Notes to Accompany the Slides

for *Spatial Vector Algebra*
by Roy Featherstone*

Preamble The course is aimed at people who already know how to do elementary rigid-body dynamics using 3D vectors. It follows the treatment of spatial vectors appearing in the book *Rigid Body Dynamics Algorithms*, which should be consulted for more details. Another source that may be useful is the *Springer Handbook of Robotics*, Chapter 2. Other materials relating to this course can be found on Dr. Featherstone's web site, including software to implement spatial vector arithmetic. At the time of writing, the URL is "`http://users.rsise.anu.edu.au/-~roy/`". If that doesn't work, try Google.

Slide 3 The scalar product can be written either $\boldsymbol{m} \cdot \boldsymbol{f}$ or $\boldsymbol{f} \cdot \boldsymbol{m}$. Both expressions mean the same. However, neither M^6 nor F^6 has an inner product, so the two expressions $\boldsymbol{m} \cdot \boldsymbol{m}$ and $\boldsymbol{f} \cdot \boldsymbol{f}$ are not defined. Formally, this scalar product makes F^6 the dual vector space of M^6, and vice versa.

Slide 4 We shall use the term *abstract vector* to refer to the quantity that is represented by a coordinate vector. On this slide, $\boldsymbol{m}$, $\boldsymbol{f}$, $\boldsymbol{d}_1 \dots \boldsymbol{d}_6$ and $\boldsymbol{e}_1 \dots \boldsymbol{e}_6$ are abstract vectors, while $\underline{\boldsymbol{m}}$ and $\underline{\boldsymbol{f}}$ are the coordinate vectors that represent $\boldsymbol{m}$ and $\boldsymbol{f}$ in the bases $\{\boldsymbol{d}_1, \dots, \boldsymbol{d}_6\}$ and $\{\boldsymbol{e}_1, \dots, \boldsymbol{e}_6\}$, respectively. The elements of a basis are always abstract vectors.

On this and some other slides, the symbols denoting coordinate vectors have been underlined in order to distinguish them from symbols denoting abstract vectors. This notational device is used only in the few places where it helps the discussion.

Slide 5 A reciprocal basis is also called a dual basis. If we name the bases $D = \{\boldsymbol{d}_1, \dots, \boldsymbol{d}_6\}$ and $E = \{\boldsymbol{e}_1, \dots, \boldsymbol{e}_6\}$, then the basis pair (D, E) defines a single coordinate system covering both M^6 and F^6, in which D defines the coordinates in M^6 and E defines the coordinates in F^6. We call this a *dual coordinate system.* Spatial vectors are always expressed in dual coordinate systems. Their usefulness is exactly the fact that if $\underline{\boldsymbol{m}}$ and $\underline{\boldsymbol{f}}$ represent $\boldsymbol{m}$ and $\boldsymbol{f}$ in a dual coordinate system then $\boldsymbol{m} \cdot \boldsymbol{f} = \underline{\boldsymbol{m}}^{\mathrm{T}} \underline{\boldsymbol{f}}$.

Slide 7 Rigid-body rotation always occurs about an axis, which is a line somewhere in space. $\boldsymbol{\omega}$ specifies the direction of this line, but does not say where it is. Thus, $\boldsymbol{\omega}$ on its own does not provide a complete description of the rotational motion, but the pair $(\boldsymbol{\omega}, P)$ does.

Slide 9 A body-fixed point is a point that maintains a fixed position relative to a rigid body. If the body moves, then the point moves with it.

It can be helpful to imagine the whole of space to be filled with body-fixed points. As the body moves, all of these points move with it, creating a flow of points everywhere in space. In particular, there will be a stream of points passing through O, and $\boldsymbol{v}_O$ gives the velocity of this stream at O.

The important step that is happening on slides 8 and 9 is the dissociation of the measurement point from the motion of the body. On slide 6, P is a body-fixed point and $\boldsymbol{v}_P$ is the velocity *of* P. By slide 9, we have established O as a fixed point in space, and defined $\boldsymbol{v}_O$ to be the velocity *measured at* O.

*Dept. Information Engineering, RSISE, ANU, Canberra ACT 0200, Australia. © 2008 Roy Featherstone

Slide 13 Ox is the x axis of the coordinate frame. $\boldsymbol{d}_{Ox}$ is therefore a unit rotation about the x axis, and similarly for Oy and Oz.

Slide 14 Spatial vectors are normally written using standard vector notation. Thus, the symbol for spatial velocity would normally be $\boldsymbol{v}$. However, this practice results in a few name clashes between spatial vectors and other kinds of vector. To resolve this problem, we add a hat to the symbol denoting the spatial vector wherever there is a risk of confusion. On this slide, the symbol $\hat{\boldsymbol{v}}$ is used to denote spatial velocity in order to avoid possible confusion with the symbols $\boldsymbol{v}_O$ and $\boldsymbol{v}_P$, which denote 3D linear velocities; but no hats appear on the symbols $\boldsymbol{d}_{Ox}$, $\boldsymbol{d}_{Oy}$, etc., because there is no need. Hats can also be used to emphasise that a particular quantity is spatial.

Slide 15 The expression $\hat{\boldsymbol{v}} = \omega_x \boldsymbol{d}_{Ox} + \cdots + v_{Oz} \boldsymbol{d}_z$ on slide 14 is invariant in exactly the same sense that the expression $\boldsymbol{\omega} = \omega_x \boldsymbol{i} + \omega_y \boldsymbol{j} + \omega \boldsymbol{k}$ on slide 11 is invariant. If we choose a different coordinate frame then the basis vectors will be different; but the coordinates will also be different, and these differences cancel out in the sum.

Slide 17 The notation $\begin{bmatrix} \boldsymbol{\omega} \\ \hline \boldsymbol{v}_O \end{bmatrix}$ appearing on this slide is very convenient, and in widespread use. However, it is important to remember that this notation is simply a short-hand way of listing the six Plücker coordinates of a spatial vector. A spatial vector is *not* an ordered pair of 3D vectors.

Slide 19 In the classical treatment of rigid-body dynamics, $\boldsymbol{f}$ and $\boldsymbol{n}_P$ would be described as the resultant and the moment about P of a system of applied forces.

Slides 22 & 23 Moment or couple 'in the x direction' means that the vector representing the moment or couple points in the x direction.

Slide 27 Using the numeric notation on slide 5, we have $\boldsymbol{d}_{Ox} = \boldsymbol{d}_1$, $\boldsymbol{e}_x = \boldsymbol{e}_1$, $\boldsymbol{d}_{Oy} = \boldsymbol{d}_2$, and so on. This particular ordering of the basis vectors causes the three angular Plücker coordinates to be listed ahead of the three linear coordinates. It is possible to choose a different ordering of the basis vectors, which implies a different ordering of the Plücker coordinates. In particular, some authors prefer to have the linear coordinates appear ahead of the angular ones.

Slide 28 $\tilde{\boldsymbol{r}}$ is the matrix that implements the vector cross product for 3D vectors: $\tilde{\boldsymbol{r}}\,\boldsymbol{p} = \boldsymbol{r} \times \boldsymbol{p}$ for any $\boldsymbol{p}$. For this reason, $\tilde{\boldsymbol{r}}$ may also be written $\boldsymbol{r}\times$. Note that $\tilde{\boldsymbol{r}}^{\mathrm{T}} = -\tilde{\boldsymbol{r}}$.

$\boldsymbol{E}$ is the coordinate transformation matrix from A to B coordinates for 3D vectors. Thus, $\boldsymbol{E}$ is a 3×3 rotation matrix. To be more explicit, we could have written $\boldsymbol{E}$ as ${}^B\boldsymbol{E}_A$.

The final equation on this slide follows from the requirement that the scalar product be invariant with respect to a change of basis. If $\boldsymbol{m}_A$, $\boldsymbol{m}_B$, $\boldsymbol{f}_A$ and $\boldsymbol{f}_B$ represent $\boldsymbol{m} \in \mathsf{M}^6$ and $\boldsymbol{f} \in \mathsf{F}^6$ in A and B coordinates, then we require $\boldsymbol{m}_A^{\mathrm{T}}\,\boldsymbol{f}_A = \boldsymbol{m}_B^{\mathrm{T}}\,\boldsymbol{f}_B$ for all $\boldsymbol{m}$ and $\boldsymbol{f}$. This implies that $({}^B\boldsymbol{X}_A)^{\mathrm{T}}\,{}^B\boldsymbol{X}_A^*$ must be the identity matrix.

Slide 33 In analogy with $\tilde{\boldsymbol{r}}$ $(= \boldsymbol{r}\times)$ on slide 28, we can define a pair of 6×6 matrices, $\hat{\boldsymbol{v}}\times$ and $\hat{\boldsymbol{v}}\times^*$, that implement the two spatial cross-product operations as follows:

$$\hat{\boldsymbol{v}}\times = \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix}\times = \begin{bmatrix} \tilde{\boldsymbol{\omega}} & \boldsymbol{0} \\ \tilde{\boldsymbol{v}} & \tilde{\boldsymbol{\omega}} \end{bmatrix} \qquad \text{and} \qquad \hat{\boldsymbol{v}}\times^* = \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix}\times^* = \begin{bmatrix} \tilde{\boldsymbol{\omega}} & \tilde{\boldsymbol{v}} \\ \boldsymbol{0} & \tilde{\boldsymbol{\omega}} \end{bmatrix} = -(\hat{\boldsymbol{v}}\times)^{\mathrm{T}}\,.$$

Note that $\hat{\boldsymbol{v}}\times$ maps motion vectors to motion vectors, while $\hat{\boldsymbol{v}}\times^*$ maps force vectors to force vectors.

Slide 34 The derivative of a motion vector is a motion vector, and the derivative of a force vector is a force vector.

The general formula on this slide applies to differentiation with respect to any scalar, not only time.

The two operators $\boldsymbol{v}\times$ and $\boldsymbol{v}\times^*$ act like differentiation operators on motion and force vectors, respectively, in exactly the same way that $\boldsymbol{\omega}\times$ acts like a differentiation operator on 3D vectors: If a 3D vector $\boldsymbol{u}$ is rotating with angular velocity $\boldsymbol{\omega}$, but not otherwise changing, then its time-derivative is $\dot{\boldsymbol{u}} = \boldsymbol{\omega} \times \boldsymbol{u}$.

Slide 35 We originally defined O to be a point fixed in space. We now relax that definition to allow O to have a nonzero velocity.

The formula on this slide is very closely analogous to the standard formula for differentiating a 3D Euclidean vector in a rotating coordinate system. Expressed in the same notation as used on the slide, the 3D vector formula is

$$\left[\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}\right]_O = \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}_O + \boldsymbol{\omega}_O \times \boldsymbol{u}_O \,,$$

where $\boldsymbol{u}$ is a 3D Euclidean vector, $\boldsymbol{u}_O$ is the coordinate vector representing $\boldsymbol{u}$ in the rotating coordinate system, and $\boldsymbol{\omega}_O$ is the coordinate vector representing the angular velocity of the coordinate frame expressed in the rotating coordinate system.

Note that a Plücker coordinate system is moving if the coordinate frame has a nonzero spatial velocity, whereas a Cartesian coordinate system is moving only if the coordinate frame has a nonzero angular velocity. If the frame has a purely translational velocity then it defines a moving Plücker coordinate system but a stationary Cartesian one.

Slide 39 The apparent discrepancy between $\boldsymbol{v}_O = \dot{\boldsymbol{r}}$ and $\dot{\boldsymbol{v}}_O \neq \ddot{\boldsymbol{r}}$ is simply because $\boldsymbol{v}_O$ is tied to O, whereas $\dot{\boldsymbol{r}}$ is tied to the body-fixed point that coincides with O at the current instant. If these two points coincide at time t, then, in general, they will not coincide at time $t + \delta t$. Thus, we have $\boldsymbol{v}_O = \dot{\boldsymbol{r}}$ at time t, but $\boldsymbol{v}_O \neq \dot{\boldsymbol{r}}$ at time $t + \delta t$.

Referring back to the example on slide 38, suppose we define $\boldsymbol{r}$ to be the position of a body-fixed point relative to a point on the fixed rotation axis. It then follows that the velocity of this body-fixed point is $\dot{\boldsymbol{r}} = \boldsymbol{\omega} \times \boldsymbol{r}$, and its acceleration is $\ddot{\boldsymbol{r}} = \boldsymbol{\omega} \times \boldsymbol{\omega} \times \boldsymbol{r}$ (given that $\dot{\boldsymbol{\omega}} = \mathbf{0}$). One can now easily verify that the spatial acceleration of this body is indeed zero.

Slide 40 Yes, there really is no Coriolis term in the formula for summation of accelerations. Nevertheless, the equation of motion correctly accounts for all dynamic effects, including Coriolis forces.

Slide 42 Spatial inertia matrices are symmetric and positive-definite. (This is true in any dual coordinate system, not just Plücker coordinates.) $\boldsymbol{I}_O$ is the rotational inertia of the body about O, and the equation $\boldsymbol{I}_O = \boldsymbol{I}_C - m\tilde{\boldsymbol{c}}\tilde{\boldsymbol{c}}$ is simply the parallel-axis formula expressed in vector notation.

Slide 43 Observe that the coordinate transformation formula is a congruence transform. Such a transform preserves symmetry and positive definiteness, but not eigenvalues. Spatial inertias do not have meaningful eigenvalues.

Slide 44 Coriolis and centrifugal forces are contained within the term $\boldsymbol{v} \times^* \boldsymbol{I}\boldsymbol{v}$.

The kinetic energy of a rigid body with inertia $\boldsymbol{I}$ and velocity $\boldsymbol{v}$ is $\frac{1}{2}\boldsymbol{v}^{\mathrm{T}}\boldsymbol{I}\boldsymbol{v}$.

Slides 45–49 Velocity constraints are typically obtained from position constraints. Suppose the position of a rigid body is given in terms of a set of six generalized coordinates, $q_1, \ldots, q_6$. A single constraint on its position can be given by the constraint equation $\phi(q_1, \ldots, q_6) = 0$, and a set of constraints by a set of equations, $\phi_i(q_1, \ldots, q_6) = 0$ for $i = 1 \ldots n_c$, where n_c is the number of constraints. Let $\boldsymbol{J}$ be the $n_c \times 6$ matrix of partial derivatives, such that $J_{ij} = \partial\phi_i/\partial q_j$, then the motion subspace, S, is the null space of $\boldsymbol{J}$, and $\boldsymbol{S}$ is any $6 \times (6 - n_c)$ full-rank matrix satisfying $\boldsymbol{J}\boldsymbol{S} = \boldsymbol{0}$. The constraint force subspace, T, is the range space of $\boldsymbol{J}^{\mathrm{T}}$, so it is possible to choose $\boldsymbol{T} = \boldsymbol{J}^{\mathrm{T}}$.

Slide 55 This is not quite the complete algorithm. A complete version is as follows:

$$
\begin{aligned}
&{}^{i}\boldsymbol{X}_{i-1} = \boldsymbol{X}_{\mathrm{J}}(q_i)\,\boldsymbol{X}_{\mathrm{L}i} && ({}^{n+1}\boldsymbol{X}_n = \boldsymbol{X}_{ee}) \\
&\boldsymbol{v}_i = {}^{i}\boldsymbol{X}_{i-1}\,\boldsymbol{v}_{i-1} + \boldsymbol{s}_i\,\dot{q}_i && (\boldsymbol{v}_0 = \boldsymbol{0}) \\
&\boldsymbol{a}_i = {}^{i}\boldsymbol{X}_{i-1}\,\boldsymbol{a}_{i-1} + \boldsymbol{s}_i\,\ddot{q}_i + \dot{\boldsymbol{s}}_i\,\dot{q}_i && (\boldsymbol{a}_0 = -\boldsymbol{a}_g) \\
&\boldsymbol{f}_i = ({}^{i+1}\boldsymbol{X}_i)^{\mathrm{T}}\boldsymbol{f}_{i+1} + \boldsymbol{I}_i\,\boldsymbol{a}_i + \boldsymbol{v}_i \times^{*} \boldsymbol{I}_i\,\boldsymbol{v}_i && (\boldsymbol{f}_{n+1} = \boldsymbol{f}_{ee}) \\
&\tau_i = \boldsymbol{s}_i^{\mathrm{T}}\boldsymbol{f}_i
\end{aligned}
$$

This algorithm associates two coordinate frames with each joint i: one embedded in link $i{-}1$ and the other in link i. The latter is defined to be the link coordinate system for link i, and the algorithm performs all calculations pertaining to link i in link i coordinates. The first line computes the coordinate transform matrix from link $i-1$ to link i coordinates. This matrix is the product of a constant matrix, $\boldsymbol{X}_{\mathrm{L}i}$, and a variable matrix, $\boldsymbol{X}_{\mathrm{J}}(q_i)$. The former depends on the relative locations of the two coordinate frames that are embedded in link $i-1$, and the latter is a function of the joint variable, q_i, and also the joint type (e.g. a code identifying whether the joint is revolute, prismatic or something else), which is not shown. $\boldsymbol{X}_{ee}$ is the transform from link n coordinates to the end-effector (or tool tip, gripper, etc.) coordinate frame; and $\boldsymbol{f}_{ee}$ is the force exerted by the robot onto whatever it is touching, holding in its gripper, etc.

$\boldsymbol{a}_g$ is the acceleration due to gravity. By setting the base's acceleration to $-\boldsymbol{a}_g$, rather than zero, we can simulate the effect of gravity on the robot. This is simpler and more efficient that explicitly calculating the gravitational force on each link. However, it does mean that $\boldsymbol{a}_i$ is not the true acceleration of link i.

The quantities $\boldsymbol{X}_{\mathrm{L}i}$ and $\boldsymbol{I}_i$ are constants in link coordinates, and their values would be stored in a data structure describing the robot mechanism. Depending on joint type, $\boldsymbol{s}_i$ may or may not be constant in link coordinates. If it is a constant (the usual case) then we have $\dot{\boldsymbol{s}}_i = \boldsymbol{v}_i \times \boldsymbol{s}_i$. If it is not constant, then both $\boldsymbol{s}_i$ and $\dot{\boldsymbol{s}}_i$ need to be calculated as functions of the joint's position and velocity variables.

Strictly speaking, this algorithm works in *stationary link coordinates*, which are stationary coordinate systems that coincide at the current instant with the moving coordinate systems that are embedded in the links.