

Parametrization and range of motion of the ball-and-socket joint

Paolo Baerlocher, Ronan Boulic
Computer Graphics Lab (LIG)
EPFL - Lausanne, Switzerland

Abstract

The ball-and-socket joint model is used to represent articulations with three rotational degrees of freedom (DOF), such as the human shoulder and the hip. The goal of this paper is to discuss two related problems: the parametrization and the definition of realistic joint boundaries for ball-and-socket joints. Doing this accurately is difficult, yet important for motion generators (such as inverse kinematics and dynamics engines) and for motion manipulators (such as motion retargeting), since the resulting motions should satisfy the anatomic constraints. The difficulty mainly comes from the complex nature of 3D orientations and of human articulations. The underlying question of parametrization must be addressed before realistic and meaningful boundaries can be defined over the set of 3D orientations. In this paper, we review and compare several known methods, and advocate the use of the swing-and-twist parametrization, that partitions an arbitrary orientation into two meaningful components. The related problem of induced twist is discussed. Finally, we review some joint boundaries representations based on this decomposition, and show an example.

1. Introduction

In fields such as robotics [8] and biomechanics, and in Computer Animation as well [11], hierarchical structures are used to model articulated bodies like (real or imaginary) robots, humans and other creatures. An articulated body is made of a set of segments, connected by *joints*. The essential feature of a joint is that it permits some degree of relative motion between the two segments it connects. Ideal kinematic joint models are defined in order to formalize this permitted relative motion, called *range of motion*, characterized by the number of parameters that describe the motion space, and constrained by joint limits. Modeling real joints can be very complex, since the range of motion depends on many factors, especially in the articulations of living organisms and the human in particular [2]. Moreover, joints may be dependent on each other, especially in living organisms. This coupling (of motion and limits) can be integrated directly in the body definition, with the concept of *joint group* [2], or at the application level, with kinematic constraints resolved by an inverse kinematics engine (for example, the scapulo-thoracic constraint [7]). In this paper, the coupling between joints is ignored.

The simplest example of joint model is the *revolute joint* that allows a rotation about an axis fixed in both segments it connects, usually within some angular limits. This joint is said to have one degree of freedom (DOF) and, because of its simplicity, is by far the most used joint in robotics. In human modeling, it is a convenient model of the interphalangeal joints of the hands and feet, for example. For more complex articulations such as the shoulder and hip, joint models allowing more degrees of freedom are required. Unfortunately, the accurate kinematic modelling of such articulations is a difficult task. First, a clear mathematical description of the allowed relative motion must be given by a proper parametrization: because of the complex non-Euclidean nature of rotations, this must be done carefully, or one may incur in the problem of singularities. Second, the range of motion should be constrained by some joint limits, to restrict the parameter space to some more realistic subset. For instance, a revolute joint whose configura-

tion is given by the current rotation angle θ may have some minimum and maximum values: $\theta_{min} \leq \theta \leq \theta_{max}$. The situation is more complex with ball-and-socket joints, because the boundaries on the three independent parameters are generally coupled.

Once a proper parametrization and proper joint limits have been defined for each joint of the articulated body, the animation engine (such as an inverse kinematics engine), has to deal with this model, for example to ensure joint limits. The last part of the problem is to gather data from real persons, if the goal is to model ranges of motions of human articulations. In Computer Animation, these topics have already been addressed by Badler [1, 2], Korein [5], Wang [9, 10], Grassia [4] and Maurel [7]. In this paper, we summarize and compare their results, and try to provide some more insight on the topics.

1.1. Notation and conventions

In this paper, we use the *column vector* convention. Moreover, we use *right-handed coordinate frames*. Vectors are denoted by small boldface letters such as \mathbf{v} . The three basis vectors of a coordinate frame are noted \mathbf{x} , \mathbf{y} and \mathbf{z} . Matrices and points are denoted by capital letters such as M .

A few operators are now defined.

- *Rotation about an axis by an angle*

The rotation (in the right-handed sense) by an angle θ about an axis passing through the origin and whose direction is given by vector \mathbf{a} , is noted $R_{\mathbf{a}}(\theta)$. Note that $R_{\mathbf{a}}(\theta) = R_{-\mathbf{a}}(-\theta)$.

- *Direct rotation*

Given two unit vectors \mathbf{a} and \mathbf{b} , we define $R_D(\mathbf{a}, \mathbf{b}) := R_{\mathbf{a} \times \mathbf{b}}(\text{acos}(\mathbf{a}^T \mathbf{b}))$ as the *direct rotation* that transforms \mathbf{a} into \mathbf{b} (see Figure 1). If $\mathbf{b} = -\mathbf{a}$, $R_D(\mathbf{a}, \mathbf{b})$ is undefined.

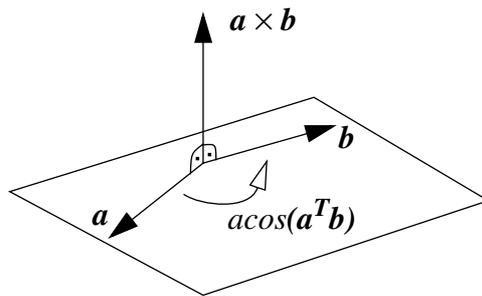


Figure 1: A direct rotation $R_D(\mathbf{a}, \mathbf{b})$ transforms a unit vector \mathbf{a} into a unit vector \mathbf{b} .

Note that any rotation whose axis of rotation lies in the bisector plane of \mathbf{a} and \mathbf{b} , with the appropriate angle of rotation, transforms vector \mathbf{a} into vector \mathbf{b} . The direct rotation is the one with minimum angle of rotation, that is $\text{acos}(\mathbf{a}^T \mathbf{b})$. Moreover, with a direct rotation, no “twisting” occurs about the rotated vector, in the sense that any vector lying in the plane of rotation is transformed into another vector still lying in that plane.

2. Parametrization of a ball-and-socket joint

A ball-and-socket joint possesses three rotational degrees of freedom. Hence, it is the most mobile of the purely rotational joints. It allows an axial motion (or *twist*) of the segment (one DOF), as well as a *spherical* motion (or *swing*) that determines its direction (two DOFs). Ball-and-socket joints are used to model articulations such as the human shoulder and hip. A mechanical illustration of this joint is given in Figure 2. By convention, in the following discussion the moving segment is aligned with the z axis of the local joint frame.

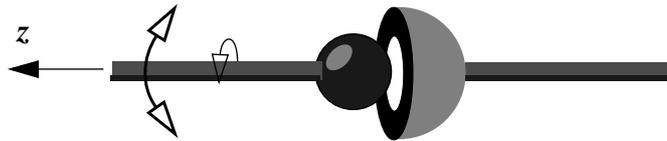


Figure 2: Illustration of a ball-and-socket joint.

2.1. Parametrization of rotations

The motion space of a ball-and-socket joint is the set of 3D rotations. There are many well-known parametrizations of rotations. The most widely used are:

- the *Euler angles* (the angles of three successive rotations about principal axes)
- the *unit quaternion* (also known as the *Euler parameters*)
- the *axis-angle vector* (also known as the *exponential map* or *versor*).

Good comparisons of such parametrizations for the purposes of animation of articulated bodies can be found in [4] and [11]. As noted by [4], no single parametrization of rotations is best. Each one possesses its advantages and drawbacks, with respect to the intended application. Hence, it is likely that several parametrizations be used simultaneously, with conversions between them. For example, the unit quaternion is ideally suited for interpolation [11], while the axis-angle vector is a more appropriate parametrization for differential control with inverse kinematics [4]. Euler angles would not be a good choice in both applications. Instead, they are a more intuitive set of parameters to manipulate a ball-and-socket joint in a graphical user interface.

An important point to consider when comparing two parametrizations is the presence of singularities. Singularities are locations in the parameter space that result in the same orientation of the joint. Sometimes these singularities are purely mathematical and only result from the choice of parametrization, but they may also reflect a physical reality. In that case, we encounter the problem known as gimbal lock [11, 4]. Because of the problems induced by the singularities not only at the singular point but often also in their neighborhood, the configuration of a joint should always be kept as far as possible from these points.

Unfortunately, any three-dimensional parametrization of rotations present at least one singularity [8]. Those of the Euler angles and of the exponential map are discussed in [4], and will be recalled later. The unit quaternion parametrization is singularity free, but at the cost of using four parameters instead of three, with a quadratic constraint of unitary norm that must then be ensured [4].

2.2. Parametrization for the purpose of range of motion definition

For the purpose of defining a range of motion, an appropriate parametrization is needed. Certainly, one can impose limits on any parametrization. For example, it is possible to impose limits on Euler angles or on quaternion parameters. For example, Lee [6] describes simple analytical constraints (such as axial, spherical or conical constraints) enforced directly in quaternion space. More complex constraints can then be defined by combining the simple ones with boolean operators. While simple and elegant, this method is not precise enough for an accurate modelling of the limits of complex joints such as the shoulder, and placing more complex meaningful limits on quaternions is difficult.

To simplify the problem, the joint limits may be decoupled. For example, independent limits may be specified on each Euler angle, or on each element of the axis-angle. However, the resulting range of motion can hardly match real motion ranges with sufficient precision [7].

For the purpose of defining a range of motion, neither the axis-angle or the unit quaternion reflect the intuitive decomposition of the rotation into a swing and a twist component. Euler angles do, since the third angle may be used to perform the twisting motion. However, in the following sections we see that the first two Euler angles can be replaced by an axis-angle vector with zero component along the z axis: this allows to alleviate the problem of singularities that affects the Euler angles.

2.3. The swing and twist decomposition of an orientation

Intuitively, the orientation R of a ball-and-socket joint can be thought as being composed of a swing component, that controls the direction of the limb directly attached to it, and a twist component that lets the limb rotate about itself [5, 4]. This may be written as: $R = R^{Twist} R^{Swing}$

The twist component is easily parametrized by a single angle of rotation, noted τ : hence, $R^{Twist} = R_z(\tau)$. However, this rotation must be done with respect to a well-defined orientation, here called the *zero twist reference orientation*. In fact, this reference orientation merely results from the swing rotation, and is not necessarily a good reference. Hence a relative twist, τ_{offset} , as a function of the swing parameters, can be added. An example of such an offset function is given by Badler [1].

The purpose of the swing motion is to orientate the outgoing limb in a prescribed direction given by a unit vector \mathbf{d} . To transform the \mathbf{z} vector into the \mathbf{d} vector, a rotation matrix R^{Swing} must be defined. We consider two solutions.

- The first is to perform two successive rotations, for instance one about the x axis and then a second one about the rotated y axis: $R^{Swing} = R_y(\beta)R_x(\alpha)$. This is equivalent to the first two rotations of the ZYX Euler angles sequence [8] (Figure 3).
- The second is to perform a single, direct rotation: $R^{Swing} = R_D(\mathbf{z}, \mathbf{d})$ (Figure 4). Note that the axis of rotation always lies in the XY plane.

The second solution has been used by Korein [5] and Grassia [4]. However, Korein parametrizes this rotation with two angles, called *halfplane* and *deviation*, that are the spherical coordinates describing the direction vector \mathbf{d} , while Grassia uses the x and y components of the axis-angle, here noted S_x and S_y .

As already noted by Korein [5], the difference between the two rotations lies in the final twist about the d axis, which is given by the different orientations of the rotated x and y vectors. Table 1 shows a sampling of the zero twist on the sphere for the two parametrizations: the outgoing arrow at each point on the sphere indicates the direction of the rotated x axis, which is taken as a reference to indicate the twist.

As said before, the singularities of a parametrization must also be considered, because the presence of singularities may be problematic for several applications. For the purpose of defining a range of motion, the twist component is affected by a singularity of the swing component: for example, no zero twist may be defined at a singularity, since an infinity of twists are possible. An arbitrary twist may be assigned to this point, but there is still a discontinuity with respect to its neighborhood. Table 1 compares the position of the singularities on the sphere, while the corresponding locations in the parameter space are shown in Figure 5, and the next two sections discusses and compares them.

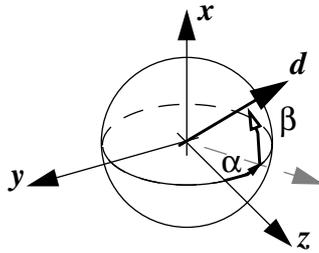


Figure 3: Euler angles parametrization of the swing motion.

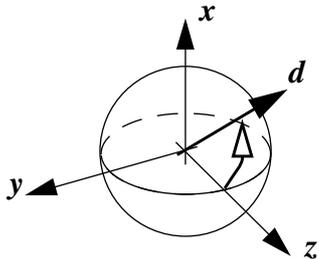


Figure 4: Axis-angle parametrization of swing (a direct rotation about an axis lying in the x - y plane).



Figure 5: Singular locations of the Euler angles parametrization ($\beta = \pm\pi/2$) on the left, and of the axis-angle parametrization (a circle of radius π) on the right.

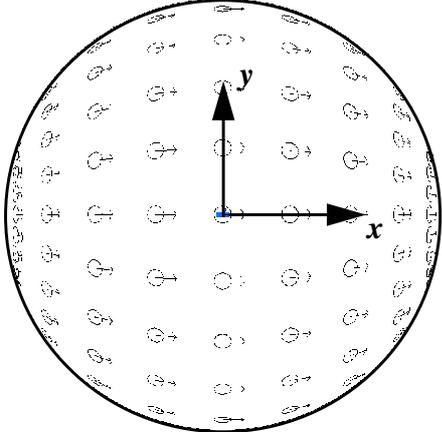
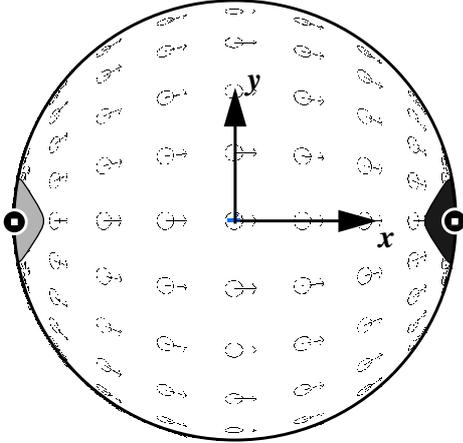
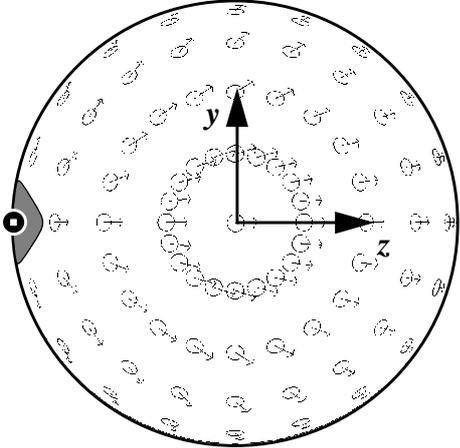
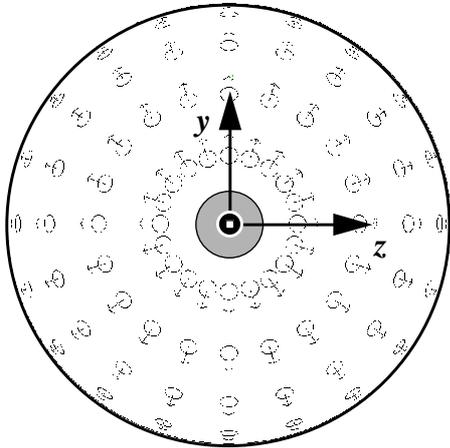
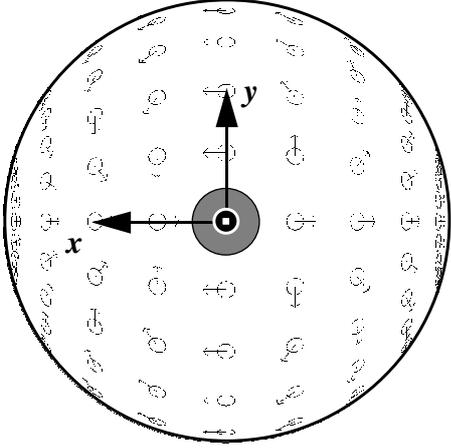
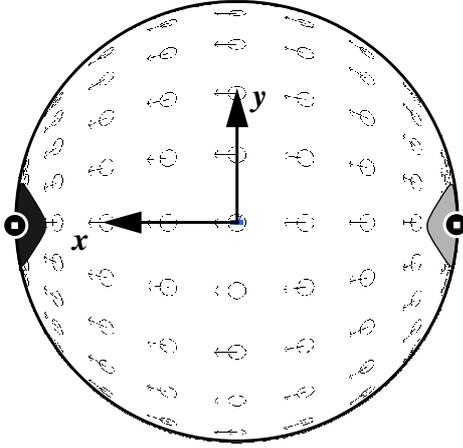
View	Axis-angle parametrization of swing (one singularity)	Euler angles parametrization of swing (two singularities)
Front		
Side		
Rear		

Table 1: Comparison of “zero” twist and singularities (●) for two parametrizations of swing. The vicinity of each singularity is marked by a grey area with a different shade.

2.4. Singularities of the XY Euler angles swing parametrization

This parametrization possesses two singularities: one at $\beta = \pi/2$ and another at $\beta = -\pi/2$. In Cartesian space, these singularities correspond to directions $\mathbf{d}_1 = [1 \ 0 \ 0]^T$ and $\mathbf{d}_2 = [-1 \ 0 \ 0]^T$ respectively, and any twist is possible there. Furthermore, moving close to these directions results in wild variations of twist. For example, moving along a closed path close to, and around the singularity, results in a complete rotation of the segment about itself (i.e. a twist of 2π radians).

Note that another convention of rotation axes could have been chosen. For example, one can perform a first rotation about the z axis, and then a second one about either the rotated x or y vector. In this case the singularities are located on directions $\mathbf{d}'_1 = [0 \ 0 \ -1]^T$ and $\mathbf{d}'_2 = [0 \ 0 \ 1]^T$. This is equivalent to our original choice, up to a rotation by 90° about the y axis, but having a singularity exactly at the initial configuration is not a good idea.

To understand the meaning of the singularities, consider a universal joint, made as a sequence of two revolute joints whose axes of rotation are orthogonal, as shown in Figure 6. A rotation about the x axis or the y axis changes the direction of the outgoing segment, and apparently no twisting is performed. However, this is not always true. When $\beta = \pm\pi/2$, which is the angle of rotation about the y axis, the outgoing segment becomes aligned with the x axis (Figure 7): as a consequence, a change in α does not change its direction anymore, but its twist. Actually, any twist is possible in this direction, but the segment cannot move up and down anymore. This phenomenon is known as Gimbal Lock, and is a well-known flaw of Euler angles [11]. Also note how the vertical swing component (along the x axis) gradually transforms into a twist of the outgoing limb, as the singular configuration is approached. This shows that the problem not only exists at the singularity, but also in its vicinity.

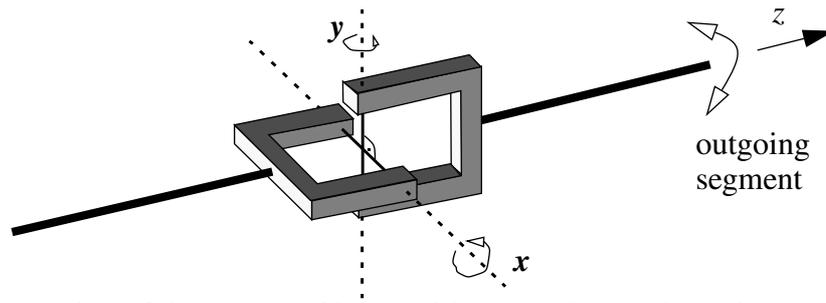


Figure 6: Illustration of the *universal* joint, with two orthogonal rotation axes.

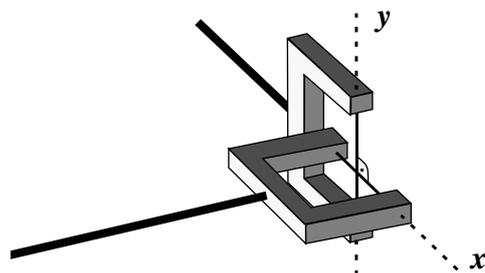


Figure 7: The universal joint in a singular configuration ($\beta = \pi/2$): the outgoing segment becomes aligned with the x axis of rotation.

Ball-and-socket joints are often built as a series of three revolute joints with intersecting axes: two for swinging (as in a universal joint), and one for twisting. Thus, it also experiences the singularities of the universal joint. An example of this is the very common joint used to transmit a torque for the control of a window blind. At a singularity however, a twisting torque is completely transformed in a swinging torque applied to the outgoing segment. Thus no twist can be transmitted anymore. Moreover, in a dynamics simulation, this singular configuration could lead to numerical problems: a torque applied about the two aligned axes of rotation of the joint may result in an infinite acceleration since mass is usually not present between the axes of the same joint. Fortunately, for simulation purposes we can choose another parametrization, such as the axis-angle parametrization.

2.5. Singularities of the *axis-angle* swing parametrization

The axis-angle possesses only one singularity on direction $\mathbf{d} = [0 \ 0 \ -1]^T$, where $S_x^2 + S_y^2 = \pi^2$. Again, any twist is possible there. However, this singularity is more “severe” since a closed path close to, and around the singularity, performs two complete rotations of the segment (i.e. a twist of 4π radians).

A geometric interpretation of this singularity is the following. Consider the problem of finding the shortest path between two given points lying on a unit sphere. The solution is the great arc connecting these two points. This solution is always unique, except when we deal with two antipodal points, since there is an infinity of great arcs between them. This corresponds to the singular situation of our swing parametrization. Now, when we are close to this singular situation, notice how a small change of one of the two points may result in a dramatic change of the solution. Hence solving the problem at the singularity, for example by choosing an arbitrary solution among the valid ones, does not necessarily solves the problem in the vicinity of the singularity.

To summarize, the axis-angle parametrizations is preferable to the Euler angles parametrization, since it is easier to avoid one single singular point than two antipodal singular points on the sphere. To be as far as possible from the singularity, the motion range should be centered about the z axis in its default configuration, or at least the singular point should not be part of the motion range.

2.6. The occurrence of induced twist

Direct rotations are a desirable way of performing a swing, since the z vector of the local joint frame is rotated to reach a given direction, without any twist being performed about that axis. However, a direct rotation performed from a direction other than the default z vector not only affects the swing component of the joint but also its twist component, by an amount called the *induced twist*, noted $\tau_{induced}$. This phenomenon has been previously discussed by Badler and Korein [5, 2]. It is not a major problem, but it has to be dealt with. As shown in Figure 8, for arbitrary unit vectors \mathbf{a} and \mathbf{b} , we have that $R_D(z, \mathbf{a})R_D(\mathbf{a}, \mathbf{b}) \neq R_D(z, \mathbf{b})$. A solution is to extract the induced twist from the rotation matrix $R_z(\tau_{induced}) = R_D(z, \mathbf{a})R_D(\mathbf{a}, \mathbf{b})R_D(\mathbf{b}, z)$, and then to subtract it from the twist variable τ [5]. In [2], it is shown how the induced twist can be computed, and thus removed, when using the Euler angles parametrization. Another possibility is to perform the update of the joint orientation with a direct rotation, and then to convert the resulting orientation back into the swing and twist parametrization (see Appendix I), to check for possible joint limits violations.

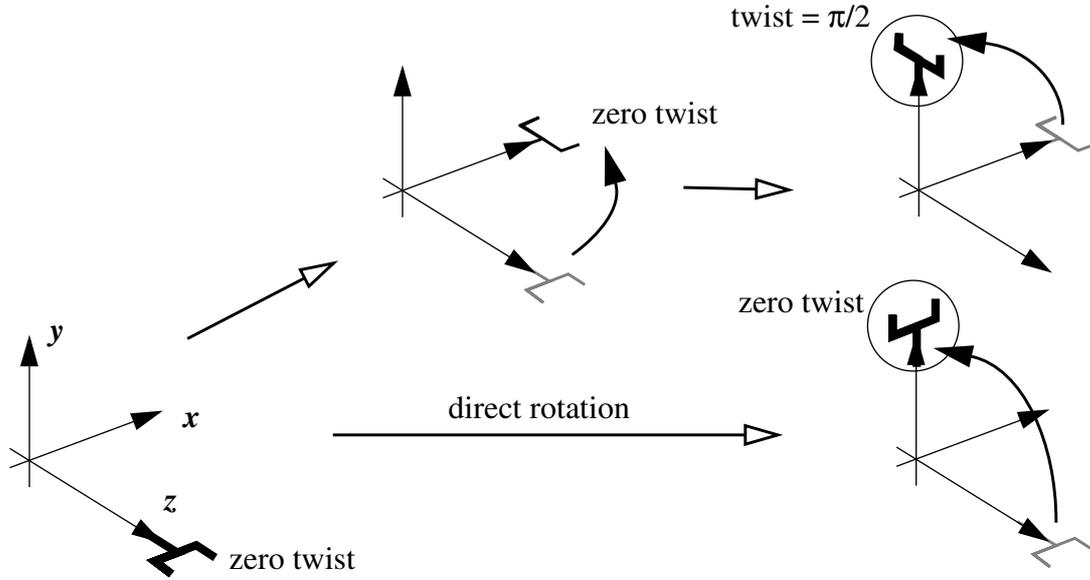


Figure 8: Reaching a target direction (here, the y axis) either with a direct rotation (bottom) or with a combination of two direct rotations (top) does not result in the same final twist. The difference is called the *induced twist* (here: $\pi/2$ radians).

3. The definition of realistic joint limits

Based on the swing and twist decomposition, it is possible to impose independent limits on both components. The limits of the swing component are best visualized as a curve on a sphere centered at the joint center. This curve delineates the valid region for the limb, and can be seen as the directrix of a general conical surface whose vertex is the center of the joint. In the following two sub-sections, we review two possible methods for defining this curve. The third section discusses the limits of the twist component.

In the following, note that the parametrization of the joint boundaries may, but not necessarily have to, be based on a particular parametrization of the swing component (such as Euler angles or axis-angle).

3.1. Swing function : the spherical ellipse

An analytical method is to use a function $f(S_x, S_y)$ which is negative only for valid swings $[S_x \ S_y]^T$. A simple example given in [4] is an ellipse with semi-axes r_x and r_y , that describe the maximum angle of rotation around the x axis and the y axis respectively: in this case, the function is given by $f(S_x, S_y) = (S_x/r_x)^2 + (S_y/r_y)^2 - 1$, with $r_x < \pi$ and $r_y < \pi$. This results in a “spherical” ellipse in the Cartesian space. Two illustrations of such a curve are given in Figure 9 (the ticks on the spherical ellipse indicate its “inside” region).

The advantage of the spherical ellipse is that, with a minimum of parameters, a meaningful boundary can be defined for the swing component.

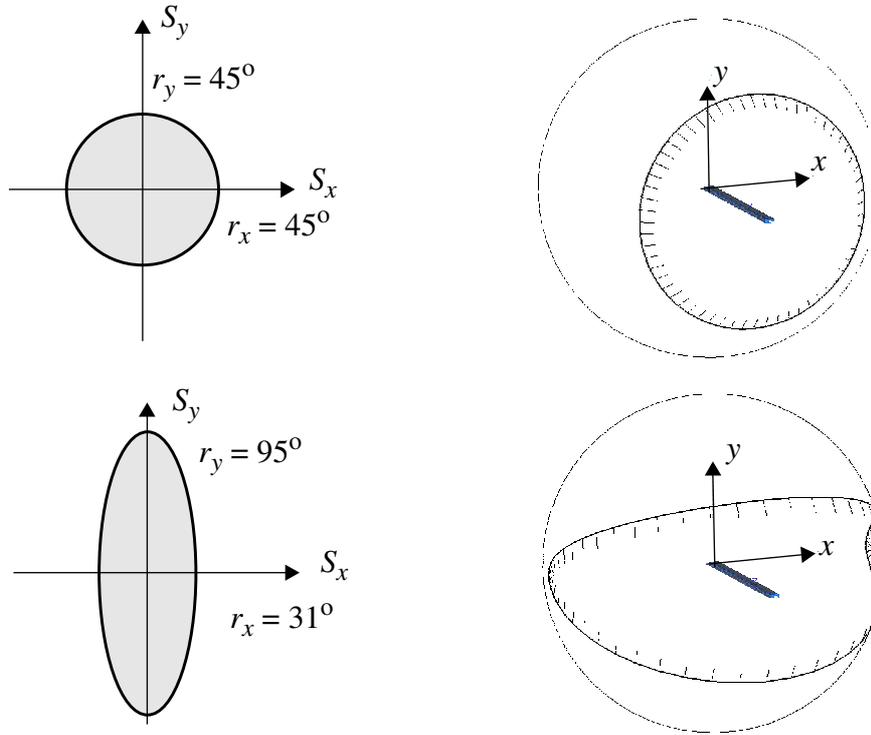


Figure 9: Two examples of spherical ellipses.

3.2. Spherical polygons

In his excellent book, Korein [5] uses a spherical polygon as directrix for the limiting cone. The edges of the spherical polygon are great arcs connecting its vertices lying on a unit sphere, and specified by three Cartesian coordinates¹. A great arc is the shortest path that binds two points on a sphere (it is a *geodesic*). Note that, by definition, only great arcs that subtend angles less than π radians are possible. The order of the vertices defines an inside region: inverting this order swaps the inside and outside regions of the polygon. A spherical polygon with five vertices is shown in Figure 10. Its inside region is filled. Korein described an algorithm to test the inclusion of a point lying on the sphere within an arbitrary (possibly concave) spherical polygon.

Of course, spherical polygons are more general than spherical ellipses. They are also more complex to deal with. A similar method has been used by Maurel [7], but with planar polygons. As a consequence, the possible motion ranges are less general than those obtained with spherical polygons. However they may suffice for the human joints, and the *point-in-planar-polygon* test algorithm is much simpler than its spherical counterpart.

1. Two spherical coordinates would also suffice to specify a vertex on a unit sphere, however they would have to be converted into Cartesian form for subsequent computations, which requires expensive trigonometric calculations.

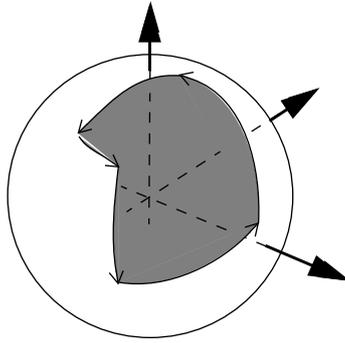


Figure 10: A spherical polygon with five directed edges.

More general boundaries can be obtained simply by using several non-overlapping spherical polygons: the set of admissible points on the sphere is then the set of points that are inside all the spherical polygons. This allows to create holes in the admissible space (see Figure 11). However, this possibility is not needed for human articulations.

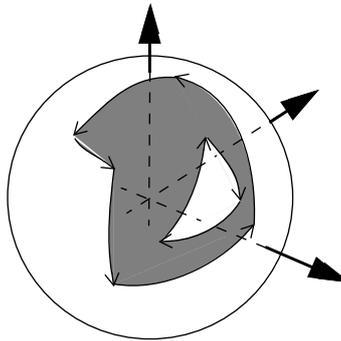


Figure 11: Two spherical polygons define a complex admissible spherical region (with a hole).

3.3. Twist limits

The twist motion possesses a single degree of freedom, parametrized by the angle of rotation τ about the outgoing segment. The important point to remember when placing limits on this parameter is that they are relative to a “zero” twist, that results from the swing motion.

In the following globographic representations involving a twist freedom, the twist range of motion is visualized by the symbol depicted in Figure 12: the valid twist range indicates the orientations that can take the reference vector, namely the x basis vector of the joint frame.

In general, twist limits may depend of the swing component (see Section 5.1). For this reason, the twist limits may be defined as functions of the swing component: $\tau_{min}(S_x, S_y)$ and $\tau_{max}(S_x, S_y)$, with the requirement that $\tau_{min} \leq \tau_{max}, \forall$ valid swings S_x, S_y .

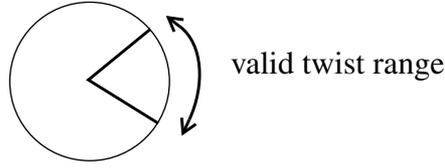


Figure 12: Representation of twist limits, for a given swing.

4. Enforcing joint limits

4.1. Clamping to the joints boundaries

When a given joint configuration must be checked against the boundaries, the first operation to do is to convert the proposed orientation into its equivalent swing-twist components (see Appendix I). Then a first test can be performed to know if the swing is in its range of motion. If it is not, most applications require that an alternative, close, valid configuration be supplied. This process is called *clamping*. Again, this is more complex to do for ball-and-socket joints than for single-DOF joints, since this requires to orthogonally project the outgoing segment on a spherical ellipse or on a spherical polygon. In general, the clamping should take place in Cartesian space, with a direct rotation that transforms the outgoing segment into its closest valid configuration. A priori, clamping on a spherical ellipse can be performed directly in the axis-angle space of the swing component, by projecting the invalid swing on the 2D ellipse (see Appendix II). This may be simpler than performing the computation in Cartesian space: however it can be objected that the use of the Euclidean distance to find the closest projection in the axis-angle space does not result in an exact orthogonal projection in Cartesian space, as could be naively expected. Moreover, since only the swing component has been adjusted, the twist induced by this relative adjustment must then be compensated.

Once the clamping of the swing component (S_x, S_y) has been performed, the twist angle τ must be clamped into its valid interval $[\tau_{min}(S_x, S_y) .. \tau_{max}(S_x, S_y)]$, so that a final, valid orientation can be computed and assigned to the joint.

4.2. Incremental (or relative) update of the configuration

If the joint boundaries are concave, another problem is raised when moving from a valid configuration to a new one: the great arc connecting those two points may not be fully contained in the valid region, even if the two extremities are. This may cause some unexpected behaviors: for example, the distal segment abruptly moves from a valid region to another, which are close in space but that require a long path to be connected (as shown on figure 13).

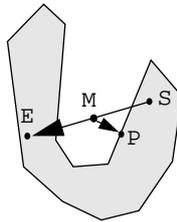


Figure 13: Computation of the next valid point (P), from a valid initial configuration (S) and a computed new configuration (E).

A correct solution would be to check if the great arc is entirely contained in the valid region. However this solution is complex and is prone to numerical inaccuracies when an extremity is very close to the joint boundaries. Therefore, we suggest a much simpler, yet approximate, solution. Instead of testing only the final configuration (E), we propose to test the middle point (M) of the great arc as well. If both final and middle point are valid, then we keep the final configuration. However, if the middle point is not valid, we keep its projection (P) on the joint boundary. This simple discretization of the arc has the effect of reducing to a minimum the unexpected behaviors described above, at least for reasonably small increments.

5. Examples

5.1. An example of shoulder boundary with swing and twist components

Figure 14. shows two boundaries for the shoulder complex, based on a spherical ellipse on the left and on a spherical polygon on the right. The distal segment (the arm) is shown in its default position. The twist limits are constant over the range of swing motion (the twist motion range is about 105°). However, in reality the twist limits depend on the position of the arm, and the range can vary between 104° and 160° on average [10]. The data for the spherical polygon, which are also used by [9] and [7], are obtained from the results of Engin [3].

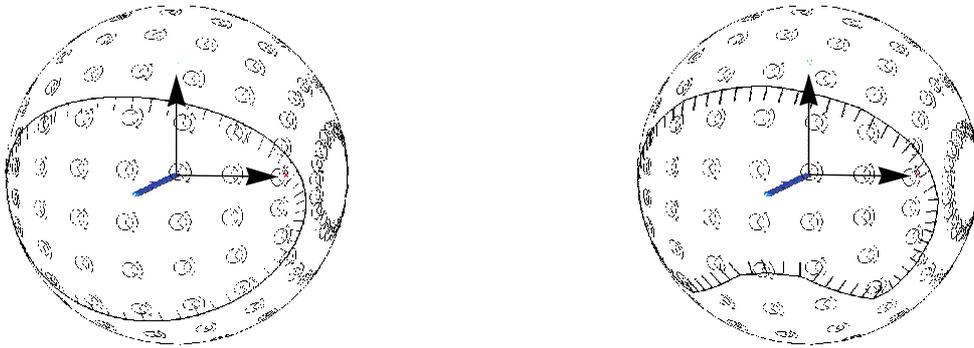


Figure 14: Shoulder motion range, with a spherical ellipse (left) and a spherical polygon (right).

5.2. Speed results

On a SGI Octane with a 195 MHz R10000 processor, our implementation of the *point-in-spherical-polygon* algorithm described by Korein takes approximately 0.01 ms to 0.05 ms, for polygons with 4 to 200 edges. This efficiency is in part due to the very limited use of trigonometric functions required to deal with spherical polygons, at least when its vertices are expressed in Cartesian coordinates.

6. Conclusion and future work

The swing-and-twist parametrization is useful for dealing with ball-and-socket joints, and is a good basis for the definition of simple yet meaningful joint limits. This decomposition had been previously discussed by Korein [5] and Grassia [4]. In this paper, we have emphasized and illustrated the difference between the well-known Euler angles parametrization and the swing-and-twist parametrization.

We have reviewed and illustrated two methods to set joint limits on the swing and twist components. The spherical ellipse proposed by Grassia is a good compromise between simplicity and accuracy, while spherical polygons are more complex but can better match real limits.

However, this discussion has ignored the coupling between the limits of multiple joints. This may be problematic for a joint such as the human hip whose range of motion is clearly coupled to the range of motion of the knee, because of thigh muscles spanning both joints.

7. Acknowledgements

The research was supported by the Swiss National Research Foundation, grant 20-53809.98. Thanks to the vision group at LIG, Prof. Andrew Hanson and Walter Maurel for interesting discussions on the topic of orientations and shoulder joint limits models.

8. References

- [1] N. Badler, J. O'Rourke and B. Kaufman, "*Special problems in human movement simulation*", Computer Graphics (SIGGRAPH 80), 14(3), pp. 189 - 197, 1980.
- [2] N. Badler, C. Phillips, B. Webber, *Simulating Humans, Computer Graphics Animation and Control*, New York, Oxford University Press, 1993.
- [3] A.E. Engin, S.T. Tuemer, "*Three-dimensional kinematic modelling of human shoulder complex - Part I: Physical model and determination of joint sinus cones*", Journal of Biomechanical Engineering, 111, pp. 107 - 112, 1989.
- [4] F.S. Grassia, "*Practical Parameterization of Rotations using the Exponential Map*", Journal of Graphics Tools, 3(3), pp. 29 - 48, 1998.
- [5] J.U. Korein, "*A Geometric Investigation of Reach*", The MIT Press, Cambridge, 1985.
- [6] J. Lee, *A Hierarchical Approach to Motion Analysis and Synthesis for Articulated Figures*, Ph.D. thesis, Department of Computer Science, KAIST, 2000.
- [7] W. Maurel and D. Thalmann, "*Human Upper Limb Modeling including Scapulo-Thoracic Constraint and Joint Sinus Cones*", Computers & Graphics, Pergamon Press, 24(2), pp. 203 - 218, 2000.
- [8] R. Murray, Z. Li, S. Sastry, *A mathematical introduction to robotic manipulation*, CRC Press, 1994.
- [9] X. Wang and J.-P. Verriest, "*A geometric algorithm to predict the arm reach posture for computer-aided ergonomic evaluation*", The Journal of Visualization and Computer Animation, 9, pp. 33 - 47, 1998.
- [10] X.G. Wang, F. Mazet, N. Maia, K. Voinot, J.P. Verriest and M. Fayet, "*Three-dimensional modelling of the motion range of axial rotation of the upper arm*", Journal of biomechanics, 31(10), pp. 899 - 908, 1998.
- [11] A. Watt, M. Watt, *Advanced Animation and Rendering Techniques*, Addison-Wesley, ACM Press, 1992.

Appendix I: Decomposition of an orientation into its swing and twist components.

Problem: Let a given orientation be described by a unit quaternion $\mathbf{e} = (e_x, e_y, e_z, e_w)$. We want to decompose it into a swing, represented by the axis-angle $\begin{bmatrix} S_x & S_y & 0 \end{bmatrix}^T$, followed by a twist by an angle τ about the z axis, represented by the axis-angle $\begin{bmatrix} 0 & 0 & \tau \end{bmatrix}^T$.

Solution: If both e_z and e_w are zero, the orientation is at the singularity of the swing component: thus, an infinity of swings and twists exist that match the orientation so that one may arbitrarily choose a twist. Otherwise, the twist is $\tau = 2 \operatorname{atan}2(e_z, e_w)$.

Once the twist component is known, the swing component is computed as follows:

$$\begin{bmatrix} S_x \\ S_y \end{bmatrix} = \frac{2}{\operatorname{sinc}(\beta)} \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) \\ \sin(\gamma) & \cos(\gamma) \end{bmatrix} \begin{bmatrix} e_x \\ e_y \end{bmatrix} \text{ where } \gamma = \tau/2 \text{ and } \beta = \operatorname{atan}2(\sqrt{e_x^2 + e_y^2}, \sqrt{e_z^2 + e_w^2}).$$

Note that $0 \leq \beta \leq \pi/2$, and that $-2\pi \leq \tau \leq 2\pi$.

Definitions: $\operatorname{sinc}(x) \equiv \frac{\sin(x)}{x}$ (or 1 if x is zero). The $\operatorname{atan}2()$ function is as in the C math library.

Appendix II: Orthogonal projection of a point onto a ellipse in 2D.

Problem: Given an axis-aligned ellipse centered at the origin, with semi-axes lengths r_x and r_y , find the point on the ellipse which has the smallest Euclidean distance to a given arbitrary point P of coordinates P_x and P_y .

Solution: First find the real solutions λ_i to the following polynomial equation of the 4th degree:

$$P_y \lambda^4 + (f - e) \lambda^3 + (f + e) \lambda - P_y = 0, \text{ where } e = 2(r_y - r_x^2/r_y) \text{ and } f = 2P_x r_x / r_y.$$

This can be done analytically. Then compute the distance between each candidate point $Q_i \left(r_x \frac{1 - \lambda_i^2}{1 + \lambda_i^2}, r_y \frac{2\lambda_i}{1 + \lambda_i^2} \right)$ and the given point P , and keep the candidate with the smaller distance.

Method: This result is obtained by the minimization of the distance between point P and an arbitrary point $Q(r_x \cos(\alpha), r_y \sin(\alpha))$ on the ellipse. Instead of solving for α , we solve for $\lambda = \tan(\alpha/2)$: this substitution transforms a transcendental equation into a simpler polynomial equation.