gamma} (K_q^{\bullet})_{\lambda\lambda} (\mathbf{z}_{j-1}^{\bullet})_{\beta} (\delta_{\chi\beta} \delta_{\gamma\lambda} - \delta_{\chi\lambda} \delta_{\gamma\beta}) = (\mathbf{z}_{i-1}^{\bullet})_{\gamma} (K_q^{\bullet})_{\lambda\lambda} (\mathbf{z}_{j-1}^{\bullet})_{\beta} Q_{\chi\epsilon}^{\gamma} Q_{\epsilon\lambda}^{\beta} \\ & = (\mathbf{z}_{i-1}^{\bullet})_{\beta} Q_{\epsilon\lambda}^{\beta} K_q^{\bullet \lambda\chi} (\mathbf{z}_{i-1}^{\bullet})_{\gamma} Q_{\chi\epsilon}^{\gamma} = \text{Tr} \{ (\mathbf{z}_{i-1}^{\bullet} \ddot{\vartheta}_j) \times K_q^{\bullet \lambda\chi} (\mathbf{z}_{i-1}^{\bullet} Q^z)^t \} \end{aligned} \quad 6.4$$

With some work the third and fourth terms can be combined in a similar manner to yield:

$$\sum_{j=1}^q \sum_{k=1}^q \text{Tr} \{ \mathbf{z}_{u-1}^{\bullet} \times (\mathbf{z}_{v-1}^{\bullet} \times K_q^{\bullet}) (\mathbf{z}_{i-1}^{\bullet} Q^z)^t \} \dot{\vartheta}_j \dot{\vartheta}_k \quad 6.5$$

Employing relation Eqn. 2.28, Eqn. 6.4 and 6.5 are combined to yield:

$$\begin{aligned}
& \text{Tr}\left\{\left[\sum_{j=1}^q \mathbf{z}_{j-1}^* \ddot{\vartheta}_j \times K_q^* + \sum_{j=1}^q \sum_{k=1}^{j-1} (\mathbf{z}_{k-1}^* \times \mathbf{z}_{j-1}^*) \times K_q^* \dot{\vartheta}_k \dot{\vartheta}_j \right. \right. \\
& \left. \left. + \sum_{k=1}^q \sum_{j=1}^q \mathbf{z}_{j-1}^* \times (\mathbf{z}_{k-1}^* \times K_q^*) \dot{\vartheta}_k \dot{\vartheta}_j\right](\mathbf{z}_{i-1}^* Q^z)^t\right\} \\
& = \text{Tr}\{\alpha_q^* \times K_q^* + \omega_q^* \times (\omega_q^* \times K_q^*)(\mathbf{z}_{i-1}^* Q^z)^t\} = \text{Tr}\{A_q^* K_q^*(\mathbf{z}_{i-1}^* Q^z)^t\} \tag{6.6}
\end{aligned}$$

Combining 6.6 and 6.2 we have Eqn. 4.26 represented in the base frame. The final connection between the Gibbs-Appell and the Newton-Euler follows from the discussion in Sec. 4.

## 7. IMPROVEMENTS TO NEWTON-EULER

Eqns 3.7 and 3.8:

$$\boldsymbol{\omega}_i = R_i^{j-1} (\boldsymbol{\omega}_{i-1} + \dot{\vartheta}_i \mathbf{z}_{i-1}) \quad 7.1$$

$$\boldsymbol{\alpha}_i = R_i^{j-1} (\boldsymbol{\alpha}_{i-1} + \boldsymbol{\omega}_i \times \mathbf{z}_{i-1} \dot{\vartheta}_i + \ddot{\vartheta}_i \mathbf{z}_{i-1}), \quad 7.2$$

are basic to the model but one can use the consolidation operation introduced in Eqn. 4.18 to reexpress Eqns. 3.9 - 3.11 as:

$$\mathbf{a}_i = A_i \mathbf{r}_i + R_i^{j-1} \mathbf{a}_{i-1} \quad 7.3$$

$$\bar{\mathbf{a}}_i = A_i \bar{\mathbf{r}}_i + \mathbf{a}_i \quad 7.4$$

$$\mathbf{f}_i = m_i \bar{\mathbf{a}}_i + R_i^{j+1} \mathbf{f}_{i+1} \quad 7.5$$

And if one uses the first term from Eqn. 4.27 for  $\tau_i$  and hence for  $\mathbf{n}_i$  one can write:

$$\mathbf{n}_i = \text{Tr} \{A_i K_i (Q^{\gamma})^t\} \hat{\mathbf{x}}_{\gamma} + [m_i (\bar{\mathbf{r}}_i + \mathbf{r}_i) \times \bar{\mathbf{a}}_i + (\mathbf{r}_i) \times R_i^{j+1} \mathbf{f}_{i+1} + R_i^{j+1} \mathbf{n}_{i+1}], \quad 7.6$$

The first term for  $\mathbf{n}$  looks ominous but amounts to a  $(A_i K_i)_{yz} - (A_i K_i)_{zy}$  contribution to  $n_x$  and similar contributions to  $n_y$  and  $n_z$ .

This form saves some computation (see next section, Sec. 8, especially Tables 8.4 and 8.5 and gives more parallelism to the computations (see Table 8.6).

## 8. COMPUTATIONAL COMPLEXITY COMPARISON

In the following we compare the Lagrange, recursive Lagrange, Newton-Euler, and modified Newton-Euler formulations to determine their relative computational complexity as a function of the number of links in the arm,  $n$ . The complexity of the three approaches is displayed in Table 8.1.

Approach	multiplications	additions
Lagrange	$\frac{81}{6}n^3 + \frac{165}{2}n^2 + 5n$	$\frac{40}{3}n^3 + 58n^2 - \frac{64}{3}n$
Recursive Lagrange	$226n$	$144n$
Newton-Euler	$108n - 12$	$100n - 9$
Modified Newton-Euler	$90n - 27$	$88n - 24$

Table 8.1. Computational Complexity of Formulations

A similar table was derived by Hollerbach [Hol80]. However, he arrives at an  $n^4$  dependence for the Lagrange formulation and a  $150n$  dependence for the linear Newton-Euler formulation. The discrepancy can be accounted for by the fact that he carried out the operations "more or less as set forth" [Hol80] and made no effort to interpret the equations more efficiently.

To see how Table 8.1 was derived consider first the Lagrange approach. Determining the kinematic contribution for the  $\ddot{\vartheta}_j$  coefficient is linear in the number of links,  $n$ , but determining the coefficient of the  $\dot{\vartheta}_j \dot{\vartheta}_k$  is of order  $n^2$  since the calculations must be done for each value of  $j$  and  $k$ . These kinematic calculations are then reperformed for all  $n$  torque calculations. So the whole process is of order  $n^3$ . A breakdown of computations is shown in the Table 8.2.

Lagrange terms	multiplications	additions
$T_j^i$	$32n(n-1)$	$24n(n-1)$
$\left[ \frac{\partial T_0^i}{\partial \vartheta_j} \right]_e$	$32n(n-1)$	$24n(n-1)$
$\left[ \frac{\partial^2 T_0^i}{\partial \vartheta_j \partial \vartheta_k} \right]_v$	$\frac{32}{3} n(n^2-1)$	$8n(n^2-1)$
$\left[ \sum_{j=1}^q \left\{ \left[ \dots \right]_e \ddot{\vartheta}_j + \sum_{k=1}^q \left[ \dots \right]_v \dot{\vartheta}_j \dot{\vartheta}_k \right\} J_q^a \right]_e$	$\frac{17}{6} n^3 + \frac{17}{2} n^2 + \frac{209}{3} n$	$\frac{16}{3} n^3 + \frac{80}{3} n$
$\text{Tr} \left\{ \left[ \dots \right]_e \frac{\partial T_0^a}{\partial \vartheta_i} \right\}$	$8n^2 + 8n$	$8n^2 + 7n$
$\sum_{q=1}^n (z_0^a)^t \frac{\partial T_1^a}{\partial \vartheta_i} m_q g \bar{r}_q^a$	$2n^2 + 2n$	$2n^2 + n$
Total	$\frac{81}{6} n^3 + \frac{165}{2} n^2 + 5n$	$\frac{40}{3} n^3 + 58n^2 - \frac{64}{3} n$

Table 8.2. Breakdown of Lagrange Terms

Multiplications by  $Q_j^i$  have been ignored since they amount to a row interchange and a row negation. Appendix A shows in more detail how the terms are computed.

Now consider the Recursive Lagrange formulation.

Recursive Lagrange terms	multiplications	additions
$\frac{dR_0^j}{dt}$	33n	24n
$\frac{d^2R_0^j}{dt^2}$	46n	36n
$\frac{\partial R_0^j}{\partial \vartheta_i}$	45n	27n
$\frac{d^2P_0^j}{dt^2}$	9n	9n
$R_i^{j+1} D_{i+1}$	27n	18n
$J_i \left( \frac{d^2R_0^j}{dt^2} \right)^t$	27n	18n
$m_i (\bar{r}_i + r_i) \left( \frac{d^2P_0^j}{dt^2} \right)^t$	9n	0
$r_i \left( \frac{d^2R_0^j}{dt^2} \bar{r}_i \right)^t$	18n	6n
$r_i e_{i+1}$	9n	0
$m_j \left( \frac{d^2P_0^j}{dt^2} \right)^t + \bar{r}_i \left( \frac{dR_0^j}{dt} \right)^t$	3n	6n
Total	226n	144n

Table 8.3. Breakdown of Recursive Lagrange Terms

Now consider the Newton-Euler computations. Using the Newton-Euler equation set one moves from the base of the arm to the hand in computing the kinematics and then from the hand to the base in computing the dynamics.

Thus to compute the torque,  $\tau$ , the kinematic and dynamic calculations are performed only once for each link. If there are  $K$  kinematic and  $D$  dynamic calculations per link, the computational complexity of the Newton-Euler set for an  $n$  link arm is  $(K+D)n$ : a linear computation scheme. Besides this linearity another advantage of the Newton-Euler set is that terms representing insignificant torque contributions can be easily identified and, if approximations are acceptable, deleted. Identifying insignificant terms in the Lagrangian formulation is made difficult because individual contributions tend to be combined in unintuitive ways. (It was noted earlier that the Newton-Euler formulation provided greater physical insight into the problem.) The Newton-Euler and the modified Newton-Euler computations are broken down in Table 8.4 and 8.5.

Many of the arithmetic operations tabulated in Table 8.4 can be performed concurrently. Table 8.6 presents the number of steps required to perform the modified Newton-Euler equations if this concurrency is accounted for.

N-E terms	multiplications	additions
$\omega_i$	$9n$	$7n$
$\alpha_i$	$9n$	$9n$
$A_i$	$6n$	$9n$
$a_i$	$18n$	$15n$
$\bar{a}_i$	$9n$	$9n$
$f_i$	$12(n-1)$	$9(n-1)$
$I_i \alpha_i$	$9n$	$6n$
$\omega_i \times (I_i \omega_i)$	$15n$	$9n$
$m(r_i + \bar{r}_i) \times \bar{a}_i$	$6n$	$3n$
$r_i \times R_i^{j+1} f_{i+1}$	$6n$	$3n$
$R_{i+1}^i n_{i+1}$	$9n$	$6n$
add $n$	$0$	$15n$
Total	$108n-12$	$100n-9$

Table 8.4. Breakdown of Newton-Euler Terms  
 (Note: It is assumed that an  $A_i$  is computed rather than all the cross product terms.)



N-E terms	multiplications	additions
$\omega_i$	$9n$	$7n$
$\alpha_i$	$9n$	$9n$
$A_i$	$6n$	$9n$
$\mathbf{a}_i$	$18(n-1)$	$15(n-1)$
$m_i \bar{\mathbf{a}}_i$	$12n$	$9n$
$\mathbf{f}_i$	$9(n-1)$	$9(n-1)$
$\text{Tr} \{A_i K_i (Q^\gamma)^t\}$	$6n$	$3n$
$m(\mathbf{r}_i + \bar{\mathbf{r}}_i) \times \bar{\mathbf{a}}_i$	$6n$	$3n$
$\mathbf{r}_i \times R_i^{1+1} \mathbf{f}_{i+1}$	$6n$	$3n$
$R_{i+1}^1 \mathbf{n}_{i+1}$	$9n$	$6n$
add n	0	$15n$
Total	$90n-27$	$88n-24$

Table 8.5. Breakdown of modified Newton-Euler Terms

N-E terms	multiplications	additions
$\omega_1$	9	7
$\alpha_1$	9	9
$A_1$	6	9
$\omega_2$	9	7
$\alpha_2$	9	9
$\omega_{i+2}, \alpha_{i+2}, A_{i+1}, \mathbf{a}_i,$ $m_i \bar{\mathbf{a}}_i, \text{Tr} \{A_i K_i Q^T\}, m_i \bar{\mathbf{r}}_{i-1} \times \bar{\mathbf{a}}_{i-1}$	18n	15n
$\mathbf{f}_i, \mathbf{n}_i, \mathbf{r}_i \times R_i^{i+1} \mathbf{f}_{i+1},$ $R_{i+1}^i \mathbf{n}_{i+1}$	9n	12n
Total	27n+42	27n+41

Table 8.6. Simultaneous steps in N-E computation.

As mentioned before some applications require that the torque be in the following form:

$$\boldsymbol{\tau} = \mathbf{M}(\boldsymbol{\vartheta}) \ddot{\boldsymbol{\vartheta}} + \mathbf{C}(\dot{\boldsymbol{\vartheta}}, \boldsymbol{\vartheta}) + \mathbf{G}(\boldsymbol{\vartheta}) + \boldsymbol{\tau}_{\text{load}} \quad 8.1$$

This form results naturally from the Lagrangian formulations, however, it can be obtained with less effort from the Newton-Euler equation set using the following technique. If the application requires the form given in Eqn. 8.1, one can obtain the  $\mathbf{M}(\boldsymbol{\vartheta})_{ij}$  matrix element by "strobing" the iterative set of equations above with an input  $\ddot{\boldsymbol{\vartheta}}$  unit vector with all inputs except  $\ddot{\vartheta}_j$  set to zero and with gravity and the load  $\mathbf{f}_n$  and  $\mathbf{n}_n$  set to zero. This is the technique we use to obtain the M matrix in our simulation program. If one also requires the  $\mathbf{C}_{jk}^i$  elements,

they can be obtained by zeroing all  $\ddot{\vartheta}$ 's and  $\dot{\vartheta}$ 's except  $\dot{\vartheta}_j$  and  $\dot{\vartheta}_k$  which are set to 1. Gravity and the load are again set to zero. The number of calculations involved will be  $36n(n+1)$  multiplications and  $26n(n+1)$  additions for the M matrix and  $54n^3+27n^2+9n$  multiplications and additions for the  $C^i$  matrices. (Symmetry of the M and  $C^i$  matrices has been taken into account) The later shows that the Lagrangian could be useful if the  $C^i$ 's were needed (See Table 8.2) However, in the following section, Sec. 9, on simulation one only needs the M matrix and a lumped C and G contribution. We never use the explicit C matrices.

## 9. APPLICATION TO SIMULATION

The improved Newton-Euler together with the strobing technique proposed at the end of section 8 can be used to perform efficient simulations. An important use for simulation is in the evaluation of arm control strategies and allow improvements to the arm model.

In simulating the arm one is given an input torque vector  $\tau(t)$  and initial values of the relative angular velocity vector,  $\dot{\vartheta}(0)$  and relative angular position vector,  $\vartheta(0)$ , and are required to determine the resulting  $\ddot{\vartheta}(t)$ ,  $\dot{\vartheta}(t)$ , and  $\vartheta(t)$ . Solving Eqn. 8.1 for  $\ddot{\vartheta}(t)$ , one has an expression of the form:

$$\ddot{\vartheta} = M(\vartheta)^{-1}[\tau - \tau_{\text{load}} - C(\dot{\vartheta}, \vartheta) - G(\vartheta)] = f(\dot{\vartheta}, \vartheta) \quad 9.1$$

If  $\dot{\vartheta}$  is represented by  $\gamma$ , one has a system of  $2n$  equations (where again  $n$  is the number of links):

$$\dot{\gamma} = f(\gamma, \vartheta) \quad 9.2$$

$$\dot{\vartheta} = \gamma \quad 9.3$$

Now that one has the equations in this form one can perform a standard Runge-Kutta four point integration. Assume that  $\mathbf{y}$  represents the augmented  $2n$  vector  $(\gamma, \vartheta)^t$  and further assume that  $\mathbf{h}$  represents the augmented  $2n$  vector  $(f, \gamma)^t$ . The equations become simply:

$$\dot{\mathbf{y}} = \mathbf{h}(\mathbf{y}) \quad 9.4$$

The new values of  $\mathbf{y}$  are, of course, just the new values of  $\dot{\vartheta}$  and  $\vartheta$ . The step size  $\varepsilon$  can be determined once one knows the maximum possible change in  $\vartheta$  or  $\dot{\vartheta}$ , which can in turn be found from the maximum arm velocity.  $\varepsilon$  should be taken small enough so that changes in angle and angular velocity are not excessive.

Evaluating  $\mathbf{h}$  at its various input values is not simple. One must obtain the  $C$  and  $G$  contributions and subtract them from  $\tau$ . Then one must determine

the  $M$  matrix, invert it, and then solve for  $\ddot{\theta}$ . The Lagrange equations yield an explicit form for the  $M$ ,  $C$  and  $G$  matrices, but as was shown in the previous section, Sec. 8, they are of order  $n^3$ . Instead the following approach works better.

First one obtains the combined  $C+G$  contribution by zeroing the the  $\ddot{\theta}$  input vector and inputting the  $\dot{\theta}$  and  $\theta$  vectors. This contribution is subtracted from  $\tau$ . Next one inputs zero gravity and also zeroes out the  $\ddot{\theta}$  contributions, but retains the  $\theta$  values. The Newton-Euler equations are then strobed for various components of  $M$  by setting all but one component of  $\ddot{\theta}$  to zero. This process takes  $36n(n+1)$  multiplications and  $26n(n+1)$  additions as mention in the previous section, Sec. 8. Inverting  $M$  can be done by Gaussian elimination with no pivoting necessary since  $M$  is a positive definite matrix. [HoT80]  $\ddot{\theta}$  can be found using:

$$\ddot{\theta} = M^{-1}(\tau - \tau_{load} - C - G) \quad 9.5$$

The resultant  $\dot{\theta}$  and  $\theta$  are determined from the input  $\tau$  using Runge Kutta four point integration discussed above.

With real time simulation, one can compare measured arm torques with the simulation predicted torques allowing one to adjust inertias vectors, friction and other parameters in the model to fit the real arm motion.

## 10. APPLICATION TO CONTROL

In order to discuss real-time control of the arm one must consider an integrated model for the arm's motion. By this is meant, given the desired hand trajectory of the arm one should be able to perform all steps necessary to determine the torques needed to move the hand along that trajectory. Also one should take into account friction at the joints as well as vibration in the arm.

### 10.1. Determining joint velocities and accelerations

Consider the trajectory problem. In order to apply the equations of motion Eqns. 7.1, 7.2, 7.3, 7.5, and 7.6. one need know the joint angular velocities and accelerations  $\dot{\vartheta}_j$ 's and  $\ddot{\vartheta}_j$ 's.

If one looks at the equations of motion one can see that one can relate the hand's angular and linear acceleration to the  $\dot{\vartheta}_j$ 's and  $\ddot{\vartheta}_j$ 's by:

$$\begin{bmatrix} \alpha_h \\ a_h \end{bmatrix} = L \ddot{\vartheta} + N(\dot{\vartheta}), \quad 10.1$$

where  $\ddot{\vartheta}$  and  $\dot{\vartheta}$  are six vectors, L is a six by six matrix which exhibits the linear dependence of  $\alpha_h$  and  $a_h$ , (the angular and linear hand accelerations) on  $\ddot{\vartheta}$  and N is a nonlinear contribution from  $\dot{\vartheta}$  terms. To solve for  $\ddot{\vartheta}$ , one could "strobe" the equations as in Sec. 8.

First  $\ddot{\vartheta}$  is set to zero. One can find the nonlinear contributions of the last computed  $\dot{\vartheta}$ 's and that of gravity to the angular and linear hand acceleration. One can then "strobe" out the matrix L, one element at a time by zeroing all nonlinear contributions and by zeroing all but one  $\ddot{\vartheta}$  element. One can then solve for  $\ddot{\vartheta}$  through:

$$\ddot{\vartheta} = L^{-1} \left[ \begin{bmatrix} \alpha_h \\ a_h \end{bmatrix} - N(\dot{\vartheta}) \right] \quad 10.2$$

One can then use a Runge Kutta scheme to integrate  $\ddot{\vartheta}$  to obtain the new  $\dot{\vartheta}$

and  $\vartheta$ . Once one has  $\vartheta$ ,  $\dot{\vartheta}$ , and  $\ddot{\vartheta}$ , one can apply the equations of motion to determine the needed torques.

## 10.2. Load adjustment trajectory planning

One would like, if possible, to plan the arm trajectory "on the fly", allowing one to eliminate an initial delay time for trajectory computation, and also allowing one to adjust for unforeseen shifts in the load (i.e. sloshing mode in a liquid being carried).

The difficulty in planning any trajectory is determining hand linear and angular accelerations which do not overload the arm's actuators. The following method is proposed.

First block out a crude trajectory solving the Eqns. 10.1 and 9.5 for  $[\alpha_h, a_h]^t$  in terms of the available  $\tau_i$ 's at widely separated points on the desired path.

$$\begin{bmatrix} \alpha_{h \max} \\ a_{h \max} \end{bmatrix} - N(\vartheta) = LM^{-1}(\tau - \tau_{\text{load}} - C - G) \quad 10.3$$

This allows one a global view of what maximum linear and angular acceleration will be attainable on the actual trajectory. During actual arm movement one can use, Eqn. 10.3 to determine what acceleration the hand is actually capable in response to loading.

## 11. CONCLUSION

A thorough investigation of the Lagrange, Recursive Lagrange, Gibbs-Appell, and Newton-Euler arm formulations was performed. They were all shown to be consistent with minor corrections. The insight gained in this study enabled us to formulate a more efficient Newton-Euler equation set suitable for real-time control applications or high speed simulation studies. Additionally, the technique of strobing, outlined in Sec. 8 allows one to identify the inertial and coriolis-centrifugal matrices used in an explicit representation of the torques without recourse to the Lagrangian formulation.



## 12. BIBLIOGRAPHY

[Bej74] Bejczy, A. K., "Robot Arm Dynamics and Control," *Technical Memo 33-669*, Jet Propulsion Laboratory, February 1974.

[DeH55] Denavit, J., R. S. Hartenberg, "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," *Journal of Applied Mechanics*, pp. 215-221, Jun. 1955.

[Hol80] Hollerbach, J. M., "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. SMC-10, No. 11, pp. 730-736, Nov. 1980.

[HoT80] Horowitz, R., M. Tomizuka, "An Adaptive Control Scheme for Mechanical Manipulators- Compensation of Nonlinearity and Decoupling Control," *Dynamic System and Control Division of the A.S.M.E.*, Winter Annual Meeting, Chicago, Ill., Nov. 1980.

[Lew74] Lewis, R. A., "Autonomous Manipulation on a Robot: Summary of Manipulator Software Functions," *Technical Memo 33-679*, Jet Propulsion Lab, Mar. 1974.

[LWP80] Luh, J. Y. S., M. W. Walker, and R. P. C. Paul, "On-Line Computational Scheme for Mechanical Manipulators," *Trans. ASME: Journal of Dynamic Systems, Measurement, and Control*, Vol. 120, Jun. 1980 pp. 69-76.

[Pau72] Paul, R. "Modeling, Trajectory Calculation, and Servoing of a Computer Controlled Arm," *Stanford Artificial Intelligence Laboratory Memo AM-177*, Nov. 1972.

[Sym71] Symon, K. R., *Mechanics, 3rd addition, Addison-Wesley, 1971.*

[Uic67] Uicker, J. J., "Dynamic Force Analysis of Spatial Linkages", *Journal of Applied Mechanics* , pp. 418-423, June 1967.

[WaL78] Walker, M. W., and J. V. S. Luh, "Manipulator Control", *Technical Report TR-EE78-12* School of Electrical Engineering, Purdue University, March 1978.

## APPENDIX

The following is a possible strategy for computing the Lagrange equations in order  $n^3$  operations.

First build up all the transformations  $T_r^s$ . Product pairs are of the form:

$$T_0^1 T_1^2 \quad T_1^2 T_2^3 \quad T_3^4 T_4^5 \quad \dots \quad T_{n-2}^{n-1} T_{n-1}^n \quad n-1 \text{ T multiplications}$$

Triples can be built up as:

$$T_0^1 T_1^2 T_2^3 \quad T_1^2 T_2^3 T_3^4 \quad \dots \quad T_{n-3}^{n-2} T_{n-2}^{n-1} T_{n-1}^n \quad n-2 \text{ T multiplications}$$

Thus to compute all the  $T_j^k$ 's takes  $\sum_{m=1}^{n-1} (n-m)$  matrix multiplications or  $32n(n-1)$  multiplications and  $24n(n-1)$  additions.

One can compute the  $\frac{\partial T_0^1}{\partial \vartheta_j}$  in the following way.

$$Q_0^0 T_0^1 \quad \text{takes one Q multiplication}$$

$$Q_0^0 T_0^2 \quad \text{takes one Q multiplication}$$

$$T_0^1 Q_1^1 T_1^2 \quad \text{takes one Q and one T multiplication}$$

$$Q_0^0 T_0^3 \quad \text{takes one Q multiplication}$$

$$T_0^1 Q_1^1 T_1^3 \quad \text{takes one T multiplication}$$

$$T_0^2 Q_1^1 T_2^3 \quad \text{takes one Q and one T multiplication}$$

One can see the pattern. Each link takes  $n-1$  T multiplications. (Q multiplications are not counted since they consist of a row negation followed by a row interchange.) The total number of calculations is  $\sum_{m=1}^n (m-1)$  T multiplications or

$32n(n-1)$  multiplications and  $24n(n-1)$  additions.

The  $\frac{\partial T_0^1}{\partial \vartheta_j \partial \vartheta_k}$  terms can be computed using the  $T_j^1$  and  $\frac{\partial T_0^1}{\partial \vartheta_j}$  fragments as the examples below:

$Q_0^0 Q_0^0 T_0^1$	takes one Q multiplication
$Q_0^0 Q_0^0 T_0^2$	takes one Q multiplication
$Q_0^0 T_0^1 Q_1^1 T_1^2$	takes one Q multiplication
$T_0^1 Q_1^1 Q_1^1 T_1^2$	takes one Q and one T multiplication
$Q_0^0 Q_0^0 T_0^3$	takes one Q multiplication
$Q_0^0 T_0^1 Q_1^1 T_1^3$	takes one Q multiplication
$Q_0^0 T_0^2 Q_2^2 T_2^3$	takes one Q multiplication
$T_0^1 Q_1^1 Q_1^1 T_1^2 T_2^3$	takes one T multiplication
$T_0^1 Q_1^1 T_1^2 Q_2^2 T_2^3$	takes one T and one Q multiplication
$T_0^2 Q_2^2 Q_2^2 T_2^3$	takes one T and two Q multiplications

For each link  $i$  it takes  $\frac{i(i-1)}{2}$  T multiplications. If this is summed over all the links, one has  $\frac{32}{3}n(n^2-1)$  multiplications and  $16n(n^2-1)$  additions.

To produce  $\sum_{j=1}^q \frac{\partial T_0^q}{\partial \vartheta_j} \vartheta_j$  takes  $16q$  multiplications and  $16(q-1)$  additions.

To produce pairs  $\vartheta_j \vartheta_k$  takes  $\frac{q(q-1)}{2}$  multiplications. Since  $\frac{\partial^2 T_1^q}{\partial \vartheta_i \partial \vartheta_j} = \frac{\partial^2 T_1^q}{\partial \vartheta_j \partial \vartheta_i}$ ,

in order to produce  $\sum_{j=1}^q \sum_{k=1}^q \frac{\partial^2 T_1^q}{\partial \vartheta_j \partial \vartheta_i} \vartheta_j \vartheta_k$  takes an additional  $\frac{16q(q-1)}{2}$  multiplications and  $16(q+1)(q-1)$  additions. To combine these two terms takes an additional 16 additions. To multiply by  $J_q$  takes 64 multiplication and 48 additions.

To produce the kinetic term:

$$\left[ \sum_{j=1}^q \left\{ \frac{\partial T_{\sigma}^g}{\partial \vartheta_j} \dot{\vartheta}_j + \sum_{k=1}^q \frac{\partial T_{\sigma}^g}{\partial \vartheta_j \partial \vartheta_k} \dot{\vartheta}_j \dot{\vartheta}_k \right\} J_q \right]_{\nu}$$

takes  $8 \frac{1}{2} q(q+1) + 64$  multiplications and  $16q(q+1) + 32$  additions.

To produce the above terms for all values of  $q$  takes:

$$\sum_{q=1}^n [8 \frac{1}{2} q(q+1) + 64] = \frac{17}{6} n(n+1)(n+2) + 64n \quad \text{multiplications}$$

$$\sum_{q=1}^n [16q(q+1) + 32] = 32n + \frac{16}{3} n(n^2 - 1) \quad \text{additions}$$

Now pick a value for  $i$ . To produce:

$$\text{Tr} \left\{ \left[ \dots \right]_{\nu} J_q \frac{\partial T_{\sigma}^g}{\partial \vartheta_i} \right\}_{\epsilon}$$

for  $q=i$  to  $n$  takes  $16(n-i+1)$  multiplications and  $15(n-i+1)$  additions since one is only interested in the diagonal terms. Besides this there is a cost for summing the terms:

$$\sum_{q=i}^n [\dots]_{\epsilon}$$

of  $n-i$  additions. Summing contributions for all  $i$ 's results in:

$$\sum_{i=1}^n 16(n-i+1) = 8n(n+1) \quad \text{multiplications}$$

and

$$\sum_{i=1}^n 15(n-i+1) + n-i = 8n^2 + 7n \quad \text{additions}$$

In the gravity terms:

$$(\mathbf{z}_0^a)^t \sum_{q=i}^n \frac{\partial T_i^g}{\partial \vartheta_i} m_q g \bar{\mathbf{r}}_q^a$$

only one component of  $\frac{\partial T_i^g}{\partial \vartheta_i} m_q g \bar{\mathbf{r}}_q^a$  need be considered since the  $(\mathbf{z}_0^a)^t$  projects out only one component. Computing the needed component requires  $4(n-i+1)$  multiplications and  $3(n-i+1)$  additions. Summing this for all  $i$  yields  $2n^2 + 2n$  mul-

tuplications and  $\frac{3}{2}n^2 + \frac{3}{2}n$  additions. Add to this  $\frac{n^2}{2} - \frac{n}{2}$  additions to sum up the results.

Summing all contributions gives us a total of:

$$\frac{81}{6}n^3 + \frac{165}{2}n^2 + 5n \quad \text{total additions}$$

$$\frac{40}{3}n^3 + 58n^2 - \frac{64}{3}n \quad \text{total multiplications}$$

These are the results reported in Tables 8.1 and 8.2.

UNIVERSITY OF MICHIGAN



**3 9015 03526 7346**

UNIVERSITY OF MICHIGAN



**3 9015 03526 7346**