KAAP686 Mathematics and Signal Processing for Biomechanics

Rotation Matrices

William C. Rose 20150928, 20170423

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, $\det(R_v)$ must = 1.
Consider what happens to the unit vectors $\begin{pmatrix}1\\0\end{pmatrix}$ and $\begin{pmatrix}0\\1\end{pmatrix}$ when rotated CCW by angle θ: $\begin{pmatrix}1\\0\end{pmatrix}$ gets sent to $\begin{pmatrix}\cos\theta\\ \sin\theta\end{pmatrix}$, and $\begin{pmatrix}0\\1\end{pmatrix}$ gets sent to $\begin{pmatrix}-\sin\theta\\ \cos\theta\end{pmatrix}$.

Therefore a vector $v_1 = \begin{pmatrix}v_{1x}\\ v_{1y}\end{pmatrix}$ gets sent to a new vector, $v_2 = \begin{pmatrix}v_{2x}\\ v_{2y}\end{pmatrix}$,
where $\begin{pmatrix}v_{2x}\\ v_{2y}\end{pmatrix} = \begin{pmatrix}v_1\cos\theta - v_2\sin\theta\\ v_1\cos\theta + v_2\sin\theta\end{pmatrix}$.
We can write this transformation of v_1 to v_2 as a matrix equation:

$$v_2 = R_v(\theta)v_1 \qquad (1.0)$$

where

$$R_v(\theta) = \begin{pmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{pmatrix} \qquad (1.1)$$

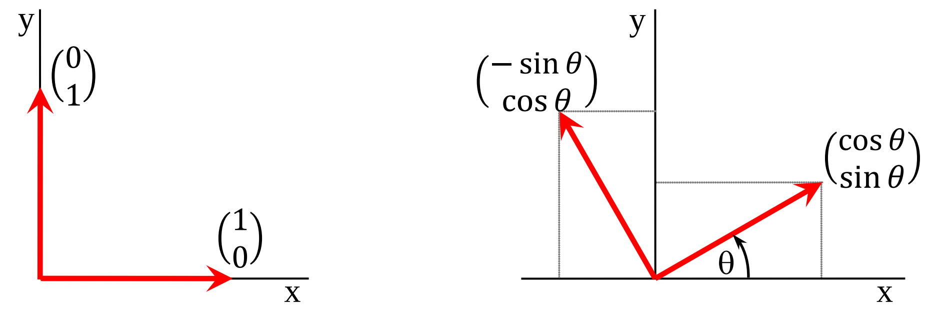
y
$\begin{pmatrix}0\\1\end{pmatrix}$
$\begin{pmatrix}1\\0\end{pmatrix}$
x
y
$\begin{pmatrix}-\sin\theta\\ \cos\theta\end{pmatrix}$
$\begin{pmatrix}\cos\theta\\ \sin\theta\end{pmatrix}$
θ
x

Fig. 1. Left: Vectors with coordinates $\begin{pmatrix}1\\0\end{pmatrix}$ and $\begin{pmatrix}0\\1\end{pmatrix}$, before rotation. Right: After rotation CCW by angle θ, the vectors have coordinates $\begin{pmatrix}\cos\theta\\ \sin\theta\end{pmatrix}$ and $\begin{pmatrix}-\sin\theta\\ \cos\theta\end{pmatrix}$.

Equations 1.0 and 1.1 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. The transformation represented by matrix $\mathbf{R_v}$ in equation 1.1 is a rotation, but other values for the matrix elements would give other transformations.

Multiple rotations: To rotate twice, just multiply two rotation matrices together. The "angle sum" formulae for sine and cosine 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 = 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.)

Exercise 1: Show $\det(R_v) = 1$.
Exercise 2: Show that two ways of figuring out the inverse of R_v give same answer.

First way: Compute R_v^{-1} using formula for inverse of a 2x2 matrix.

Second way: Compute R_v^{-1} 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 θ in the original R_v matrix to find R_v^{-1}.

Rotations of vectors vs. rotation of axes

> For more information on this topic, see Young-Hoo Kwon's web site: http://kwon3d.com/theory/transform.html . You may also find pages at Wolfram Mathworld useful: http://mathworld.wolfram.com/RotationMatrix.html and http://mathworld.wolfram.com/RotationMatrix.html, for example.

Rotations, as described in the preceding section, 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**, **j**, **k**, 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. This is common in biomechanics. In this case, the new basis vectors are just rotated versions of the original ones. So if the original basis vectors have unit length and are orthogonal, the new ones will too.

Rotation of the reference frame in two dimensions

The new reference frame is rotated by angle θ relative to the original frame. Then the components of the original unit vectors i=$\begin{bmatrix}1\\0\end{bmatrix}$ and j=$\begin{bmatrix}0\\1\end{bmatrix}$, expressed in the new frame of reference, are given by $\begin{bmatrix}\cos\theta\\-\sin\theta\end{bmatrix}$ and $\begin{bmatrix}\sin\theta\\\cos\theta\end{bmatrix}$. A point **p**, located at $\begin{bmatrix}x\\y\end{bmatrix}$ in the original reference frame is located at $\begin{bmatrix}x'\\y'\end{bmatrix}$in new reference frame. The figure below illustrates this situation.

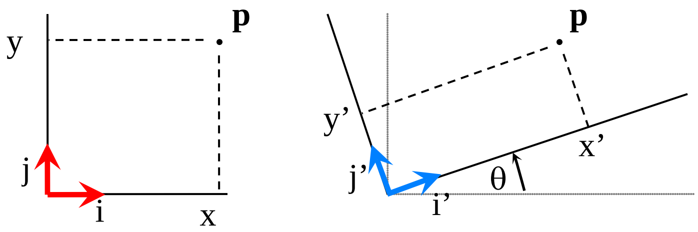

Fig. 2. Left: Point **p** has coordinates (x,y) in the original reference system, whose basis vectors are **i** and **j**. Right: The same point, **p**, has coordinates (x',y') in the rotated coordinate system, whose basis vectors are i' and j'.

The two ways of describing the point's location are related as follows:

$$\begin{bmatrix}x'\\y'\end{bmatrix} = \begin{bmatrix}\cos\theta & \sin\theta\\-\sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix} \qquad (1.2)$$

or

$$\boldsymbol{v}' = \boldsymbol{R_c}(\theta)\boldsymbol{v} \qquad (1.3)$$

where $\mathbf{R}_c(\theta)$ is the matrix associated with a rotation of the coordinate axes by θ, and ***v*** and ***v'*** are vectors describing the location of a point in the original reference frame and in the rotated-by-θ frame, respectively.

$$R_c(\theta) = \begin{bmatrix}\cos\theta & \sin\theta\\-\sin\theta & \cos\theta\end{bmatrix} \qquad (1.4)$$

One way of thinking about the rotation of coordinates matrix is that the columns of $\mathbf{R}_c$ are the projections of the original basis vectors (which are **i**=$\begin{bmatrix}1\\0\end{bmatrix}$ and **j**=$\begin{bmatrix}0\\1\end{bmatrix}$) onto the basis vectors **i'** and **j'** of the rotated frame. Thus column 1 of $\mathbf{R}_c$ is $\begin{bmatrix}i\cdot i'\\i\cdot j'\end{bmatrix}$, and the second column is $\begin{bmatrix}j\cdot i'\\j\cdot j'\end{bmatrix}$. (The inner product, denoted by a dot operator, gives the projection of one vector onto another.) Note that $\mathbf{R}_c(\theta)$ equals $\mathbf{R}_v(-\theta)$ (equation 1.1 above). This is due to the fact that rotation of the coordinate frame by θ is like rotation of all the vectors by -θ.

Rotation-of-reference-frame in 3D

> Tom Kepple's HESC690 notes (2010) use a diferent 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. TK says that Jim Richards takes the unit vectorsof the body's coordinate system (LCS) and makes them the columns of a rotation matrix.

We now consider rotation of the coordinate frame in three dimensions. If column vector **v** is the location of a point, or the orientation of an axis, in the original coordinate system, and if column vector **v'** is the same point's location, or axis's orientation, in the rotated system, then

$$\boldsymbol{v}' = \boldsymbol{R_c v} \tag{1.5}$$

We want to find mathematical formulas for $\mathbf{R_c}$ in terms of the rotation angles about the different axes. We start with the matrices for rotation of coordinate frame about the principal axes. A rotation of coordinate frame by ψ about the z-axis is described by

$$R_z(\psi) = \begin{bmatrix} c(\psi) & s(\psi) & 0 \\ -s(\psi) & c(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{1.6}$$

The positive angular direction is determined by the right-hand rule, in which the thumb points along the positive axis. Figure 1, above, describes this rotation (except now we are calling the angle of rotation ψ, instead of θ). In three dimensions, the z axis, not shown, is coming out of the page.

A rotation of coordinate frame by θ about the y-axis is described by

$$R_y(\theta) = \begin{bmatrix} c(\theta) & 0 & -s(\theta) \\ 0 & 1 & 0 \\ s(\theta) & 0 & c(\theta) \end{bmatrix}. \tag{1.7}$$

Figure 2, below, illustrates this rotation. In the figure, the y axis is coming out of the page toward the reader.

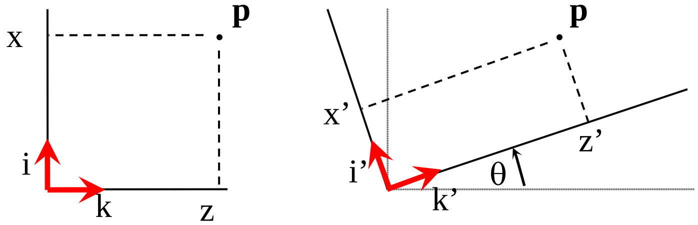

Fig. 2. Left: Point **p** has coordinates (x,y,z) in the original reference system. Basis vectors **i** and **k**.are shown. Basis vector j (which points out of the page) and the y coordinate are not shown. Right: The same point, **p**, has coordinates (x',y',z') in the rotated coordinate system, whose basis vectors are **i'**, **j'**, **k'**. Vector **j'** points out of the page and is not shown. Because the rotation is about the y axis, vector **j'** is the same as vector **j**, and therefore the y coordinate is the same in both coordinate systems: y'=y.

A rotation of coordinate frame by ϕ about the x-axis is described by

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\phi) & s(\phi) \\ 0 & -s(\phi) & c(\phi) \end{bmatrix}. \tag{1.8}$$

Figure 3, below, illustrates this rotation. In the figure, the x axis is coming out of the page toward the reader.

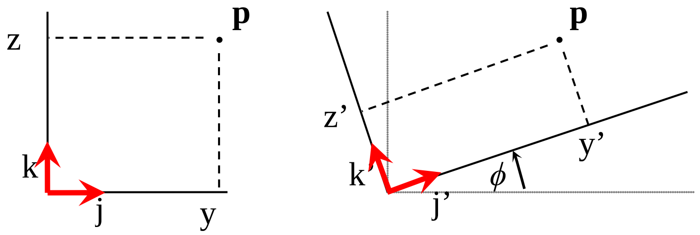

Fig. 3. Left: Point **p** has coordinates (x,y,z) in the original reference system. Basis vectors **j** and **k** are shown. Basis vector **i** (which points out of the page) and the x coordinate are not shown. Right: The same point, **p**, has coordinates (x',y',z') in the rotated coordinate system, whose basis vectors are **i'**, **j'**, **k'**. Vector **i'** points out of the page and is not shown. Because the rotation is about the x axis, vector **i'** is the same as vector **i**, and therefore the x coordinate is the same in both coordinate systems: x'=x.

If we rotate the coordinate system by angle ϕ about x, then by angle θ about the (new) y', then by angle ψ about the (new new) z", we get

$$T = R_z(\psi)R_y(\theta)R_x(\phi) = \begin{bmatrix} c(\theta)c(\psi) & s(\phi)s(\theta)c(\psi)+c(\phi)s(\psi) & -c(\phi)s(\theta)c(\psi)+s(\phi)s(\psi) \\ -c(\theta)s(\psi) & -s(\phi)s(\theta)s(\psi)+c(\phi)c(\psi) & c(\phi)s(\theta)s(\psi)+s(\phi)c(\psi) \\ s(\theta) & -s(\phi)c(\theta) & c(\phi)c(\theta) \end{bmatrix} \tag{1.9}$$

(Equation above agrees with Winter, 3rd ed. (2005), Eq. 7.15, p. 183.)
Multiplying matrix $\mathbf{T_c}$ times a vector gives a new vector, which is the original 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. See Winter, 3rd ed., pp. 169-170, or http://mathworld.wolfram.com/Eulerangles.html. I think Winter's Eq. 6.19 (3rd ed.) and Eq. 8.19 (4th ed.) are wrong.

Going backwards

The preceding section explained how to compute the coordinates of a point in a new reference system, if the rotation angles of the new system, relative to the old, are known. In biomechanics we often want to do the opposite, i.e. we want to go backward. We can measure the coordinates of points after they have been moved from their original position, and from those measurements, we wish to deduce the rotation angles. Typically, the proximal segment axes are considered the original reference frame, and we desire 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.) The anatomical axes of both segments are determined from markers on the segments. We call the proximal segment axes **i**=(i_x,i_y,i_z), **j**, and **k** (orthogonal unit direction vectors expressed in terms of the global, or laboratory, reference system, the GRS). We can define a 3x3 matrix

$$\boldsymbol{R_{prox}} = (\boldsymbol{i} \quad \boldsymbol{j} \quad \boldsymbol{k}) = \begin{pmatrix} i_x & j_x & k_x \\ i_y & j_y & k_y \\ i_z & j_z & k_z \end{pmatrix} \tag{1.10}$$

whose columns are the vectors **i**, **j**, and **k**, expressed in the GRS. The anatomical axes of the distal segment are **i'**=(i_x', i_y', i_z'), **j'**, and **k'** , expressed in the GRS. We define a 3x3 matrix

$$\boldsymbol{R_{dist}} = (\boldsymbol{i'} \quad \boldsymbol{j'} \quad \boldsymbol{k'}) = \begin{pmatrix} i'_x & j'_x & k'_x \\ i'_y & j'_y & k'_y \\ i'_z & j'_z & k'_z \end{pmatrix} \tag{1.11}$$

whose columns are the vectors **i'**, **j'**, and **k'**.

We measure $\mathbf{R_{prox}}$ and $\mathbf{R_{dist}}$ experimentally, by motion capture of the marker locations on the proximal and distal segments. We want to use those results to determine an experimental value for the rotation-of-coordinates matrix $\mathbf{T_c}$. Once we have the experimental estimate of $\mathbf{T_c}$, we compare it to the analytic formula for $\mathbf{T_c}$, which is given by eq. 1.9, for the case of rotation about X, then about rotated Y, then about twice-rotated Z. Finally, we will compute the rotation angles (ϕ, θ, ψ) that will give an analytic matrix that matches the experimentally-measured matrix. These are the joint angles which we are seeking to determine.

We determine the experimental value of $\mathbf{T_c}$ by remembering (by analogy with the 2D case discussed above) that column 1 of $\mathbf{T_c}$ is the projection of **i** along the axes defined by **i'**, **j'**, and **k'**.

$$\boldsymbol{T_{c_{i1}}} = \begin{pmatrix} \mathbf{i'} \cdot \mathbf{i} \\ \mathbf{j'} \cdot \mathbf{i} \\ \mathbf{k'} \cdot \mathbf{i} \end{pmatrix} \tag{1.12a}$$

Column 2 of $\mathbf{T_c}$ is the projection of **j** along **i'**, **j'**, and **k'**:

$$\boldsymbol{T_{c_{i2}}} = \begin{pmatrix} \mathbf{i'} \cdot \mathbf{j} \\ \mathbf{j'} \cdot \mathbf{j} \\ \mathbf{k'} \cdot \mathbf{j} \end{pmatrix} \tag{1.12b}$$

Column 3 of $\mathbf{T_c}$ is the projection of **k** along **i'**, **j'**, and **k'**:

$$\boldsymbol{T_{c_{i3}}} = \begin{pmatrix} \mathbf{i'} \cdot \mathbf{k} \\ \mathbf{j'} \cdot \mathbf{k} \\ \mathbf{k'} \cdot \mathbf{k} \end{pmatrix} \tag{1.12c}$$

We combine the three preceding equations:

$$\boldsymbol{T_c} = \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{pmatrix} = \begin{pmatrix} \mathbf{i'} \cdot \mathbf{i} & \mathbf{i'} \cdot \mathbf{j} & \mathbf{i'} \cdot \mathbf{k} \\ \mathbf{j'} \cdot \mathbf{i} & \mathbf{j'} \cdot \mathbf{k} & \mathbf{j'} \cdot \mathbf{k} \\ \mathbf{k'} \cdot \mathbf{i} & \mathbf{k'} \cdot \mathbf{j} & \mathbf{k'} \cdot \mathbf{k} \end{pmatrix} \tag{1.13}$$

where t_{11}=**i'**·**i**, t_{12}=**i'**·**j**, t_{13}=**i'**·**k**, t_{21}=**j'**·**i**, etc. This matrix is also called the direction cosines matrix, because the inner product of two unit vectors is the cosine of the angle between them: **i'**·**i**=cos *a*, where *a*=angle between **i'** and **i**, and so on.

Matrix $\mathbf{T_c}$ can be written as

$$\boldsymbol{T_c} = \begin{pmatrix} \mathbf{i'}^{\boldsymbol{T}} \\ \mathbf{j'}^{\boldsymbol{T}} \\ \mathbf{k'}^{\boldsymbol{T}} \end{pmatrix} (\boldsymbol{i} \quad \boldsymbol{j} \quad \boldsymbol{k}) \tag{1.14}$$

which can be written as

$$\boldsymbol{T_c} = \boldsymbol{R_{dist}}^{\boldsymbol{T}} \boldsymbol{R_{prox}} \, . \tag{1.15}$$

We measure $\mathbf{R_{dist}}$ and $\mathbf{R_{prox}}$ experimentally during motion capture, during each video frame, from markers attached to body segments. Therefore we can use eq. 1.13 to compute $\mathbf{T_c}$ during each video frame.

> There is another way to think about $\mathbf{R_{prox}}$, $\mathbf{R_{dist}}$, and the transformation between them. $\mathbf{R_{prox}}$ and $\mathbf{R_{dist}}$ 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 $\mathbf{R_{prox}}$ and $\mathbf{R_{dist}}$ are sets of orthogonal unit vectors, there is a rotation $\mathbf{T_v}$ that move the vectors of $\mathbf{R_{prox}}$ into the vectors of $\mathbf{R_{dist}}$. We define
>
> $$\mathbf{T_v\ R_{prox} = R_{dist}}$$
>
> Therefore
>
> $$\mathbf{T_v\ R_{prox}\ R_{prox}^{-1} = R_{dist}\ R_{prox}^{-1}}$$
> $$\mathbf{T_v = R_{dist}\ R_{prox}^{-1}}$$
>
> But for rotation matrices, inverse = transpose, so
>
> $$\boldsymbol{T_v = R_{dist} R_{prox}}^{\boldsymbol{T}}$$

Now we (finally) have the measured $\mathbf{T_c}$. It can be calculated from marker motion capture data at each time point. The measured $\mathbf{T_c}$ 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 could rotate about axis 1, then axis 2, then axis 1 again (x-y-x, y-z-y, etc.). 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. Note that, when an Euler sequence is used, 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.

> Example: The International Shoulder Group made recommendations for axis definitions and joint motion for upper limb joints, including the "thoracohumeral" joint (Wu et al., J Biomech 2005). (The thoracohumeral joint does not really exist anatomically. There is a thoracoclavicular joint, a scapuloclavicular joint, and a glenohumeral joint. But the motion at these anatomically-real joints is hard to quantify, because the position and orientation of the clavicle and scapula are hard to measure. Therefore the hypothetical thoracohumeral joint, which skips the clavicle and scapula, is often analyzed.) ISG define +X pointing anteriorly, +Y pointing superiorly, and +Z pointing to the right, for the thorax and for the humerus in the resting position. ISG recommends (section 2.4.7) analyzing thoracohumeral motion as an internal/external rotation about Y, followed by an "elevation" about the rotated X, followed by a final internal/external rotation about the twice-rotated Y. This is a Y-X-Y sequence. This way of analyzing shoulder joint angles works well for motions that are principally ab/adduction (as traditionally defined), but it also has some disadvantages. See file matrices_rotations_2.doc for more detail, and see powerpoint presentation by James Richards for animations, and for comparison of the different sequences, and which ones work well for which types of movements.

We know the measured rotation matrix $\mathbf{T_c}$ from analysis of 3D motion capture data (equation 1.13). We find the values of ϕ, θ, and ψ to make the elements in the "analytical" rotation matrix $\mathbf{T_c}$ match the experimentally-measured $\mathbf{T_c}$. Equation 1.9, above, is the analytic rotation matrix for the sequence x-y-z, in which there is a rotation by angle ϕ about the x axis, then by angle θ about the y axis, and finally by ψ about the z axis. The analytic matrices for other sequences, such as y-x-y, z-y-x, etc., are different than the matrix in eq. 1.9. Different rotation sequences will require different angles to match the experimental $\mathbf{T_c}$, due to the non-commutativity of the 3D rotation matrices.

Gimbal lock

Gimbal lock is a problem that can occur with three-dimensional rotations described by Cardan sequences. It occurs when the "middle angle", the angle of the second rotation, is 90°. The middle angle is called θ above. When θ = π/2, then cos θ=0 and sin θ=1, the T matrix above simplifies to

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

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 ϕ and ψ . (It is easy to figure out θ: if we experimentally measure a T matrix in which t_{31}=1, then, by comparison to the analytic T matrix above (which is for rotation about x, then y', then z"), we see that sin θ=1, which means θ= π/2.) The reason we cannot determine ϕ and ψ is that the elements of T predicted by the theory of rotations, when θ= π/2, as shown above, don't depend on ϕ and ψ individually; they only depend on the sum of ϕ and ψ. So our experimental measurement, in this situation, allows us to know the <u>sum</u> of ϕ and ψ , but not their individual values. (ϕ+ψ=arcsin(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 ϕ and ψ 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.

You may think that gimbal lock is not something to worry about, since it is highly unlikely that we would ever experimentally observe a $\mathbf{T_c}$ matrix with t_{31}=1.000 and t_{11}=t_{21}=t_{32}=t_{33}=0. Fair enough. But we might observe a matrix pretty close to that, and if we do, the estimated values assigned to ϕ and ψ will become very sensitive to small measurement errors. The shoulder joint is one at which gimbal lock problems may occur. If the humerus is abducted 90°, i.e. straight out from the trunk, and if our rotation sequence has ab/adduction as the middle rotation, we have a gimbal lock problem. This position, or quite close to it, occurs during an overhand throw or tennis serve. A solution might be to describe the motion using helical axes, or by a different sequence, such as the Euler sequence recommended by the ISG (Wu et al., J Biomech 2005).

Appendix: Winter's Approach to 2D Coordinate System Transformations

This section is similar to the section above on two-dimensional rotations. but it uses the notation and approach of D.A.Winter, 3rd edition, 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 **R** and **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 **R** for rotation matrices, but Winter uses **A** in these sections of his book. This section of the notes follows Winter's notation.

Suppose a point **a** has coordinates $\mathbf{R_a} = \begin{pmatrix} X_a \\ Y_a \end{pmatrix}$ 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 θ. The coordinates of point **a** in LRS1 are $\mathbf{r_{1a}} = \begin{pmatrix} x_{1a} \\ y_{1a} \end{pmatrix}$

(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.)

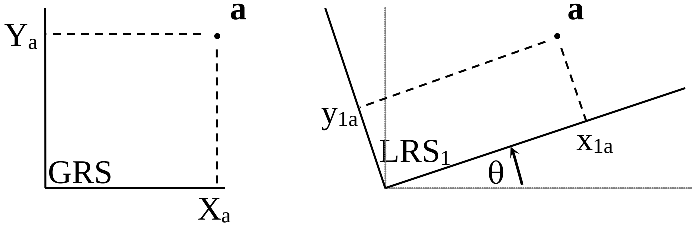

Fig. A1. Left: Point **a** in the Global Reference System coordinates. Right: Point **a** in local reference system 1 coordinates, with GRS axes shown for reference.

How are $\mathbf{R_a}$ and $\mathbf{r_{1a}}$ related? One can prove the following:

$$\begin{aligned} X_a &= x_{1a}\cos\theta - y_{1a}\sin\theta \\ Y_a &= x_{1a}\sin\theta + y_{1a}\cos\theta \end{aligned} \qquad \text{(local to global) (A.1)}$$

This can be rewritten with matrices:

$$\begin{pmatrix} X_a \\ Y_a \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x_{1a} \\ y_{1a} \end{pmatrix} \qquad \text{(A.2)}$$

or

$$\mathbf{R_a} = \mathbf{A}\,\mathbf{r_{1a}} \qquad \text{(local to global) (A.3)}$$

where **A** is the rotation-of-coordinates matrix

$$\mathbf{A} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} = \begin{pmatrix} \mathrm{c} & -\mathrm{s} \\ \mathrm{s} & \mathrm{c} \end{pmatrix} \qquad \text{(A.4)}$$

(c=cosθ, s=sinθ). The first column of **A** is the global coords that **i** would have after it (the vector **i**, not the coordinate system) is rotated by angle +θ. The second column of **A** is the global coordinates that **j** would have after being rotated by angle +θ.

There are two different interpretations of A, both valid.

1. **A** moves a vector by angle +θ. $\mathbf{Ar_1}=\mathbf{r_2}$, where $\mathbf{r_1}$ and $\mathbf{r_2}$ are the coordinates of the vector before and after rotation by +θ. There is no change of coordinate system.

The matrix **A** is the same as the matrix $\mathbf{R_v}$(θ) in equation 1.1. It is the transpose, or inverse (they're the same for rotation matrices), of the matrix $\mathbf{R_c}$(θ), in equations 1.3 and 1.4.

2. Matrix **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 **Ar**=**R**, where **r** and **R** are the same point, expressed in LRS and GRS respectively. The LRS is rotated by +θ with respect to the GRS. **A** doesn't move the vector. Why aren't **A** and $\mathbf{R_c}(\theta)$ the same, since they both describe rotations of coordinate systems? Because $\mathbf{R_c}(\theta)$ goes from GRS to LRS and **A** goes from LRS to GRS.

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.

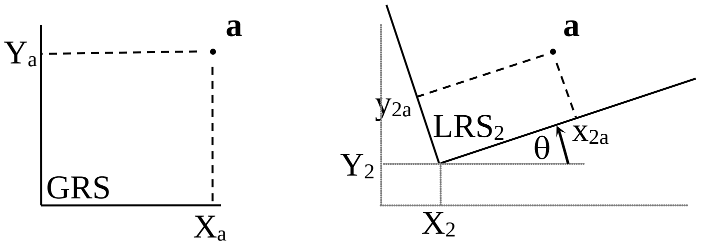

Fig. A2. Left: Point **a** in the Global Reference System coordinates. Right: Point **a** in local reference system 2 coordinates, with GRS axes and offset shown for reference.

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:

$$\begin{pmatrix} X_a \\ Y_a \end{pmatrix} = \begin{pmatrix} X2 \\ Y2 \end{pmatrix} + \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x_a \\ y_a \end{pmatrix}$$

(some subscripts in above eqn missing - due to bug in how MS Word saves eqn as HTML?) (A.4)

or

$$\mathbf{R_a} = \mathbf{R_2} + \mathbf{A}\,\mathbf{r_{2a}} \tag{A.5}$$

As before, the <u>columns</u> of **A** are the coordinates of **i** and **j** after rotation by angle +θ. As before, **A** may be thought of as

1. The matrix that rotates a vector by +θ, when it multiplies that column vector. It does not change coordinate systems.
2. The matrix that changes a vector's representation (when A multiplies that column vector) from LRS coordinates to GRS coordinates, where the LRS is rotated by +θ relative to the GRS. Note that this is different from the matrix $\mathbf{R_c}$ in equations 1.3 and 1.4 above. $\mathbf{R_c}$ converts a vector representation in the oppostite direction: from the GRS to the LRS.