

MINIMAL PARAMETER SOLUTION  
OF THE  
ORTHOGONAL MATRIX DIFFERENTIAL EQUATION

by

Itzhack Y. Bar-Itzhack\* and F. Landis Markley\*\*

NASA - Goddard Space Flight Center  
Greenbelt, MD 20771

Abstract

As proven in this work, all orthogonal matrices solve a first order differential equation. The straightforward solution of this equation requires  $n^2$  integrations to obtain the elements of the the  $n$ -th order matrix. There are, however, only  $n(n-1)/2$  independent parameters which determine an orthogonal matrix. The questions of choosing them, finding their differential equation and expressing the orthogonal matrix in terms of these parameters are considered in this work. Several possibilities which are based on attitude determination in three dimensions (3-D) are examined. It is shown that not all 3-D methods have useful extensions to other dimensions. It is also shown why the rate of change of the matrix elements, which are the elements of the angular rate vector in 3-D, are the elements of a tensor of the second rank (dyadic) in spaces other than three dimensional. It is proven that the 3-D Gibbs vector (or Cayley Parameters) are extendible to other dimensions. An algorithm is developed employing the resulting parameters, which are termed Extended Rodrigues Parameters, and numerical results are presented of the application of the algorithm to a fourth order matrix.

---

\* National Research Council - NASA Resident Research Associate, Flight Dynamics Analysis Branch. On Sabbatical leave from the Aeronautical Engineering Department, Technion-Israel Institute of Technology, Haifa 32000, Israel.

\*\* Head, Mathematical Analysis Section, Systems Development Branch.

## I. INTRODUCTION

In a recent paper [1] a new algorithm for solving the matrix Riccati equation was introduced. The algorithm requires the solution of two matrix differential equations. The solution of one of the equations yields a diagonal matrix of the eigenvalues of  $P$ , the solution matrix of the Riccati equation. The other equation is

$$\dot{V}(t) = W(t)V(t) \quad (1)$$

where  $V$  is a matrix of the eigenvectors of  $P$ . Since  $P$  is a real symmetric matrix its eigenvectors are orthonormal, consequently  $V$  is an orthonormal matrix. (In the ensuing we will refer to an orthonormal matrix as an orthogonal one). The matrix  $W$  is a skew-symmetric matrix. (Note that in [1] the order of  $V$  and  $W$  on the right-hand side of (1) is reversed. This difference should cause no difficulty since  $V$  is the transpose of the corresponding matrix in [1] and  $W$  is the negative of its corresponding matrix).

Let  $n$  be the order of the square matrix  $V$ . The number of scalar integrations implied by (1) is  $n^2$ ; however, the orthogonality of  $V$  invokes  $n(n+1)/2$  relations among its elements. Therefore there are really only  $m = n(n-1)/2$  independent elements in  $V$ . The superfluous computational burden involved in the solution of (1) can, then, be reduced by properly defining the  $m$  independent parameters of  $V$ , solving a differential equation only for them and then performing an algebraic computation in order to transform these  $m$  elements into  $V$ .

We observe that (1) is identical to the famous differential equation of the transformation matrix in the three dimensional Euclidean space which is solved on-line for attitude determination of navigation and satellite systems. That matrix, of course, is also orthogonal, and  $W$  is a skew-symmetric matrix whose entries are the three components of the angular velocity vector at which the body rotates with respect to some reference coordinates. One question that comes immediately to mind is: does (1) always yield a solution which is orthogonal? and conversely, do all orthogonal matrices solve such a differential equation?

The answer to these two questions is formulated in the following two theorems.

**Theorem I.1:** Given equation (1) for  $t_0 < t < t_1$  where

$$W^T(t) = -W(t) \quad (2)$$

then:

- (I) The matrix  $V^T(t)V(t)$  is a constant matrix.

(II) If the initial matrix  $V(t_0)$  is orthogonal, then  $V(t)$  is orthogonal too.

**Proof:**

$$\frac{d}{dt} [V^T(t)V(t)] = \dot{V}^T(t)V(t) + V^T(t)\dot{V}(t) \quad (3)$$

substituting (1) into (3) yields

$$\frac{d}{dt} [V^T(t)V(t)] = V^T(t)W^T(t)V(t) + V^T(t)W(t)V(t) \quad (4)$$

and when (2) is substituted into (4), it is seen that

$$\frac{d}{dt} [V^T(t)V(t)] = 0 \quad (5)$$

Consequently

$$V^T(t)V(t) = \text{Const.} \quad (6)$$

and thus (I) has been proven.

Now when  $V(t_0)$  is orthogonal, then

$$V^T(t_0)V(t_0) = I$$

(where  $I$  denotes the identity matrix) and due to (6) also

$$V^T(t)V(t) = I$$

which proves assertion (II). ■

**Theorem I.2:** Any time varying orthogonal matrix,  $V(t)$ , satisfies the matrix differential equation

$$\dot{V}(t) = W(t)V(t) \quad (7)$$

where

$$W^T(t) = -W(t) \quad (8)$$

**Proof:** Since  $V(t)$  is orthogonal

$$\dot{V}(t) = \dot{V}(t)V^T(t)V(t) \quad (9)$$

Denote

$$W(t) = \dot{V}(t)V^T(t)$$

then (9) can be written as

$$\dot{V}(t) = W(t)V(t) \quad (10)$$

which is (7).

Using (10) we write

$$\begin{aligned}\dot{V}^T(t)V(t) + V^T(t)\dot{V}(t) &= V^T(t)W^T(t)V(t) + V^T(t)W(t)V(t) \\ &= V^T(t) [W^T(t) + W(t)] V(t)\end{aligned}\quad (11)$$

The left-hand side of (10) is the time derivative of  $V^T(t)V(t)$  hence (10) can be written as

$$\frac{d}{dt} [V^T(t)V(t)] = V^T(t) [W^T(t) + W(t)] V(t)\quad (12)$$

But

$$V^T(t)V(t) = I$$

hence the left-hand side of (11) is zero which implies that

$$W^T(t) = -W(t)$$

as stated in (8). This completes the proof. ■

In view of the preceding, it is realized that the problem we are concerned with is an extension of the three dimensional attitude determination problem and conversely, the latter is a special case of the problem at hand. It is interesting to investigate the correspondence of the various elements involved in three dimensional attitude determination with the eventual solution and features of our present problem. For this reason the pertinent background material of attitude determination will be reviewed in Section III following a formal definition of the problem in the next section. In Section IV we discuss a possible solution using Extended Euler Angles followed, in Section V, by an introduction of the chosen Extended Rodrigues Parameter solution. In Section VI we probe the issue of presenting angular rate in n-D and in Section VII we discuss numerical issues involved in the implementation of the solution. Numerical results are then presented and conclusions are drawn in Section VIII.

## II. PROBLEM STATEMENT

We state our problem as follows. Given the matrix differential equation

$$\dot{V}(t) = W(t)V(t)$$

in which  $W$  is a skew symmetric matrix and for which the initial matrix  $V(t_0)$  is known to be orthogonal, find the following:

- a)  $m=n(n-1)/2$  parameters which unambiguously define V,
- b) the differential equation needed to be solved in order to compute these parameters,
- c) the functional relations between the parameters and V which will enable the computation of V based on the parameters, and
- d) a simple algorithm to implement the solution of the differential equation as well as the computation of V.

### III. BACKGROUND IN THREE DIMENSIONAL SPACE

#### Euler Angles [2-6]

The best known parameters describing a 3-D rotation and the resulting transformation matrix are Euler Angles. Three such angles are necessary and sufficient to describe any transformation from one Cartesian coordinate system to any other one. There are 12 sequences of 3 right-hand Euler Angle rotation sequences. If for example one chooses the sequence z-y-x rotations by the respective angles p, t and f, then the corresponding differential equations of the Euler Angles are

$$\dot{p} = (w_y \sin f + w_z \cos f) / \cos t \quad (13.a)$$

$$\dot{t} = w_y \cos f - w_z \sin f \quad (13.b)$$

$$\dot{f} = w_x + \tan t (w_y \sin f + w_z \cos f) \quad (13.c)$$

where  $w_x$ ,  $w_y$  and  $w_z$  are the three components of the angular rate vector at which the final coordinate system turns with respect to the initial one when this vector is resolved in the final system. The transformation matrix, D, which transforms vectors from the initial coordinate system into the rotated one is computable using the solution of (13) in the following expression

$$D = \begin{vmatrix} \text{cp ct} & \text{sp ct} & \text{-st} \\ \text{-sp cf} & \text{cp cf} & \text{ct sf} \\ \text{+cp st sf} & \text{+sp st sf} & \\ \text{sp sf} & \text{-cp sf} & \text{ct cf} \\ \text{+sp st cf} & \text{+sp st cf} & \end{vmatrix}$$

where s denotes the sine and c denotes the cosine functions.

We note two shortcomings of this method. First, we run into a singularity problem as  $t$  approaches  $90^\circ$  or  $-90^\circ$  and, secondly we need to compute trigonometric functions. For this reason the use of Quaternions is usually preferred.

### Quaternion [2,5,6]

Quaternions consist of 4 elements; that is, the Quaternion is a 4 parameter rotation specifier. One parameter is, of course, superfluous but this is acceptable since, using Quaternions, the two aforementioned shortcomings, involved in the usage of Euler Angles, are eliminated. Denote the 4 elements of the Quaternion of rotation by  $q_0, q_1, q_2$  and  $q_3$  then the differential equation of the Quaternion elements is

$$\frac{d}{dt} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -w_x & -w_y & -w_z \\ w_x & 0 & w_z & -w_y \\ w_y & -w_z & 0 & w_x \\ w_z & w_y & -w_x & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (14)$$

The solution of (14) yields the components of the Quaternion which can be used to compute D as follows

$$D = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_3 q_1 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_3 q_1 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

The Quaternion of rotation is based on Euler's theorem which states that any orientation of a 3-D Cartesian coordinate system with respect to any reference system can be obtained by a single rotation of the initial coordinate system about an axis fixed in both systems. Let the positive direction (according to

the right-hand rule) of this axis be denoted by a unit vector  $\hat{f}$  and the rotation angle by  $f$ , then the components and the

magnitude of the rotation vector  $\hat{f}f$  (also known as Euler Vector) are used to define the Quaternion as follows

$$q_0 = \cos(f/2) \quad ; \quad q_1 = \sin(f/2) f_x / f$$

$$q_2 = \sin(f/2) f_y / f \quad ; \quad q_3 = \sin(f/2) f_z / f$$

were  $f_i$ ,  $i=x,y,z$ , are the 3 components of the rotation vector.

The Quaternion is, then, a 4 component element constructed on a 3 component vector.

### Rodrigues Parameters [7,8,6]

Another 3 parameter representation of 3-D rotation is due to Rodrigues [7]. Denote the parameters by  $g_1$ ,  $g_2$  and  $g_3$  then the differential equation which these parameters satisfy is

$$\frac{d}{dt} \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1+g_1^2 & g_3+g_1g_2 & -g_2+g_1g_3 \\ -g_3+g_1g_2 & 1+g_2^2 & g_1+g_2g_3 \\ g_2+g_1g_3 & -g_1+g_2g_3 & 1+g_3^2 \end{bmatrix} \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \quad (15)$$

The solution of (15) can, then, be used to compute D as follows

$$d = 1+g_1^2+g_2^2+g_3^2$$

$$D = \frac{1}{d} \begin{bmatrix} 1+g_1^2-g_2^2-g_3^2 & 2(g_1g_2-g_3) & 2(g_1g_3+g_2) \\ 2(g_1g_2+g_3) & 1-g_1^2+g_2^2-g_3^2 & 2(g_2g_3-g_1) \\ 2(g_1g_3-g_2) & 2(g_2g_3+g_1) & 1-g_1^2-g_2^2+g_3^2 \end{bmatrix}$$

The relationship between the Rodrigues Parameters and the rotation vector are

$$g_1 = \tan(f/2)f_x/f \quad ; \quad g_2 = \tan(f/2)f_y/f \quad ; \quad g_3 = \tan(f/2)f_z/f$$

Since both the Quaternion of rotation and Rodrigues Parameters are based in a similar manner on the rotation vector, there is a rather simple relationship between them; namely,  $g_i = q_i/q_0$   $i=1,2,3$ .

The preceding equations for the time change of the Rodrigues Parameter and for converting the parameters into D can be cast in matrix form as follows [9,10]. Define a G matrix such that

$$G = \begin{bmatrix} 0 & g_3 & -g_2 \\ -g_3 & 0 & g_1 \\ g_2 & -g_1 & 0 \end{bmatrix} \quad (16)$$

and, similarly a W matrix

$$W = \begin{bmatrix} 0 & w_3 & -w_2 \\ -w_3 & 0 & w_1 \\ w_2 & -w_1 & 0 \end{bmatrix} \quad (17)$$

then

$$\dot{G} = -\frac{1}{2}(I+G)W(I-G) \quad (18)$$

$$D = (I-G)(I+G)^{-1} \quad (19)$$

where  $I$  is the identity matrix. Like with the 3 parameter Euler Angle representation, here too singularity may occur whenever the size of the rotation vector reaches a magnitude of  $180^\circ$ .

After having discussed the possible solutions to the problem in 3-D we will consider, next, the possibility of extending these solutions to n-D (whenever mentioning n dimensional spaces we mean Euclidean spaces whose dimension  $n \neq 3$ ).

#### IV. POSSIBLE SOLUTIONS

##### Extended Euler Angles

When trying to solve our problem (as defined in Section II) the first question that comes to mind is: can the Euler Angle parametrization presented in the preceding section be extended to higher dimensional Euclidean spaces? As it turns out [11], Euler himself showed that this was possible. This was also shown later by Lagrange [12]. (See also Jacobi's observation on their and others' work [13]). However, the use of the Extended Euler Angles for  $n > 3$  is cumbersome since, for calculating  $V$ , the sine and cosine functions of  $m=n(n-1)/2$  angles must be computed, these functions have to be multiplied through in a long string of multiplications, and the resultant products have to be added and subtracted. For  $n=4$ , for example, the 1,1 element of  $V$  is

$$v_{1,1} = \cos a_1 \cos a_3 \cos a_5 + \sin a_1 \sin a_4 \sin a_5$$

and there are 16 elements, all equally long, in  $V$ . When compared with the simplicity of the solution which we will eventually choose the complexity of the present one will be striking. Moreover, to complete the algorithm it is necessary to find the differential equations governing the Extended Euler Angles and solve them. Merely finding the equations, let alone solving them, is a formidable task. As an example for the work involved in deriving those equations, consider the following



approach. Let the Extended Euler Angles be denoted by  $a_1, a_2, \dots, a_m$ . Denote the column vector whose elements are the these angles by  $\underline{a}$ . We may express  $V$  as a product of the individual matrices  $V(a_i)$  of the transformation matrix related to a single angle  $a_i$ , thus

$$V(\underline{a}) = \prod_{i=1}^m V(a_i) \quad (20)$$

Differentiation of (20) yields

$$\dot{V}(\underline{a}) = \sum_{j=1}^m a_j \left[ \prod_{i=1}^{j-1} V(a_i) \right] \frac{dV(a_j)}{da_j} \left[ \prod_{i=j+1}^m V(a_i) \right] \quad (21)$$

On the other hand (20) and (1) yield

$$\dot{V}(\underline{a}) = W \sum_{i=1}^m V(a_i) \quad (22)$$

equating the right-hand sides of (21) and (22) yields  $m$  equations

in  $a_j$ . After cumbersome manipulations we obtain the required  $m$  differential equations for  $a_j, j=1,2,\dots,m$  whose solution yields  $\underline{a}$ , the elements of which are needed in order to compute  $V$ . We conclude that finding the differential equations for the Extended Euler Angles, solving them, and then using the solutions to compute the corresponding  $V$  matrix, while possible, is indeed a formidable task which we reject in favor of the method which we will eventually select.

### Extended Quaternion

The use of the quaternion of rotation in 3-D is motivated by the following considerations. It does not suffer from singularities, it does not require the computation of trigonometric functions, it has a simple linear differential equation and a simple geometric interpretation related to the rotation vector. Finally, the only price paid for using it, is the need to deal with 4 (rather than 3) parameters. Because of these merits, one is motivated to try to extend the notion of quaternions to  $n$ -D. This approach though does not seem to yield a non-singular parametrization even if one is willing to use  $m+1$  parameters to define an extended quaternion.

Of the three 3-D parametrization methods reviewed in Section III only the Rodrigues Parameters are extendible to a compact easily implementable algorithm. This will be shown in the next section.

### V. EXTENDED RODRIGUES PARAMETERS

We start the presentation of this parametrization method in  $n$ -D with two lemmas which will be helpful in the ensuing.

**Lemma V.1:** Let  $A$  be an  $n \times n$  matrix, then the matrix  $(I+A)$  is invertible iff none of the eigenvalues of  $A$  is equal to  $-1$ .

**Proof:**

The eigenvalues  $b_i$ ,  $i=1,2,\dots,n$  of  $(I+A)$  are the roots of the polynomial

$$|(I+A) - bI| = 0 \quad (23)$$

which can be written as

$$|A - (b-1)I| = 0 \quad (24)$$

or

$$|A - aI| = 0 \quad (25)$$

where

$$a = b - 1 \quad (26)$$

The condition for  $(I+A)$  to be invertible is  $b_i \neq 0$ ,  $i=1,2,\dots,n$  or, in view of (26),  $a_i \neq -1$ ,  $i=1,2,\dots,n$ . But in view of (25),  $a_i$  are the eigenvalues of  $A$ . This ends the proof. ■

**Lemma V.2:** Let  $(I+A)^{-1}$  exist and let

$$B = (I-A)(I+A)^{-1}$$

then  $(I+B)$  is invertible.

**Proof:**

$$\begin{aligned} (I+B) &= I + (I-A)(I+A)^{-1} \\ &= (I+A)(I+A)^{-1} + (I-A)(I+A)^{-1} \\ &= 2(I+A)^{-1} \end{aligned}$$

Obviously,  $(I+A)^{-1}$  has an inverse which is  $(I+A)$ , thus

$$(I+B)^{-1} = (I+A)/2. \quad \blacksquare$$

With these lemma on hand we can proceed and prove the following theorem.

**Theorem V.1:** Let  $V$  be an  $n$ -th order orthogonal matrix with none of its eigenvalues equal to  $-1$  then

I) there exists a matrix  $G$  defined as follows

$$G = (I-V)(I+V)^{-1} \quad (27)$$

II)  $G$  is skew-symmetric

III)  $V$  is the following function of  $G$

$$V = (I-G)(I+G)^{-1} \quad (28)$$

IV) the rate of change of  $G$  is given by

$$\dot{G} = -\frac{1}{2}(I+G)W(I+G)^T \quad (29)$$

Proof: From lemma V.1 the matrix  $(I+V)$  has an inverse; thus  $G$  as defined in (27) exists. To show that  $G$  is skew-symmetric use (27) to write

$$G^T = (I+V)^{-T}(I-V)^T$$

where  $-T$  is the inverse of the transpose (or vice-versa). Using the last equation and the orthogonality of  $V$  we observe that

$$\begin{aligned} G^T &= (I+V^T)^{-1}(I-V^T) = (V^TV+V^T)^{-1}(V^TV-V^T) \\ &= [V^T(V+I)]^{-1}V^T(V-I) \\ &= (I+V)^{-1}VV^T(V-I) = (I+V)^{-1}(V-I) \\ &= -(I+V)^{-1}[2I-(I+V)] = -2(I+V)^{-1} + I \end{aligned} \quad (30)$$

Now

$$\begin{aligned} -2(I+V)^{-1} + I &= -2(I+V)^{-1} + (I+V)(I+V)^{-1} \\ &= [-2I+(I+V)](I+V)^{-1} \\ &= -(I-V)(I+V)^{-1} = -G \end{aligned} \quad (31)$$

Substitution of (31) into (30) yields the result  $G^T = -G$ , i.e.  $G$  is skew-symmetric.

From lemma V.2,  $(I+G)$  is invertible which gives legitimacy to the right-hand side of (28). To prove the truth of (28) re-write (30) as

$$G^T = I - 2(I+V)^{-1}$$

hence

$$G = I - 2(I+V^T)^{-1}$$

from which we obtain

$$I - G = 2(I + V^T)^{-1} \quad (32)$$

and

$$I + G = 2I - 2(I + V^T)^{-1} \quad (33)$$

We can further write

$$\begin{aligned} 2I - 2(I + V^T)^{-1} &= 2(I + V^T)(I + V^T)^{-1} - 2(I + V^T)^{-1} \\ &= 2V^T(I + V^T)^{-1} \end{aligned}$$

thus

$$I + G = 2V^T(I + V^T)^{-1}$$

and

$$(I + G)^{-1} = \frac{1}{2}(I + V^T)V \quad (34)$$

Substitution of (32) and (34) in the right-hand side of (28) yields the proof of III.

To prove (29) differentiate (27)

$$\begin{aligned} \dot{G} &= -\dot{V}(I + V)^{-1} - (I - V)(I + V)^{-1}\dot{V}(I + V)^{-1} \\ &= -[I + (I - V)(I + V)^{-1}]\dot{V}(I + V)^{-1} \end{aligned}$$

Substitute (27) in the last equation to obtain

$$\dot{G} = -(I + G)\dot{V}(I + V)^{-1}$$

Using (1) the last equation can be written as

$$\dot{G} = -(I + G)WV(I + V)^{-1}$$

Substitution of (28) into the last equation yields

$$\dot{G} = -(I + G)W(I - G)(I + G)^{-1}[I + (I - G)(I + G)^{-1}]^{-1} \quad (35)$$

The expression in the brackets can be written as follows

$$I + (I - G)(I + G)^{-1} = (I + G)(I + G)^{-1} + (I - G)(I + G)^{-1} = 2(I + G)^{-1}$$

therefore (35) can be written as

$$\dot{G} = -(I + G)W(I - G)(I + G)^{-1}[2(I + G)^{-1}]^{-1} = -\frac{1}{2}(I + G)W(I - G)$$

and since  $G$  is skew-symmetric the last equation can be written also as

$$\dot{G} = -\frac{1}{2}(I+G)W(I+G)^T$$

which ends the proof. ■

Note, from lemma V.1, that the condition for the invertibility of  $(I+G)$  is that it has no eigenvalues at  $-1$ , which is analogous to the condition for  $(I+V)$  to be invertible, i.e. that  $V$  has no eigenvalues at  $-1$ . However while  $V$  always exists,  $G$  does not exist when  $V$  has an eigenvalue at  $-1$ . The parametrization of  $V$  by  $G$  fails when the latter is the case. However this can be overcome as will be shown in Section VII.

The parametrization of  $V$  by the Extended Rodrigues Parameters is  $n$ -dimensional since the foregoing proofs were not restricted to any value of  $n$ , nor did they hinge on a rotation vector or any other geometric quality in  $n$ -D. In fact, the Extended Rodrigues Parameters, which are the elements of  $G$ , are the answer to the first three parts of our problem as posed in Section II. That is we found  $m$  parameters which define the  $n$ -dimensional orthogonal matrix,  $V$ . We also found a first order differential equation for  $G$ , and we showed how to calculate  $V$ , once  $G$  is found.

What is needed to fully answer our problem is a simple algorithm to implement the solution; this will be presented in Section VII. For now, after having obtained a parametrization in  $n$ -D, we are prepared to discuss the meaning of the skew-symmetric matrix,  $W$ , its geometric interpretation, and the difference between  $W$  in 3 and in  $n$ -D.

## VI. ANGULAR RATE IN $n$ -D

Recall (1)

$$\dot{V}(t) = W(t)V(t) \tag{1}$$

The matrix  $V$  can be viewed as a transformation matrix which transforms vector components in an  $n$ -D Euclidean space. In particular it transforms a set of unit vectors, which form a Cartesian coordinate system, to another such set. Let us denote the former as the initial coordinate system and the latter as the final one. The rows of  $V$  are components of unit vectors of the initial set resolved in the final Cartesian coordinate system such that  $v_{i,j}$  is the  $i$ -th component in the final system of the  $j$ -th unit vector of the initial coordinate system. From (1)

$$\dot{v}_{i,j} = \sum_{k=1}^n w_{i,k} v_{k,j}$$

hence  $w_{i,k}$  is the relative weight that the  $k$ -th component in the final system, of a unit vector in the initial system, has on the rate of change of the  $i$ -th component in the final system of the same unit vector in the initial system. Note that this weight is independent of  $j$ ; i.e. of which unit vector in the initial system we consider. To give  $w_{i,j}$  a more descriptive interpretation and to see the role of  $W$  more clearly, consider the 3-D case where, for example

$$\dot{v}_{3,1} = w_{3,1}v_{1,1} + w_{3,2}v_{2,1} \quad (36)$$

(note that the term  $w_{3,3}v_{3,1}$  was dropped since  $w_{3,3} = 0$  for skew-symmetric  $W$ ). In 3-D (36) can be written as

$$\dot{v}_{3,1} = w_2 v_{1,1} - w_1 v_{2,1} \quad (37)$$

where  $w_1$  and  $w_2$  are the respective angular rates at which the final coordinate system instantaneously rotates about its 1 and 2 axes. The components  $w_i$ ,  $i=1,2,3$ , are those of the 3-D angular rate vector describing the instantaneous rotation of the final system. In 3-D  $w_i$  is also the angular rate at which the  $j$  axis turns towards the  $k$  axis, and so on in a cyclic manner for  $w_j$  and  $w_k$ . Indeed a comparison between (36) and (37) reveals that

$$w_2 = w_{3,1}$$

$$-w_1 = w_{3,2}$$

We conclude that the following can be said about  $W$  in 3-D

- (A) The elements of  $W$  are angular rates.
- (B) Each components of  $W$  is a rate of turn of one coordinate axis towards another such that  $w_{p,q}$  is the angular rate at which axis  $p$  turns towards axis  $q$ . Obviously,  $w_{p,q} = -w_{q,p}$ .
- (C) Both the  $p$  and the  $q$  axes turn at the angular rate  $w_{p,q}$  about the third axis  $r$ .
- (D) The elements of  $W$  are components of an angular rate vector.

When we turn now to  $n$ -D, we realize that the preceding observation cannot be fully extended from 3 to  $n$ -D. In  $n$ -D  $W$  has  $m=n(n-1)/2$  independent components such that the elements of  $W$  cannot be components of a rate vector whose number is necessarily only  $n$ . We cannot, therefore, consider the elements of  $W$  as angular rates about (coordinate) axes. Consequently, of the four features of the elements of  $W$  in 3-D, mentioned above, the only ones which also prevail in  $n$ -D are (A) and (B).

Realizing that the angular rates in  $n$ -D cannot be described by a vector, one is motivated to examine the possibility of

expressing the angular rate by a tensor. To accomplish that, choose one, say the  $i$ -th, column of  $V(t)$  and the  $i$ -th column of  $V(t)$  and denote them correspondingly by  $\dot{\bar{v}}$  and  $\bar{v}$  such that  $\dot{\bar{v}} = [\dot{\bar{v}}_1, \dot{\bar{v}}_2, \dots, \dot{\bar{v}}_n]^T$  and  $\bar{v} = [v_1, v_2, \dots, v_n]^T$ . Using their components express them as vectors in the same arbitrarily chosen coordinate system such that

$$\dot{\bar{v}} = \bar{i}_1 \dot{\bar{v}}_1 + \bar{i}_2 \dot{\bar{v}}_2 + \dots + \bar{i}_n \dot{\bar{v}}_n$$

where  $\bar{i}_1, \bar{i}_2, \dots, \bar{i}_n$  are unit vectors along the coordinate axes 1, 2, ...,  $n$  respectively. Similarly

$$\bar{v} = \bar{i}_1 v_1 + \bar{i}_2 v_2 + \dots + \bar{i}_n v_n$$

Define a tensor of the second rank,  $\bar{W}$ , using the elements of  $W$  as follows

$$\begin{aligned} \bar{W} = & \bar{i}_1 \bar{i}_1^0 + \bar{i}_1 \bar{i}_2 w_{1,2} + \dots + \bar{i}_1 \bar{i}_n w_{1,n} \\ & + \bar{i}_2 \bar{i}_1 w_{2,1} + \bar{i}_2 \bar{i}_2^0 + \dots + \bar{i}_2 \bar{i}_n w_{2,n} \\ & \vdots \\ & + \bar{i}_n \bar{i}_1 w_{n,1} + \bar{i}_n \bar{i}_2 w_{n,2} + \dots + \bar{i}_n \bar{i}_n^0 \end{aligned}$$

then obviously

$$\dot{\bar{v}} = \bar{W} \bar{v}$$

that is, when the angular rate components are treated as elements of a tensor of the second rank, (1) is fully satisfied. A tensor of the second rank is also known as dyadic [14].

The fact that the angular rate in 3-D is basically a tensor is known [8,15] but is not reflected in the applied literature. The reason for it stems, perhaps, from the unique possibility to express angular rates in 3-D by a vector such that its description as a tensor might have been perceived merely as a philosophical formalism. (Even when treated as a tensor, the angular rate is usually that of a 3-D coordinate system). Indeed, the creation, in 3-D, of the so called "vector cross-product matrix" based on the angular velocity vector is conceived as a useful gimmick rather than a restoration of the true mathematical description of the angular rate. So far, the consideration of angular rates in dimensions higher than 3 probably was not required nor known. Thus it was not recognized that in higher dimensions the angular rate cannot be described

by a vector but must be described as another entity, and that the ability to describe it in 3-D by a vector is just a matter of good fortune. (In fact, even in 2-D the angular rate is not truly expressible as a vector. This is evident when we note that the expression of rotation in a plane by a vector normal to it is necessarily a 3-D expression. The correct and only 2-D expression is

$$\dot{\bar{v}} = (\bar{i}_1 \bar{i}_2 w - \bar{i}_2 \bar{i}_1 w) \bar{v}$$

or

$$\begin{bmatrix} \dot{\bar{v}}_1 \\ \dot{\bar{v}}_2 \end{bmatrix} = \begin{bmatrix} 0 & w \\ -w & 0 \end{bmatrix} \begin{bmatrix} \bar{v}_1 \\ \bar{v}_2 \end{bmatrix}$$

where the first expression is in a tensor form and the second is in a matrix form). Another possible cause for the disregard of the fact that angular rate is a tensor stems from the fact that the tensor of the second rank; that is, the dyadic, is replaceable by a matrix (as demonstrated in the last 2-D representation and in equation 1). Therefore all practical work in any dimension can be carried out without resorting to the tensor concept.

After having cleared the issue of angular rate representation we are prepared to consider the implementation of the algorithm for solving (1) using the Extended Rodrigues Parameters, thereby solving our problem in its entirety.

## VII. NUMERICAL IMPLEMENTATION

Recall the differential equation (1)

$$\dot{\bar{v}} = W \bar{v} \quad (1)$$

in which W is given. We wish to solve (1) using the extended Rodrigues Parameters. The solution process requires first the solution of

$$\dot{\bar{G}} = -\frac{1}{2}(\bar{I} + \bar{G})W(\bar{I} + \bar{G})^T \quad (29)$$

and then the computation of V according to (28)

$$\bar{v} = (\bar{I} - \bar{G})(\bar{I} + \bar{G})^{-1} \quad (28)$$

There are two caveats which we have to be alerted to. One of them is the non-existence of G when V has an eigenvalue at -1, and the other is the need to invert the matrix (I+G), which may be so



burdensome as to render the whole approach inefficient in comparison with the direct solution of (1). The first problem can be easily avoided if we can keep the elements of  $G$  small, for then, as can be readily seen from (28),  $V$  is close to  $I$  whose eigenvalues are all equal to  $+1$ . That is, if we are free to control its size, we can always choose  $G$  so small as to make the eigenvalues of  $V$  as close to  $+1$  (and thus as far from  $-1$ ) as we wish. Indeed, we are able to control the magnitude of  $G$ . The ability to do it is based on the following proposition.

Proposition: Given the differential equation of (1)

$$\dot{V}(t) = W(t)V(t) \quad (1)$$

with the initial condition  $V(t_0)$  where  $V(t_0)$  is orthogonal, then  $V(t)$ , the solution of (1) at time  $t > t_0$ , can be written as a product of two matrices as follows

$$V(t) = V(t, t_0)V(t_0) \quad (38)$$

where  $V(t, t_0)$  is the solution of (1) at time  $t$  given the initial condition  $V(t_0, t_0) = I$ .

Proof: Since  $V(t_0)$  is orthogonal it always has an inverse. Therefore one can always compute a matrix

$$V(t, t_0) = V(t)V^T(t_0) \quad (39)$$

such that (38) holds. Now if (38) is differentiated with respect to time the following is obtained

$$\dot{V}(t) = \dot{V}(t, t_0)V(t_0)$$

Equating the right-hand side of the last equation to that of (1) and using (38) results in

$$\dot{V}(t, t_0)V(t_0) = W(t)V(t, t_0)V(t_0)$$

Since  $V(t_0)$  is invertible, the last equation yields

$$\dot{V}(t, t_0) = W(t)V(t, t_0)$$

hence  $V(t, t_0)$  solves (1). Finally setting  $t$  in (39) to  $t_0$  results in

$$V(t_0, t_0) = I$$

which ends the proof. ■

In computing  $V(t)$  we make use of the last proposition when we consider  $V(t)$  as a product of  $V(t, t_0)$  and  $V(t_0)$  as follows

$$V(t) = V(t, t_0)V(t_0) \quad (40)$$

and instead of computing  $V(t)$  directly we compute  $V(t, t_0)$  from time  $t_0$  to  $t$  and then use (40) to compute  $V(t)$ . Actually instead of computing  $V(t, t_0)$  we use (29) to compute  $G$ , the parametrization of  $V(t, t_0)$ , from time  $t_0$  to  $t$  with the initial condition  $G(t_0) = 0$  which corresponds to  $V(t, t_0) = I$ . The computation of  $G$  is stopped periodically at, say,  $t_1$  and  $V(t_1)$  is computed according to (28) yielding

$$V(t_1, t_0) = [I - G(t_1)][I + G(t_1)]^{-1}$$

and then  $V(t_1)$  is computed using (40) as follows

$$V(t_1) = V(t_1, t_0)V(t_0)$$

Next the computation of  $V(t)$  proceeds into the following time interval using the same algorithm that produced  $V(t_1)$  once  $V(t_0)$  was given. We start, of course, with the initial condition  $G(t_1) = 0$  which corresponds to  $V(t, t_1) = I$ . Using this algorithm we proceed to compute  $G$  and  $V$  at times  $t_2, t_3, \dots, t_k$ . By properly choosing the size of the intervals  $t_2 - t_1, t_3 - t_2, \dots, t_k - t_{k-1}$  we can impose an upper bound on  $G$  which can practically be as small as we wish. We term the operation of resetting the value of  $V$  and  $G$  at the beginning of an interval reset operation.

The foregoing policy rids us of the singularity problem. In fact, if singularity were the only issue, one can choose the time intervals  $t_i - t_{i-1}$  quite large and still not encounter singularity. However, we are still left with the second problem mentioned before; namely, the inversion of  $[I + G(t_i)]$ . We overcome this problem by approximating the inverse without really performing any matrix inversion. Before discussing the options for approximating this inverse we list without proof two well known theorems (e.g. Ref. 16 p.129) needed in the ensuing.

**Theorem VII.1:** Let  $G$  be a square matrix then the

series  $\sum_{i=0}^{\infty} (-1)^i G^i$  converges to  $(I + G)^{-1}$  iff all the eigenvalue of  $G$  lie inside the unit circle about the origin of the complex plane.

**Theorem VII.2:** Denote the elements of the  $n \times n$  matrix  $G$  by  $g_{i,j}$ . If the sums

$\sum_{j=1}^n |g_{i,j}| \quad i=1, 2, \dots, n$   
are all less than 1, or if

$$\sum_{i=1}^n |g_{i,j}| \quad j=1, 2, \dots, n$$

are all less than 1, then all the eigenvalues of  $G$  lie inside the unit circle about the origin of the complex plane.

The algorithm we use to approximate the inverse of  $(I-G)$  is based on the fact that if the matrix  $X_i$  is a good approximation of the inverse of some matrix  $A$  then a better approximation,  $X_{i+1}$ , can be obtained using the Newton-Raphson-type iteration [16 p.52, 17]

$$X_{i+1} = X_i(2I - AX_i) \quad (41)$$

This algorithm converges if and only if the eigenvalues of  $I-AX_i$  are all of absolute value less than 1 [16, p.52]. If indeed  $X_i$  is almost the inverse of  $A$  then this condition is met. If now  $V$  is computed without reset taking place at the end of the previous time increment, then we use as a first approximation of  $[I+G(t)]^{-1}$ , the value used as an inverse at the previous time point. This is based on the presumption that the time increments of the integration are small enough such that the change of the inverse is small too, hence its previous accurate value can serve now as an approximate value. If, however, reset did take place at the previous time point then  $G$  was set to zero and the previous inverse of  $I+G$  is simply  $I$ . For the sake of computation reduction it is desired to keep at minimum the number of iterations used to compute an accurate inverse. Normally one iteration is sufficient. However, when reset takes place and consequently the previous inverse of  $I+G$  (i.e. the inverse of  $I+G_0$  for  $G_0 = 0$ ) is taken as  $I$  then a single iteration produces

$$(I+G_1)^{-1} \sim I-G_1 \quad (42)$$

where  $G_1$  is  $G$  at the present time. If, however, we enter the iteration with the value  $X_0 = I-G_1$  then, due to the quadratic convergence characteristic of the process, a single iteration produces

$$(I+G_1)^{-1} \sim I-G_1+G_1^2-G_1^3 \quad (43)$$

obviously the approximate inverse given in (43) is more accurate than that of (42) since it contains more terms of the series which expresses the inverse of  $(I+G_1)$ . Note that the series generated by (41) converges since due to the reset operation,  $G$  is kept at a very small value such that the condition of theorem VII.2 is met. Thus the eigenvalues of  $G$  are in the unit circle which, in view of theorem VII.1, assures convergence.

Another point of interest is the ability to use an alternate equation for computing  $G$ . From (31) it is obvious that

$$2(I+V)^{-1} = I+G$$

which yields

$$V = 2(I+G)^{-1} - I \quad (44)$$

The computation of  $V$  using (44) is simpler than when (28) is used. However, if the reset operation took place at the previous time point then the use of (28) at the present time point yields better results. This is evident in particular when the previous inverse,  $(I+G_{i-1})^{-1}$ , is approximated by  $I$ . In this case the use of (41) yields the approximation of (42) for which the use of (44) yields

$$V(t_i, t_{i-1}) = I - 2G_i$$

whereas the use of (28) yields

$$V(t_i, t_{i-1}) = I - 2G_i + G_i^2$$

which is more accurate than the preceding result. Even when the approximation of (43) is used, the use of (28) yields better results than that obtained using (44). Then, however, the difference is smaller since the term of the series which is being added is smaller than the added term in the previous case which was  $G_i^2$ . If, of course, an exact inverse is used then the use of (44) rather than (28) is preferable since then the computation of  $V(t_i, t_{i-1})$  is simplified without the penalty of accuracy degradation.

The algorithm which results from the preceding considerations is shown in Table I. Note that (28) rather than (44) is implemented for the reasons discussed above.

If one chooses to perform reset after each integration step of  $G$  then the computation of  $V(t_i, t_0)$  as given in Table I produces

$$V(t_i, t_0) = I - 2G_i + 2G_i^2 - 2G_i^3 + G_i^4 \quad (45)$$

where  $G_i = G(t_i)$ . This is a truncated series of the expression for  $V$  as a function of  $G$  given in (28) with the special feature that the last term in the series lacks the multiplier 2. A more computationally efficient algorithm than that is

$$V(t_i, t_0) = I - G_i\{2I - G_i[2I - G_i(2I - G_i)]\} \quad (46)$$

or better yet, if  $G_i^4$  is added to (45) to generate the true truncation of the series expansion of  $(I-G_i)(I+G_i)^{-1}$  then the new expression can be written as

$$V(t_i, t_0) = I - 2G_i\{I - G_i[I - G_i(I - G_i)]\} \quad (47)$$

Consequently one can use either the algorithm of Table I as is or compute  $V(t_i, t_0)$  using either (45) or (46) or (47). These, however are not the only possible variants of the algorithm. As a result of the discussions presented in this section it is clear that one has the following additional choices:

Table I

Given:  $V(t_0) = V_0$  and  $W(t)$

- 
- (1) Set the initial condition  $G(t_0) = 0$ .
  - (2) Solve  $\dot{G}(t) = -\frac{1}{2}[I+G(t)]W(t)[I+G(t)]^T$  from  $t_0$  to  $t_i$ .
  - (3) If reset didn't take place at the end of the preceding cycle, go to (4).  
     compute  $X_i = I - G(t_i)$  and go to (5).
  - (4) Compute  $X_i = X_{i-1}^*$
  - (5) Compute  $A_i = I + G(t_i)$  and  $X_i^* = X_i(2I - A_i X_i)$
  - (6) If reset is not requested go to (8).
  - (7) Perform a reset as follows. If reset didn't take place at the end of the preceding cycle compute (b).  
     (a)  $V(t_i, t_0) = X_i X_i^*$  and go to (c).  
     (b)  $V(t_i, t_0) = [I - G(t_i)] X_i^*$   
     (c)  $V(t_i) = V(t_i, t_0) V(t_0)$   
     (d) set  $t_0 = t_i$
  - (8) If the current time is equal to the final time go to step (10):
  - (9) If (7) was executed go to (1). Otherwise go to (2) and increase all indices by 1.
  - (10) Stop.

- Perform or not perform resets.
- Use more terms of the series

$$V(t_i, t_{i-1}) = I + 2 \sum_{n=1}^{\infty} (-1)^n G_i^n$$

- Use either (28) or (44) to compute  $V(t_1, t_0)$  from  $G_i$ .

The choices should correspond to the particular problem on hand.

As an example we ran a 4th dimensional case where

$$V(0) = I \quad ; \quad W(t) = \begin{bmatrix} 0 & -0.1 & -1.0 & -7.5 \\ 0.1 & 0 & 3.0 & 0 \\ 1.0 & -3.0 & 0 & -0.9 \\ -7.5 & 0 & 0.9 & 0 \end{bmatrix} * \sin(6.28t)$$

the initial time  $t_0 = 0$ .

the final time  $t_f = 1_{\text{sec}}$

the integration time  $dt = 0.001_{\text{sec}}$

The algorithm used in the solution of  $V$  was the one given in Table I where reset was performed after each integration step. Equation (1) was solved to yield a reference with which the algorithm output was compared. The reference matrix was denoted by  $V_r$  and the one generated by the algorithm was denoted by  $V$ . The integration routine which was used to solve the differential equation for  $V_r$  as well as for  $G$  was a 4-th order Runge-Kutta routine. The difference matrix between the two solutions was computed and denoted by  $E = V - V_r$ . A scalar which constitutes a measure of the size of the error was defined as follows

$$e = [\text{Tr}\{EE^T\}]^{1/2}$$

The scalar  $e$  is the square root of the sum of the squares of the elements of  $E$ . The results at  $t = 0.5_{\text{sec}}$  were:

$$V_r = \begin{bmatrix} -.72765515\text{E}+00 & .15285696\text{E}+00 & -.24387237\text{E}+00 & -.62263874\text{E}+00 \\ .10217642\text{E}-01 & .58373643\text{E}+00 & .79194147\text{E}+00 & -.17881859\text{E}+00 \\ -.13935294\text{E}+00 & -.79737729\text{E}+00 & .53481405\text{E}+00 & -.24237192\text{E}+00 \\ .67156112\text{E}+00 & -.87171959\text{E}-02 & -.16531458\text{E}+00 & -.72221933\text{E}+00 \end{bmatrix}$$

# V

```

-.72765512E+00   .15285696E+00   -.24387236E+00   -.62263872E+00
.10217638E-01   .58373643E+00   .79194147E+00   -.17881859E+00
-.13935294E+00   -.79737729E+00   .53481405E+00   -.24237191E+00
.67156110E+00   -.87171923E-02   -.16531458E+00   -.72221930E+00

```

# E

```

.29723949E-07   -.18713478E-09   .83326148E-08   .24813968E-07
-.35405291E-08   -.25268088E-09   -.11170641E-08   .71958844E-09
.42739559E-09   .12413215E-08   .86561824E-09   .83593105E-08
-.25932268E-07   .35672249E-08   .95984923E-09   .29599694E-07

```

e = .56724776E-07

As mentioned earlier the algorithm of Table I with a reset at the end of each integration cycle amounts to the use of (46) in the computation of  $V(t_i, t_0)$ . As suggested, (47) can be used instead. In Table II we show a comparison between the use of (46) and (47) for different series lengths. The table presents the error measure, e, for the two series truncated after different powers, n, of G. The error measure was recorded at  $t = 0.5_{\text{sec}}$  for at that point, which is half the period of the oscillating W, the value of e is the highest during the first period, i.e., in the domain  $0. < t < 1_{\text{sec}}$ . As can be seen from the results, algorithm 2 is superior. It can be also seen that there is a distinct power

Table II

	n				
	1	2	3	4	5
1. $V(t_i, t_0) = I - 2G_i + 2G_i^2 - \dots G_i^n$	.17 E 01	.52 E-02	.17 E-04	.57 E-07	.13 E-09
2. $V(t_i, t_0) = I - 2G_i + 2G_i^2 - \dots 2G_i^n$	.10 E-01	.34 E-04	.11 E-06	.33 E-09	.63 E-10

beyond which the addition of more terms yields little return. In view of these conclusions we recommend the use of the algorithm listed in Table III which in fact was used in the first example.

Table III

Given:  $V(t_0) = V_0$  and  $W(t)$

=====

- (1) Initialize  $i = 0$
- (2) Set the initial condition  $G(t_i) = 0$ .
- (3) Solve  $\dot{G}(t) = -\frac{1}{2}[I+G(t)]W(t)[I+G(t)]^T$   
from  $t_i$  to  $t_{i+1}$ .
- (4) Compute
$$V(t_{i+1}, t_i) = I - 2G_{i+1}\{I - G_{i+1}[I - G_{i+1}(I - G_{i+1})]\}$$
$$V(t_{i+1}) = V(t_{i+1}, t_i)V(t_i)$$
- (5) If the current time is smaller than the final time go back to step (2) and increase all indices by 1, otherwise STOP.

#### VIII. CONCLUSIONS

This work addressed the problem of solving the first order differential equation, which every orthogonal matrix satisfies, using the minimum number of parameters necessary to uniquely determine the matrix. The major question was: which are the parameters that do determine this matrix. The other questions were: what are the differential equation which one has to solve in order to find the parameters, and: once the parameters are found, how to use them in order to find the corresponding matrix. All these questions were answered and several algorithms for computing the orthogonal matrix via the parameters were suggested and investigated.

In search for solutions the familiar special 3-D case was examined with the purpose of extending the methods used there to the general n-D case. Accordingly, the first thought that came to mind was the idea of extending the concept of Euler angles to the n-D case. It turned out that, although not well known, Euler himself succeeded in using Euler angles to parametrize higher dimensional orthogonal matrices. Euler, however, was not concerned with the dynamic case; that is, with the differential equation which describe their change in time (neither did he do it for the 3-D case). Lagrange improved Euler's approach and presented it in the first edition of his book on analytic



mechanics. We did not adopt this approach because of the multitude of the trigonometric functions that one has to trace and compute and because of the complexity of the differential equations which describe the time change of the extended Euler angles.

Another popular 3-D parametrization which was considered was the quaternion of rotation. This approach did not seem to lead to any solution and was abandoned. The last parametrization which was examined was that of Rodrigues. In the vast literature on 3-D methods the elements of this parametrization are known as: Gibbs vector or Cayley-Rodrigues parameters; however, the presentation of these parameters by Rodrigues in 1840 [7] preceded the work of Cayley who, as a matter of fact, credits Rodrigues with their discovery [18,19]. Rodrigues' work certainly preceded that of Gibbs who first published his research of these parameters in 1884 (see Ref. 9, p. 17). Although it seems that Rodrigues was the first one to present them, it turns out, as noted by Jacobi [13] and by Roberson [8], that even these parameters were first presented by Euler [20] although in a different form. Ironically, while Rodrigues based his development on the, by now, very famous theorem of Euler [21], Euler himself was not aware of the possible use of his own theorem in the derivation of these parameters. (The theorem states that any final sequence of 3-D rotations can be represented by just one rotation about a single fixed axis)\*. It is, however, Rodrigues who developed the parameters in their present known form. For this reason we refer to their extension to n-D as the Extended Rodrigues Parameters. It was shown that the parameters can be conveniently extended to n-D. In fact there is nothing that limit their validity to 3-D only. Indeed, the theorems used in the presentation of the Extended Rodrigues Parameters in this work do not assume any restriction on the dimensionality of the space in which they are used.

Projecting the 3-D concepts into n-D raises the question of the correct mathematical representation of angular rates in spaces whose dimension is not 3. It is shown that angular rate has to be represented by a tensor of the second rank, also known as dyadic. The ability to represent angular rate as a vector is unique to 3-D. This fact, while known before, was not paid sufficient attention because the vectorial representation satisfied the intuition and the practical needs of its users. In other dimensions the vectorial representation fails and the use of the dyadic representation is required. Finally it should be

pointed out that when, as in our case, matrices are used, the skew-symmetric dyadic which represents angular rate in n-D is simply represented by a skew symmetric matrix.

---

\* As noted by Jacobi, Lagrange too presented this theorem in the first edition of his book on analytic mechanics [12] but dropped it as well as the treatment of rotations from the second edition of this book.

### Acknowledgment

The authors wish to thank Ms Constance Carter of the US Library of Congress for her unyielding search which culminated with the discovery of the original Russian and French journals in which Euler's and Rodrigues' papers were published respectively.

### REFERENCES

- [1] Y. Oshman and I. Y. Bar-Itzhack, "Eigenfactor Solution of the Matrix Riccati Equation - a Continuous Square Root Algorithm," IEEE Transactions on Automatic Control, Vol. AC-30, No. 10, October 1985, pp. 971-978.
- [2] P. L. Pio, "Euler Angle Transformations," IEEE Transactions on Automatic Control, Vol. AC-11, No. 5, October 1960, pp. 707-715.
- [3] H. Goldstein, Classical Mechanics, 2nd edition, Addison-Wesley, Reading Mass, 1980.
- [4] E. T. Whittaker, A Treatise on the Analytical Dynamics of Particles and Rigid Bodies, 4th edition, Cambridge University Press, England, 1964.
- [5] T. F. Wiener, "Theoretical Analysis of Gimballess Inertial Reference Equipment Using Delta-Modulated Instruments," Doctoral Dissertation, M.I.T. Cambridge Mass, 1962.
- [6] J. R. Wertz (ed.), Spacecraft Attitude Determination and Control, D. Reidel, Dordrecht Holland, 1978.
- [7] M. O. Rodrigues, "Des Lois Geometriques Qui Regissent les Deplacement d'un Systeme Solide dans L'espace, et de la Variation des Coordonnees Provenant de ces Deplacements Consideres Independamment des Causes qui Peuvent les Produire," J. de Mathematiques Pures et Appliquees, (Liouville), Vol. 5, 1840, pp. 380-440.
- [8] R. E. Roberson, "Kinematic Equations for Bodies Whose Rotation is Described by the Euler-Rodrigues Parameters," AIAA Journal, Vol.6, No. 5, January 1968, pp. 916,917.
- [9] J. W. Gibbs Scientific Papers, Vol. II, Dover Publication Inc., New-York, 1961, p. 65.

- [10] J. Stuelpnagel, "On the Parametrization of the Three-Dimensional Rotation Group," SIAM Review, Vol. 6 No. 4, October 1964, pp. 422-430.
- [11] L. Euler, "Problema Algebraicum ob Affectiones Prorsus Singulares Memorabile," Novi Comm. Acad. Sci. Petrop., Vol. 15, 1770, pp. 75-126
- [12] J. L. Lagrange, *Mecanique Analytique*, 1st edition, part II, Section VI: Sur la Rotation des Corps, Chez La Veuve de Saint, Libraire, Rue du Foin S. Jacques, 1788, p.353.
- [13] C. G. J. Jacobi, "Bemerkungen zu Einer Abhandlung Euler's uber die Orthogonale Substitution," in C. G. J. Jacobi's *Gesammelte Werke*, 2nd edition, Vol. III, Chelsea Publishing Co. New-York, 1969, pp. 601-609.
- [14] L. P. Page, *Introduction to Theoretical Physics*, 3rd edition, Van Nostrand, Princeton NJ, 1952, pp. 46-55.
- [15] A. M. Goodbody, *Cartesian Tensors: with Application to Mechanics, Fluid Mechanics and Elasticity*, Ellis Horwood Ltd. Publishers, Chichester UK, 1982, pp. 221, 222.
- [16] J. R. Westlake, *A Handbook of Numerical Matrix Inversion and Solution of Linear Equations*, John Wiley and Sons, New-York, 1968.
- [17] G. E. Forsyth and C. B. Moler, *Computer Solution of Linear Algebraic Systems*, Prentice-Hall, Englewood Cliffs NJ, 1967, p.78.
- [18] A. Cayley, "On the Motion of Rotation of a Solid Body," Cambridge Mathematical Journal, Vol. III, 1843, pp. 224-232, also in *The Collected Mathematical Papers of Arthur Cayley*, Vol. I, The Cambridge University Press, 1889. Johnson Reprint Corp. New York, 1963, pp.28-35.
- [19] A. Cayley, "On certain Results Relating to Quaternions," Cambridge Mathematical Journal, Vol. III, 1843, pp. 141-145, also in *The Collected Mathematical Papers of Arthur Cayley*, Vol. I, The Cambridge University Press, 1889. Johnson Reprint Corp. New York, 1963, pp.123-126.
- [20] L. Euler, "Nova Methodus Motum Corporum Rigidorum Determinandi," Novi Comm. Acad. Sci. Petrop., Vol. 20, 1775, pp. 208-238.
- [21] L. Euler, "Formulae Generales pro Translatione Quacunque Corporum Rigidorum," Novi Acad. Sci. Petrop., Vol. 20, 1775, pp. 189-207.