Rotation Matrices

Rotation matrices are essential for understanding how to convert from one reference system to another. Converting from one reference system to another is essential for computing joint angles, a key task in the analysis of human movement. First we will discuss rotations in 2-dimensional space (i.e. rotations in the plane). Then we will discuss rotations in 3 dimensions.

Two-dimensional rotation matrices
Consider the 2x2 matrices corresponding to rotations of the plane. Call $R_v(\theta)$ the 2x2 matrix corresponding to rotation of all vectors by angle $+\theta$.

Since a rotation doesn’t change the size of a unit square or flip its orientation, $\det(R_v) = 1$.

Consideration of what happens to the unit column vectors $(1;0)$ and $(0;1)$ when rotated CCW by angle $\theta$, shows that $(1;0)$ gets sent to $(\cos \theta;\sin \theta)$, and $(0;1)$ gets sent to $(-\sin \theta;\cos \theta)$. Therefore $R_v(\theta) = [\cos \theta, -\sin \theta; \sin \theta, \cos \theta]$ (Matlab-style representation of $R$).

Show $\det(R_v) = 1$.

Show that 2 ways of figuring out what the inverse of $R_v$ is give same answer:
First way: $R^{-1}_v = \ldots$, determined using formula for inverse of any 2x2 matrix.
Second way: $R^{-1}_v = \ldots$, determined by considering what you have to do to “go back to where you started”. In case of an initial rotation by $\theta$, the way to go back is to rotate by $-\theta$. So plug in $(-\theta)$ for $\theta$ in the original $R_v$ matrix to find $R^{-1}_v$.

Multiple rotations: To rotate twice, just multiply two rotation matrices together. The “angle sum” formulae for sin, cos can be derived this way. We know from thinking about it that when doing rotations of the plane, it doesn’t matter whether you first rotate by 30, then by 60, or if you rotate by 60, then 30: You end up in the same place either way, rotated 90 from where you started. The fact that the order doesn’t matter means that, if our 2D rotation matrices act the way real 2D rotations act, they should commute: in other words, for example, $R_v(30)*R_v(60)$ should equal $R_v(60)*R_v(30)$ (and both should equal $R_v(90)$). It can be shown that the 2-dimensional (i.e. 2x2) rotation matrices DO commute, as expected. (The 3D rotation matrices do not commute; more on this below.)

Rotations of vectors vs. rotation of axes
For more information see Young-Hoo Kwon’s web site:

Rotations, as described above, are vector rotations: transformations that take vectors and “move them” to new positions, by rotating them. A closely related, but subtly different, concept is to change the way we describe a vector, but not change the vector itself. This is called (in math books) a change of basis; it is also called a change of reference system or a change of coordinates. It means that instead of expressing a vector in terms of its components along the unit vectors $i=(100)$, $j=(010)$, and $k=(001)$, we express the same vector in terms of its components (or projections) along a new set of basis vectors $i'$, $j'$, and $k'$. 
(The term basis, and basis vectors, refers to a set of vectors that can be linearly combined, using scalar multiplication and vector addition, to point to any vector in the vector space. It takes 2 basis vectors to “span” (i.e. get to all points in) a 2D space, 3 basis vectors to span 3D, etc. It is convenient, but not required, that basis vectors have length 1. A set of basis vectors to span 2 dimensions need not be orthogonal they just can’t be parallel. A set of basis vector that span 3 dimensions need not be orthogonal they just can’t all lie in the same plane.)

We can use matrix multiplication to compute the components of vectors in the new reference system. To do this, we have to know the coordinates of the new basis vectors in terms of the original reference frame. (We’re dotting the vector with the new unit vectors to find its components in the new frame of reference. Matrix multiplication is just a convenient way of computing the dot products.)

An important special case of changing the reference frame is a rotation (and translation) of the reference frame. Common in biomechanics. In this case, the new basis vectors are just rotated versions of the old ones. So if the old ones had unit length and were orthogonal, the new ones will too.

2D case first: the new reference frame is rotated by angle θ relative to the original frame. Then the components of the old unit vectors \( \begin{bmatrix} 1 \\ 0 \end{bmatrix} \) and \( \begin{bmatrix} 0 \\ 1 \end{bmatrix} \), expressed in the new frame of reference, are given by \( \begin{bmatrix} \cos \theta \\ -\sin \theta \end{bmatrix} \) and \( \begin{bmatrix} \sin \theta \\ \cos \theta \end{bmatrix} \). The components of a vector \( \begin{bmatrix} x \\ y \end{bmatrix} \) in the new reference frame are \( \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \)

Therefore the matrix associated with a rotation of the coordinate axes by \( \theta \) is
\[
R_c(\theta) = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}
\]

Note that \( R_c(\theta) \) looks like \( R_v(-\theta) \). This is due to the fact that rotation of the coordinate frame by \( \theta \) is like rotation of all the vectors by \(-\theta\). This is true in 3D as well: a rotation of the axes is equivalent to a rotation of the vectors in the opposite direction.

Using matrices to convert from one reference system to another.
This section is similar to the preceding section but it uses the notation and approach of Winter. (See Winter, 3rd ed., sections 6.2.6, 6.2.7, 7.0, 7.1, and Winter, 4th ed., sections 7.0, 7.1, 8.2.6, 8.2.7. A warning about notation: Winter uses \( \mathbf{R} \) and \( \mathbf{r} \) to denote position vectors, which differs from our usual habit of using upper case for matrices and lower case for vectors. Some authors use \( \mathbf{R} \) for rotation matrices, but Winter uses \( \mathbf{A} \) in this section of his book. This section of the notes follows Winter’s notation.)

Suppose a point \( \mathbf{a} \) has coordinates \( \mathbf{R}_a = \begin{bmatrix} X_a \\ Y_a \end{bmatrix} \) in the global reference system (the GRS).

Suppose there is a local reference system 1 (LRS1) with the same origin as the GRS but rotated with respect to it, by an angle \( \theta \). The coordinates of point \( \mathbf{a} \) in LRS1 are \( \mathbf{r}_{1a} = \begin{bmatrix} x_{1a} \\ y_{1a} \end{bmatrix} \)
(There is just one Global Reference System for a data record, but there may be many local reference systems for the same data - for example, the LRS of the pelvis, the LRS of the hip, etc. We’ll use uppercase for data expressed in terms of the Global Reference System, and lower case for data expressed in terms of a local reference system.)
How are $\mathbf{R}_a$ and $\mathbf{r}_{1a}$ related? One can prove the following:

$$X_a = x_{1a} \cos \theta - y_{1a} \sin \theta$$
$$Y_a = x_{1a} \sin \theta + y_{1a} \cos \theta$$

(local to global) \hspace{1cm} (1.1)

This can be rewritten with matrices:

$$\begin{pmatrix} X_a \\ Y_a \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x_{1a} \\ y_{1a} \end{pmatrix}$$

(1.2)

or

$$\mathbf{R}_a = \mathbf{A} \mathbf{r}_{1a}$$

(local to global) \hspace{1cm} (1.3)

where $\mathbf{A}$ is the rotation-of-coordinates matrix

$$\mathbf{A} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} = \begin{pmatrix} c & -s \\ s & c \end{pmatrix}$$

(c=$\cos \theta$, s=$\sin \theta$). The first column of $\mathbf{A}$ is the global coords that $\mathbf{i}$ would have after it (the vector $\mathbf{i}$, not the coordinate system) is rotated by angle $+\theta$. The second column of $\mathbf{A}$ is the global coordinates that $\mathbf{j}$ would have after being rotated by angle $+\theta$.

There are two different interpretations of $\mathbf{A}$, both valid.

1. $\mathbf{A}$ re-expresses any vector in terms of a new coordinate system (i.e. in terms of a new basis).
   Original basis = LRS, new basis=GRS. In other words $\mathbf{Ar} = \mathbf{R}$, where $\mathbf{r}$ and $\mathbf{R}$ are the same point, expressed in LRS and GRS respectively. The LRS is rotated by $+\theta$ with respect to the GRS. $\mathbf{A}$ doesn’t move the vector.

2. $\mathbf{A}$ moves a vector by angle $+\theta$. $\mathbf{Ar}_1 = \mathbf{r}_2$, where $\mathbf{r}_1$ and $\mathbf{r}_2$ are the coordinates of the vector before and after rotation by $+\theta$. There is no change of coordinate system.

The matrix $\mathbf{A}$ is the same as the matrix $\mathbf{R}_c(\theta)$ above. It is the transpose, or inverse (they’re the same for rotation matrices), of the matrix $\mathbf{R}_c(\theta)$. Why aren’t $\mathbf{A}$ and $\mathbf{R}_c(\theta)$ the same, if they both describe rotations of coordinate systems? Because $\mathbf{R}_c(\theta)$ goes from GRS to LRS and $\mathbf{A}$ goes from LRS to GRS.

Equations (1.1) and (1.2) illustrate the important idea that a matrix can be thought of as the “instructions” for a linear transformation of a vector to another vector. A 2x2 matrix represents a
transformation that maps the set of all 2d vectors, i.e. all points in the x-y plane, into a new set of 2d vectors (or, equivalently, a new set of points). A 3x3 matrix maps 3d vectors into 3d vectors. In this case, the transformation represented by the matrix in equation 1.2 is a rotation, but other values for the elements of A would give other transformations.

Now consider a second local reference system, LRS2. It is rotated like LRS1, but, unlike LRS1, LRS2 is also offset, i.e. the origins of the GRS and LRS2 are not the same. In other words, the coordinate systems are related by a translation as well as a rotation.

If the coordinates of origin of LRS2 are \( \mathbf{R}_2 = (X_2, Y_2) \) in the GRS, then \( \mathbf{R}_a \) and \( \mathbf{r}_{2a} \) are related as follows:

\[
\begin{pmatrix} X_a \\ Y_a \end{pmatrix} = \begin{pmatrix} X_2 \\ Y_2 \end{pmatrix} + \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x_a \\ y_a \end{pmatrix}
\]

(some subscripts in above eqn missing - due to bug in how MS Word saves eqn as HTML?) (1.4)

or

\[
\mathbf{R}_a = \mathbf{R}_2 + \mathbf{A} \mathbf{r}_{2a}
\] (1.5)

As before, the columns of \( \mathbf{A} \) are the coordinates of \( \mathbf{i} \) and \( \mathbf{j} \) after rotation by angle +\( \theta \). As before, \( \mathbf{A} \) may be thought of as

1. The matrix that changes a vector’s representation (when \( \mathbf{A} \) multiplies that column vector) from LRS coordinates to GRS coordinates, where the LRS is rotated by +\( \theta \) relative to the GRS.
2. The matrix that rotates a vector by +\( \theta \), when it multiplies that column vector. It does not change coordinate systems.

**Rotation-of-reference-frame matrices in 3D**

Tom Kepple’s HESC690 notes (2010) use a different convention than Jim Richards’s notes. Tom K takes the unit vectors of the body’s coordinate system (LCS) and makes them the rows of a rotation matrix. Jim Richards (according to TK) takes the unit vectors of the body’s coordinate system (LCS) and makes them the columns of a rotation matrix. The powerpoints I have from TK do not display equations properly on my computer so I don’t know for sure that this is all true. I think TK’s rotation matrix, when multiplied by a following column vector, will map from local to global. JR’s matrix (as described by TK) will, I think, when multiplied by a following column vector, map from global to local.)

Describing rotations in 3 dimensions.
First start with the 2D rotations, in 3D:

\[
R_z(\psi) = \begin{bmatrix}
c(\psi) & s(\psi) & 0 \\
-s(\psi) & c(\psi) & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

= rotation of coordinate frame by \( \psi \) about the z-axis

\[
R_y(\theta) = \begin{bmatrix}
c(\theta) & 0 & -s(\theta) \\
0 & 1 & 0 \\
s(\theta) & 0 & c(\theta)
\end{bmatrix}
\]

= rotation of coordinate frame by \( \theta \) about the y-axis

\[
R_x(\phi) = \begin{bmatrix}
1 & 0 & 0 \\
0 & c(\phi) & s(\phi) \\
0 & -s(\phi) & c(\phi)
\end{bmatrix}
\]

= rotation of coordinate frame by \( \phi \) about the x-axis

If we rotate about x, then about the (new) y’, then about the (new new) z”, we get

\[
T = R_z(\psi) R_y(\theta) R_x(\phi) \begin{bmatrix}
c(\theta)c(\psi) & s(\phi)s(\theta)c(\psi)+c(\phi)s(\psi) & -c(\phi)s(\theta)c(\psi)+s(\phi)s(\psi) \\
-c(\theta)s(\psi) & -s(\phi)s(\theta)s(\psi)+c(\phi)c(\psi) & c(\phi)s(\theta)s(\psi)+s(\phi)c(\psi) \\
s(\theta) & -s(\phi)c(\theta) & c(\phi)c(\theta)
\end{bmatrix}
\]

(Equation above agrees with Winter, 3rd ed. (2005), Eq. 7.15, p. 183.)

Multiplying this matrix times a vector gives a new vector, which is the old vector expressed in terms of the new rotated reference frame. The same rotations in a different order will give a different result. In other words, these rotation matrices do not commute. Order matters.


Going backwards

How do we determine and express the relative orientation of two body segments? Typically the proximal segment axes are considered the reference, and it is desired to know the rotation of the distal segment relative to the proximal segment. Example: orientation of thigh relative to pelvis. (Note that we are just interested in rotations, not translations, for now.) Anatomical axes of proximal segment are determined from markers and we call the axes \( \text{i}=(i_x, i_y, i_z), \text{j}, \text{ and k} \) (orthogonal unit direction vectors expressed in terms of the global, or laboratory, reference system, the GRS). We can define a 3x3 matrix

\[
R_{prox} = (i \ j \ k) = \begin{bmatrix}
i_x & j_x & k_x \\
i_y & j_y & k_y \\
i_z & j_z & k_z
\end{bmatrix}
\]

whose columns are the vectors \( \text{i}, \text{j}, \text{ and k} \), expressed in the GRS. The anatomical axes of the distal segment are also computed from markers and we’ll call them \( \text{i'}=(i'_{x'}, i'_{y'}, i'_{z'}), \text{j'}, \text{ and k'} \), expressed in the GRS. We define a 3x3 matrix

\[
R_{dist} = (i' \ j' \ k') = \begin{bmatrix}
i'_{x'} & j'_{x'} & k'_{x'} \\
i'_{y'} & j'_{y'} & k'_{y'} \\
i'_{z'} & j'_{z'} & k'_{z'}
\end{bmatrix}
\]
whose columns are the vectors $i'$, $j'$, and $k'$. We next find the expression for $i'$, $j'$, $k'$ in terms of vectors $i$, $j$, $k$:

\[ i' = (i' \cdot i)i + (i' \cdot j)j + (i' \cdot k)k \]
\[ j' = (j' \cdot i)i + (j' \cdot j)j + (j' \cdot k)k \]
\[ k' = (k' \cdot i)i + (k' \cdot j)j + (k' \cdot k)k \]

...  

This means that $i' \cdot i$, $i' \cdot j$, and $i' \cdot k$ are the components of $i'$ in the directions $i$, $j$, and $k$, and so on. This matrix is also the rotation-of-coordinates matrix $T_c$ derived above, since one definition of $T_c$ is that $T_c$ times a vector (such as $i'$, $j'$, or $k'$) gives the projection of that vector onto the axes of the reference frame defined by $i$, $j$, and $k$. We know $i$, $j$, and $k$, and $i'$, $j'$, and $k'$, so we can compute the elements of $T_c$:

\[
T_c = \begin{pmatrix}
t_{11} & t_{12} & t_{13} \\
t_{21} & t_{22} & t_{23} \\
t_{31} & t_{32} & t_{33}
\end{pmatrix}
\]

where $t_{11} = i' \cdot i$, $t_{12} = i' \cdot j$, $t_{13} = i' \cdot k$, $t_{21} = j' \cdot i$, etc. This can be expressed as a matrix equation:

\[
T_c = R_{dist}^T R_{prox}
\]

This matrix is also called the direction cosines matrix (Winter 2005 p. 169), since the dot product of two unit vectors equals the cosine of the angle between them: $i' \cdot i = \cos a$, where $a$ = angle between $i'$ and $i$, and so on.

There is another way to think about $R_{prox}$, $R_{dist}$, and the transformation between them. $R_{prox}$ and $R_{dist}$ are rotation matrices. They describe $i, j, k$ and $i', j', k'$ in the GRS, and they are also the rotation matrices that describe how to move the $ijk$ of the GRS to the $ijk$ and $i'j'k'$ of the two LRSs. Since $R_{prox}$ and $R_{dist}$ are sets of orthogonal unit vectors, there is a rotation $T_v$ that move the vectors of $R_{prox}$ into the vectors of $R_{dist}$. We define

\[
T_v R_{prox} = R_{dist}
\]

Therefore

\[
T_v R_{prox} R_{prox}^{-1} = R_{dist} R_{prox}^{-1}
\]

\[
T_v = R_{dist} R_{prox}^{-1}
\]

But for rotation matrices, inverse = transpose, so

\[
T_v = R_{dist}^T R_{prox}
\]

Now we (finally) have $T_v$. Note that it is easy to show, using the properties of matrices and the equations above, that $T_c$ and $T_v$ are transposes of one another. Since they are rotation matrices, that also means they are inverses of one another. Matrix $T_v$ can be calculated from marker data at each time point, and is independent of which Cardan or Euler sequence we prefer. (More on that in a moment.) Next, we want to know the joint rotation angles, i.e. how much flexion/extension, ab/adduction, and/or internal/external rotation is necessary to move the distal segment from the “reference” position to the observed position, which is the position specified by the distal segment axes $i'$, $j'$, $k'$. We mentioned above that the 3D rotation matrices do not commute: that means a sequence of rotations about certain axes will get you to a different final orientation if you do them in a different order. Therefore, to find the joint angles, we have to specify the order of rotation, for example: flex/ext followed by ab/ad followed by int/ext rotate, or another sequence. Furthermore, we could do rotations about 3 different axes (x-y-z, z-x-y, etc.), or we rotate about axis1, then axis 2, then axis1 again (x-y-x, y-z-y, etc.). It is worth noting that the first axis of rotation has moved to a new orientation, due the second rotation, by the time we rotate about it again during the third and final rotation. This is important, because if it hadn’t
moved, then the first and third rotations would be equivalent to a single rotation, and we would only have 2 degrees of freedom in our overall sequence, not the 3 we need to achieve any possible final orientation. Sequences that fit the x-y-z pattern are called Cardan sequences; x-y-x type sequences are called Euler sequences. There are 6 of each. The choice of sequence depends on practical and anatomical considerations. The International Shoulder Group has recommendations for the shoulder (Wu et al., J Biomech, 2005), but they are only useful for certain types of shoulder movements. (See powerpoint presentation by James Richards for animations of this, and for comparison of the different sequences, and which ones work well for which types of movements.) Different sequences will give different angles due to the non-commutativity of the 3D rotation matrices. If we know the “measured” rotation matrix $T_v$ from analysis of 3D video, we set it equal to the “analytical” rotation matrix $T_v$ which corresponds to the desired rotation sequence. An example of an analytic rotation matrix is the one in the previous section for the sequence x-y-z, in which there is a rotation by angle $\phi$ about the x axis, then by angle $\theta$ about the y axis, and finally by $\psi$ about the z axis. Then we choose values of $\phi$, $\theta$, and $\psi$ to make the corresponding elements in the analytic matrix equal the elements in the measured matrix. The analytic matrices for x-z-x, z-y-x, etc., will be different than the x-y-z matrix given above.

“Gimbal lock”: If $\theta = \pi/2$, then $\cos \theta=0$ and $\sin \theta=1$, and the $T$ matrix above simplifies to

$$
T = R_x(\psi) R_y(\pi/2) R_z(\phi) = \begin{bmatrix} 0 & s(\phi+\psi) & -c(\phi+\psi) \\ 0 & c(\phi+\psi) & s(\phi+\psi) \\ 1 & 0 & 0 \end{bmatrix}.
$$

Simpler is not necessarily better though: when we experimentally determine the elements $t_{ij}$ of the matrix $T$ in this situation, we cannot “go backwards” to figure out $\phi$, $\theta$, and $\psi$ because the elements of $T$ predicted by the theory of rotations, when $\theta=\pi/2$, as shown above, don’t depend on $\phi$ and $\psi$ individually; they only depend on the sum of $\phi$ and $\psi$. So all we can say about the rotations that must happened to get us to the experimentally measured situation is that the sum of $\phi$ and $\psi$ equals a certain value (the arcsin of $t_{12}$). This situation is called gimbal lock. You can see it with a tinkertoy model of rotation about 3 axes: bicycle pedal effect. In the model, you can rotate about x and z simultaneously (i.e. change $\phi$ and $\psi$ simultaneously, keeping their sum constant) without altering the orientation of the final coordinate frame. So the rotation model of the change in axes has no unique solution in this case. It is reassuring that the physical and mathematical analyses are telling us the same thing.