BodyBuilder session 5

Advanced Segments and 3D Angles

Course Objectives

1. Advanced *Segment Definition*
    1. *Point functions*
    2. *Dead Band*
    3. *Anti-flip lines*
    4. *Linking segments*

2. *Segment Manipulation*
    2. *Attitude*
    3. *Rotation*
    4. *Align*

2. *3D Angles*
    - *Absolute Angles*
    - *Relative Angles*

Advanced Segment Definition

Point Functions

Segment Origin – The origin point of a segment can then be used in a calculation or to describe a new segment

- O(SegmentP) or SegmentP(O)
  - Either the number “0” or the letter “O” can be used

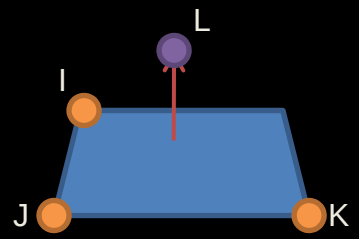
Segment Components – The three components of a segment are always 1:(X), 2:(Y), 3:(Z). These segment direction components can be called for used in other functions

- 1(SegmnetP), 2(SegmentP), or 3(SegmentP)
- SegmentP(1), SegmentP(2), or SegmentP(3)

Point Functions

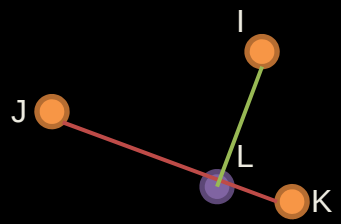
NORM - describes a direction perpendicular to the plane created by three points. The output will be a unit vector describing direction e.g. {1,0,0}. Because these are global coordinated, this point will show near the origin of the volume.

- PointL = NORM(PointI, PointJ, PointK)



PERP - describes a point at the perpendicular bisector from one point to a line created by two other points.

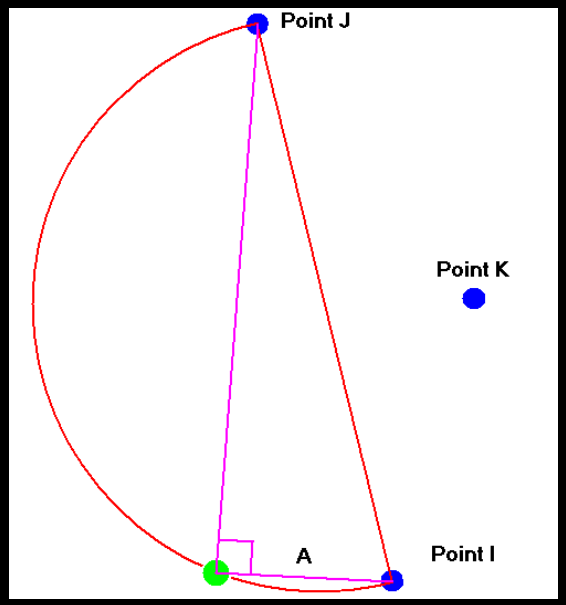
- PointL = PERP(PointI, PointJ, PointK)
  - Will find the point of intersection between a line dropped perpendicularly from pointI onto a line created by pointJ and pointK



The Chord Function

CHORD - used to find a virtual point by creating a chord of a circle from existing points

- Virtual = CHORD(numberA,pointI,pointJ,pointK)
  - Creates a virtual point at distance A from I in plane IJK forming a right angle between I and J on the opposite side of IJ from K see diagram:

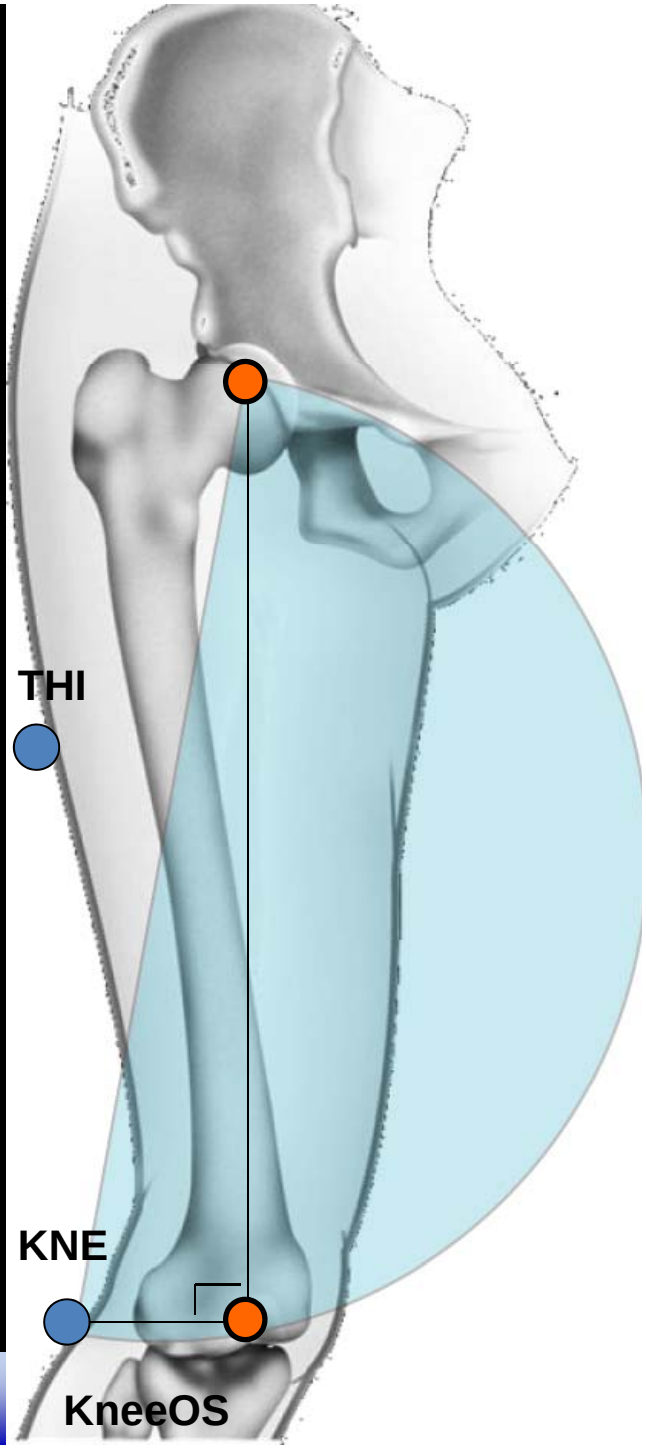


In the illustration above, the green point is created using the CHORD function. The line between points I and J is used as the diameter of a circle. The plane of the circle is the plane of this line and point K. Any point on the red semicircle will fulfil the right-angle criterion, and the distance A specifies the virtual green point indicated. The distance A is thus used in the construction as the chord of a circle, hence the name of the function.

The Chord Function for joint centres

The CHORD function may be useful in locating the centre of the knee joint given the hip joint centre and a marker on the mid-thigh and lateral condyle of the knee. This method is used in Plug in Gait gait analysis software.

$KneeOS = (*MarkerDiameter+KneeWidth*)/2
RKJC = CHORD($KneeOS,RKNE,RHJC,RTHI)



DeadBand

DeadBand – Because singularities can occur when markers used to define a segment become co-linear, users can define a deadband region which will:

- Check attitudes (orientation) of the segments on either side of the area
- Create smooth transition between segment axes if there are singularities

The default value for the dead band is 1° on either side of straight.

A dead band value can be set by including the following assignment anywhere in the script, either as part of the model itself, or more conveniently as a parameter.

```
DeadBand=numberA
```

Deadband is not required in Static trials and it is best to leave it out of this section due to subjects' outstretched limbs remaining in the deadband range the entire trial

DeadBand – test for singularities

To test the DeadBand is being applied correctly, you can use the special function: SINGULAR

```
IF SINGULAR(Tibia) DEAD={1,1,1}
OUTPUT(DEAD)
```

creates a point labeled DEAD at the location 1,1,1 in each field in which the segment Tibia is in the dead band

Anti-Flip lines

Anti-Flip Lines – Used to address hyperextensions where the shortest angle from the second defining line to the first becomes in front of the segment causing the axes to “flip”.

By specifying an anti-flip line, when an axes tries to “flip” the right-hand rule is reversed and the segment axes definition remains the same throughout the trial

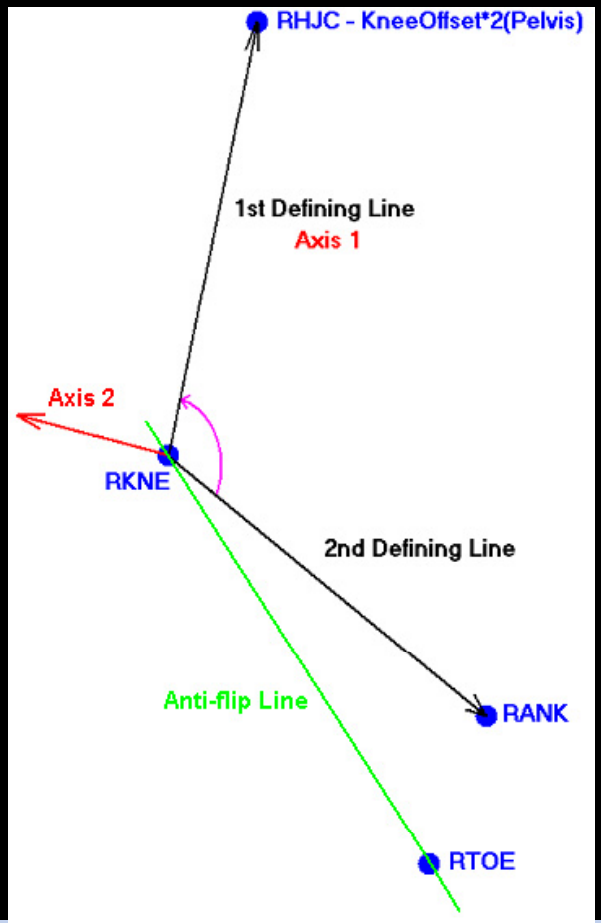
The anti-flip line is specified as part of the segment definition:

SegmentName = [Origin, DL1, DL2, token, anti-flip_line]

Anti-Flip lines – Knee example

RFemur=[RKNE,RHJC-KneeOffset*2(Pelvis)-RKNE,RANK-RKNE,zyx,RTOE-RKNE]

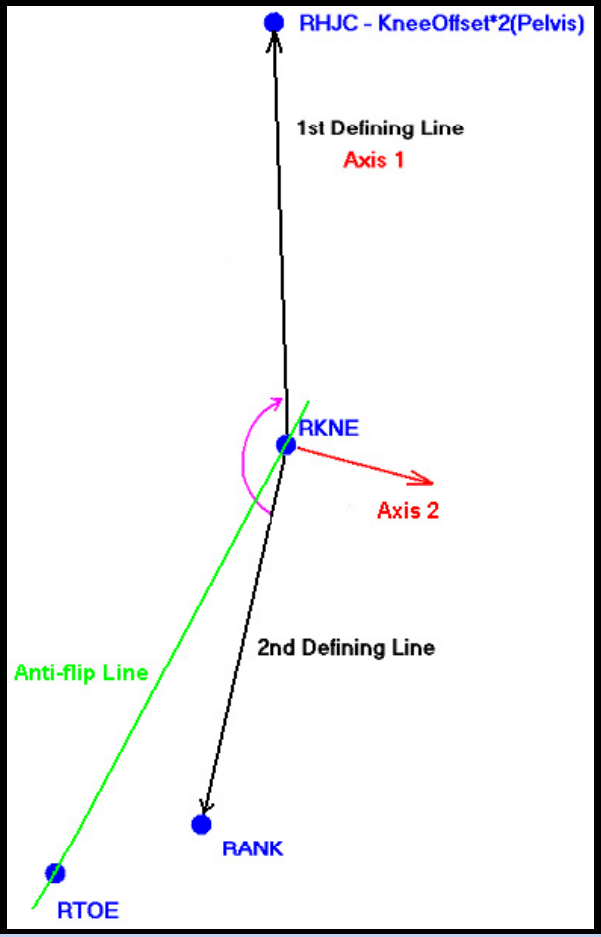
The anti-flip line is from the RKNE to the RTOE shown in Green
The shortest angle between the defining lines is the angle shown in Purple



In the standard case, when the right-hand rule is applied to the defining lines, the second axes is created as protruding out of the page.

Anti-Flip lines – Knee example

The anti-flip line is from the RKNE to the RTOE shown in Green
The shortest angle between the defining lines is the angle shown in Purple



When the joint hyperextends, the shortest angle between the defining lines in now in front of the joint. This causes the second axes to "flip" and protrude into the page.

However, because the shortest angle passes through the anti-flip line, the right-hand rule will be reversed and the axes definition will remain as in the standard case

Linking Segments

It is not necessary to specify the interconnections of segments in a kinematic model.

Any number of segments may be independently defined using separate sets of three or more markers.

However, in many models, especially those which use the kinetic functions, it is necessary for segments to be linked in specified ways, corresponding to the actual physical joints between the body parts they represent.

A hierarchy of segments begins with a root segment, from which chains of other segments branch out.

- Pelvis – Root segment
- Thigh – Child segment to the Pelvis
- Shank – Child segment to the Thigh
- Foot – Child segment to the Shank

Linking Segments – Types of joints

In some cases it may be desirable to limit the degrees of freedom between the parent and the child, but this is controlled entirely by the kinematic part of the segment definition (the choice of origin and defining lines) not by the hierarchy.

Free Joint

- Allows for six degree of freedom translation and rotation between two segments in space

- A joint chosen to be a free joint will allow the calculation of range of motion in all three segment axis components and should therefore be chosen based on its range of motion

- Each segment must be defined with three independent markers or so that there is no common axis shared by the segments surrounding a free joint

Linking Segments – Types of Joints

Ball and socket Joint

- Allows for rotations about all three axes to be calculated but does not allow translation at the joint

- Segments surrounding a ball and socket joint can share a point in their respective segment definitions

- The point shared between the segment definitions should be the joint centre about which the angle rotations will be calculated

Hinge Joint

- Allows for only one axis of rotation between two segments

Segments surrounding a hinge joint must share an axis in their respective segment definitions

- The shared axis will be the axis of rotation for the hinge and therefore the joint centre about which this rotation is calculated

- Hinge joint calculations are usually done when flexion-extension is the only quantifiable angle measurement for a joint e.g Elbow joint

Linking Segments - Syntax

```
SegmentName=[Origin,(DL1),(DL2),tolkin, anti-flip, parent, Attachement]
```

Parent – Name of the parent segment this segment attaches to
Attachment– connection point in the child, defined in global space

You can also add a parent segment and attachment point to an existing segment definition with the following syntax:

```
SegmentName=[SegmentName, parent, Attachment]
```

Segment Manipulation

ATTITUDE

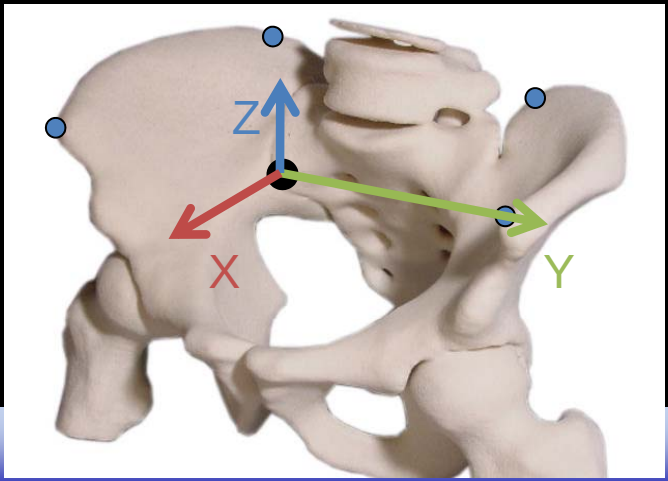
Used to move a local axes set while maintaining the axes orientation

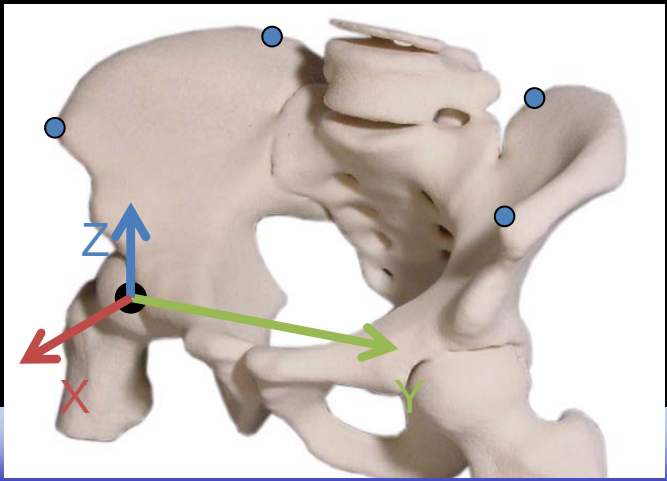
ATTITUDE (segment_name) Moves the orientation axes to the global origin

Now, a new segment origin can be defined while using the same “attitude” (orientation) of the original segment axes

New_Segment = New origin + ATTITUDE(Old_Segment)

NewPelvis = HJC + ATTITUDE(Pelvis)





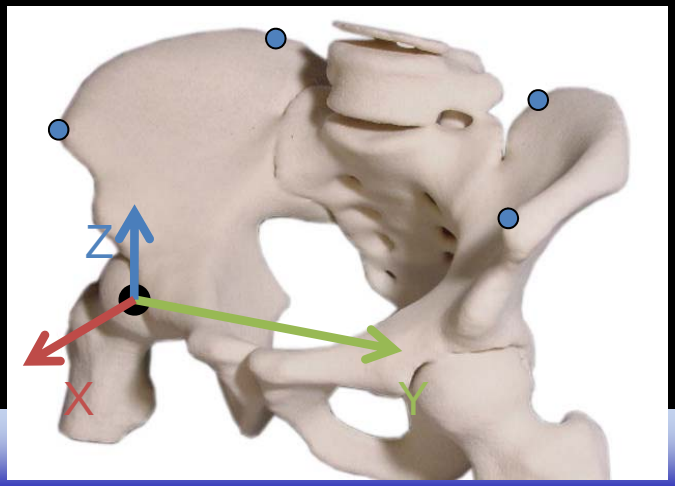
Rotate (ROT)

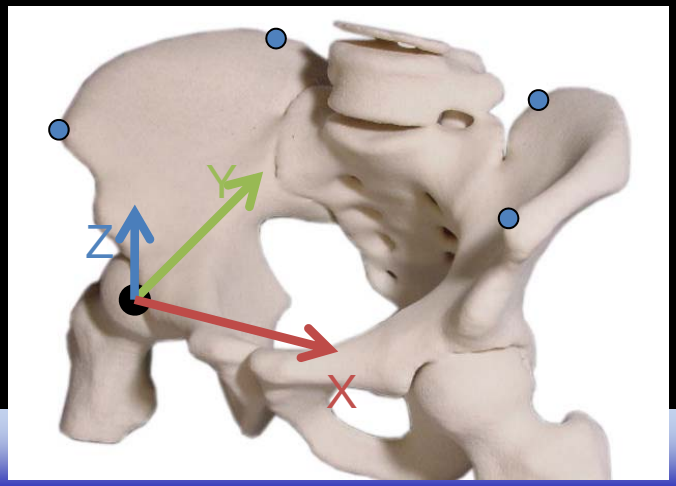
Used to rotate a segment about an axis by x number of degrees

The angle of rotation can be stored as a parameter in the MP file or calculated from the Static trial.

Segmetn = ROT(segment, axis of rotation, number of degrees)

Pelvis = ROT(Pelvis,3(Pelvis),90)





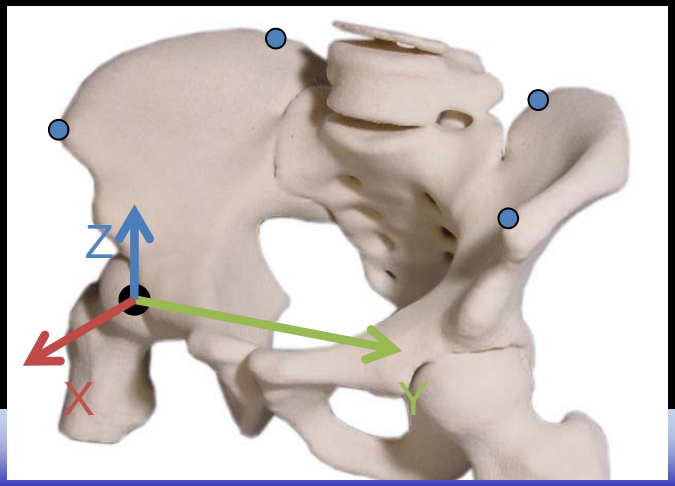
ALIGN

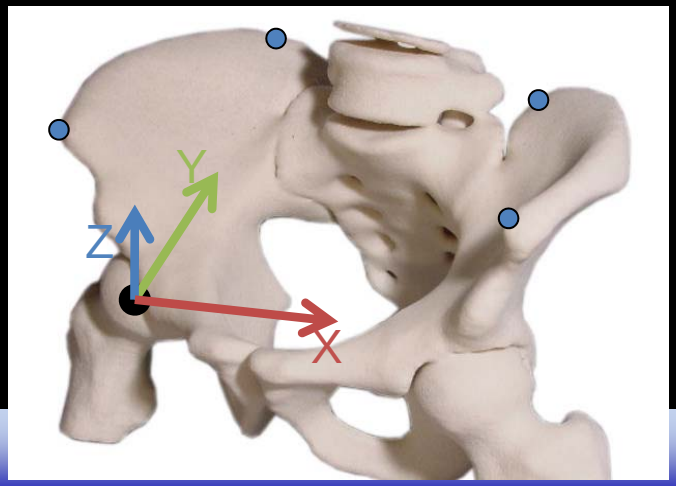
Used to align segment with axis and point projections

SegmentName = ALIGN(SegmentName, component1, component2, point)

Rotates the Segment axes about the specified “component1” axes to align the projection of the “component2” axis of the segment with the point specified, in the plane perpendicular to the component1 axis.

Pelvis = ALIGN(Pelvis,3,2,RPSI





3D angles

Absolute Angles

Absolute Angles – rotation of a local segment coordinate system with respect to the global reference frame

AngleName = <Segment name, token>
OUTPUT (AngleName)

For Example: GlobalFoot = <Rfoot,zxy>

- An angle is created called “GlobalFoot”.
- The angle is made up of 3 components.
- The token, zxy specifies:
  - GlobalFoot(1) -> Z of the segment rotated with respect to Z global
  - GlobalFoot(2) -> X of the segment rotated with respect to X global
  - GlobalFoot(3) -> Y of the segment rotated with respect to Y global

Relative Angles

Relative Angles – rotation of a local segment coordinate system with respect to another local segment coordinate system that is held constant

RelativeAngle = <ChildSegment, ParentSegment, token>
OUTPUT (RelativeAngle)

For Example: RkneeAngle = <Rshank, Rthigh, zxy>

- An angle is created called “RKneeAngle”
- The angle is made up of 3 components.
- The token, zxy specifies:
  - RKneeAngle(1) -> Rotation from the Rthigh to the Rshank around the z axis of the Rthigh
  - RKneeAngle(2) -> Rotation from the first intermediate axis system to the Rshank around the x axis of the Rthigh
  - RKneeAngle(3) -> Rotation from the second intermediate axis system to the Rshank around the z axis of the Rthigh

Note - For a given rotation between two segments, there are twelve possible sets of Euler angles. BodyLanguage specifically performs a "fixed axis" calculation where the three angles are calculated in the given order around the fixed axes of the parent segment.

Assignment

1. Create virtual points using the NORM and PERP functions
2. Update the segment definition for the L/R Thigh segments using the CHORD function to find the knee joint centers.
3. Add a parent segment definition to the Thigh and Tibia segments
4. Create a global angle for the Pelvis segment
5. Create local Hip and Knee angles. Be sure to add them to the MKR file to graph the outputs in the BodyBuilder workspace.