BodyBuilder session 6

Special Functions and Intro to Kinetics

Course Objectives

1. *Review of angle introduction*
2. *Advanced Angles*
3. Special Functions
    - LAST
    - EXIST
    - EXISTATALL
    - EXISTALWAYS
    - AVERAGE
    - MIN/MAX/STDDEV
    - FIELD/SAMPLE/FIRSTSAMPLE/LASTSAMPLE
4. *Working along the time line in BodyBuilder*
5. *Introduction to Kinetic Analysis*
    - *Creating Hierarchies*
    - Integration of inertial characteristics in segment definitions
        - Direct method vs. Anthropometry table

Advanced Angles

Summary of what has been done in session 5:

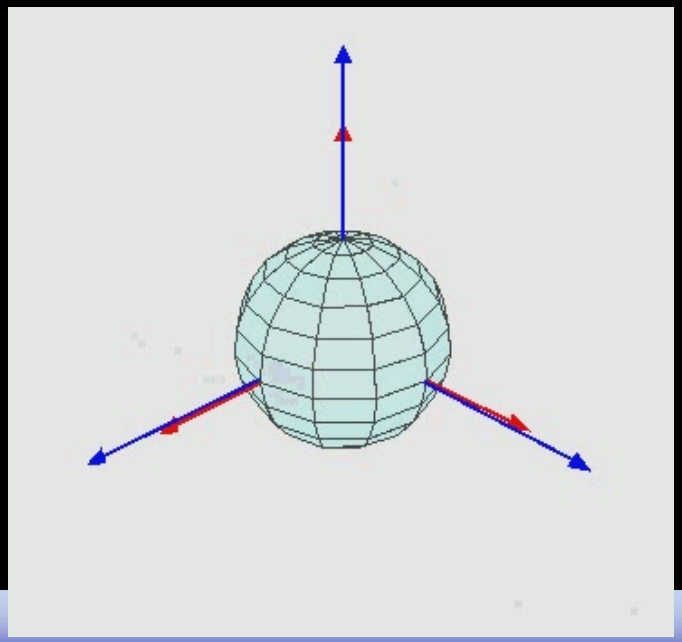
1. Introduction to the Euler Angles technique
   - Assuming some initial conditions, a composition of three rotations can be used to reach any target frame from the reference frame. The value of the rotations are the Euler Angles
2. Fixed VS Floating Axes convention
3. Absolute Angles
   - 3D angles calculated with respect to the global Vicon coordinate system
4. Relative Angles
   - 3D angles between two local coordinate systems in space (i.e. joint angles)

BodyLanguage and 3D angles

There are many different ways of describing the angular position of a rigid body in space. Among those, the most commonly used are

- Euler Angles
- Rotation Matrix
- Helical Axis
- Quaternion

BodyLanguage defaults to the ‘Euler Angles’ technique in order to specify the orientation of a rigid body (i.e. a 3D coordinate system) in space.
Assuming some initial conditions, a composition of three rotations can be used to reach any target frame from the reference frame. The value of the rotations are the Euler Angles.



Fixed VS Floating

The three Euler rotations can be carried out in two different ways:

1. Rotation of the moving frame around the fixed axes of the parent
   - **FIXED AXES CONVENTION**

2. Rotation of the moving frame around the moving axes of the intermediate coordinate systems obtained after each single rotation
   - **MOVING AXES CONVENTION**

BodyLanguage defaults to the FIXED AXES CONVENTION

From FIXED to FLOATING

As already mentioned, BodyLanguage defaults to the FIXED frame convention when calculating a set of 3D angles.
You give the child segment first, then the parent then an order of rotations:

RKneeAngles = <RShank,RThigh,yxz>

It is very common in biomechanical modeling to actually calculate a joint angle using the FLOATING axes convention.
Mathematically, switching between the FIXED and the FLOATING convention involves inverting the order of the rotations.

RKneeAngles = <RShank,RThigh,yxz>

The above example has two different interpretations:

1. RKneeAngles are the angles between the shank and the thigh, obtained using the **yxz** sequence around the fixed axes of the parent

   OR

2. RKneeAngles are the angles between the shank and the thigh, obtained using the **zxy** sequence around the moving axes of the intermediate coordinate systems generated after each rotation

From FIXED to FLOATING cont.

Also valid in mathematical terms is the following rule:

The same result from an Euler angles calculation could be obtained by

- Swapping the segments
- Inverting the order of the rotations
- Negating the resulting angles

Example:

Angle = <child,parent,xyz> = -<parent,child,zyx>

Absolute Angles

A set of Euler angles is defined absolute angles when the orientation of a segment with respect to the global Vicon coordinate system is calculated:

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

Example

GlobalFoot = <Rfoot,zxy>
equivalent to
Global = [{1,0,0},{0,0,1},xyz]
GlobalFoot = <Rfoot,Global,zxy>

- An angle is created called “GlobalFoot”.
- The token, zxy specifies:
  - GlobalFoot(1) -> Rotation of the Global to the Rfoot segment around the Z axis of the Global
  - GlobalFoot(2) -> Rotation of the first intermediate axis system to the Rfoot segment around the X axis of the Global
  - GlobalFoot(3) -> Rotation of the second intermediate axis system ti the Rfoot segment around the Y axis of the 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.

Euler Angles and BodyLanguage

Example:

1. Pelvis Segment
    1. Y medio-lateral
    2. X anterior-posterior
    3. Z vertical
2. Thigh Segment
    1. Y medio-lateral
    2. X anterior-posterior
    3. Z vertical

As a general rule, the sequence of Euler rotation is

1. Rotation on the sagittal plane (around the medio-lateral axis)
2. Rotation on the frontal plane (around the anterior-posterior axis)
3. Rotation on the transverse plane (around the vertical axis)

BodyLanguage defaults to the FIXED axes convention:

Angle = <Thigh, Pelvis, YXZ>

The above returns the angles between the Thigh and the Pelvis around the FIXED axes of the Pelvis.

Euler Angles and BodyLanguage cont

Angle = <Thigh, Pelvis, YXZ>

As stated in the previous slides, in order to obtain the Euler angles in the FLOATING axes convention, it is necessary to invert the order of the rotations:

AngleFloat = <Thigh, Pelvis, ZXY>

The expression above returns the angles expressed in the FLOATING axes convention, but the order of the rotations is actually inverted to that originally planned. In order to perform the rotations in the order planned, we can now swap the segments, invert the order of the rotations and negate the result (see previous slide).

AngleFloat = -<Pelvis, Thigh, YXZ>

Advanced 3D Angles

User Defined rotation – User can specify the components of a rotation

UserDefinedAngle = <AngleA, AngleB, AngleC>

Manipulating rotation outputs – any component of a rotation can be specified

Angle = <AngleA, AngleB, AngleC>

1(Angle) or Angle(1) = AngleA	Angle(-1) = -AngleA
2(Angle) or Angle(2) = AngleB	Angle(-2) = -AngleB
3(Angle) or Angle(3) = AngleC	Angle(-3) = -AngleC

Special Functions

LAST() - used to recall the most recent arithmetic value of an object from the current or any previous field. If the object has never previously been defined, it remains undefined.

LAST(I)	I from the last occasion it was defined

Special Functions

EXIST() - used in a logical expression to establish whether or not an object (number, point, segment, or rotation) is defined in the current field.

EXIST(I) true if point I is defined in current field

EXISTATALL() - used in a logical expression to establish whether or not an object (number, point, segment, or rotation) is defined in at least one field during the trial. More than one object may be supplied to the function, in which case all such objects must exist for at least one frame during the trial.

EXISTATALL(I, J) true if both I and J are ever defined

EXISTALWAYS() - used to determine whether an object or objects have values for every frame in the trial.

EXISTALWAYS(I) True if point I exists in every frame in the trial

NOTE: All the arguments of the EXIST(), EXISTATALL(), EXISTALWAYS() need to be declared as Optional in order to avoid error messages at the execution of the code when not present.

Special Functions

Example

Running different parts of the model according to what markers are being used.

```
OptionalPoints(LPSI, RPSI, SACR)

IF EXISTATALL(LPSI, RPSI)
        SACR = (LPSI+RPSI)/2
ELSEIF EXISTATALL(SACR)
        {*do something else*}
ENDIF
```

AVERAGE() - applies an average to any object (number, point, segment, or rotation) across all frames of a trial.

- The AVERAGE function is evaluated in a different way from all other functions.

- Normally, each assigned expression is evaluated once per field through the full set of data to which the model is applied. However, any expression containing AVERAGE() function is evaluated only once for all fields.

- Therefore, AVERAGE can not be used inside an IF statement.

AVERAGE(numberA)	Average numberA over all fields
AVERAGE(pointI)	Average pointI over all fields
AVERAGE(segmentP)	Average segmentP over all fields
AVERAGE(rotationq)	Average rotationq over all fields

Special Functions

MIN(), MAX(), STDDEV() - functions calculate the minimum, maximum and standard deviation (n-1) values for scalar values (number, point, segment, or rotation) over the whole trial.

- These functions are not allowed inside an IF block.

MIN(PointI)	Minimum value of PointI for all frames of the trial
MAX(PointI)	Maximum value of PointI for all frames of the trial
STDDEV (PointI)	Standard Deviation of PointI for all frames of the trial (n-1)

Working along the time line in BodyBuilder

Special Functions

[ ] - is a special "field offset" post-fix function used to find the value of an expression in any field, relative to the current one.

The “field offset" function has many uses, including the creation of special filters.

Point[-1]	Point in previous field
Point[5]	Point in the field 5 prior to the current field

SAMPLE, **FIRSTSAMPLE** and **LASTSAMPLE** - special predefined values that can be used in the script to provide the current, first and last sample numbers in the trial that is being processed.

SAMPLE	Provides the current sample number
FIRSTSAMPLE	Provides the first sample number of the trial
LASTSAMPLE	Provides the last sample number of the trial

Field Offset Operator - Example

EXAMPLE

Need to create a fixed segment in space using markers from the first sample when all of them are visible.

1. Identify the frames when all the markers participating to the segment definition are visible
2. Identify the first frame when all the markers participating to the segment definition are visible
3. Use the field offset operator to keep the value of the points always at the value of the same points at the frame identified at 2.

```
IF EXIST(LASI) AND EXIST(RASI) AND EXIST(LPSI)
      sCount = Sample              {* sCount is a vector and will be populated with all the frame
                                   numbers when the needed points are visible in the trial *}
ENDIF
ref = MIN(sCount)                          {* the minimum value of the sCount vector represents also the
                                           first frame where the needed points are all present *}
{* now create virtual points that always have the same value, that is the value of the needed points at the first
      frame where all of them are present *}

a=LASI[(ref - Sample)]                     {* if for example the first valid frame is frame 34 (i.e. ref), and
                                           we are at frame 18, then ref-Sample=34-18=16 *}
                                   {* LASI[16] means that we want to know the coordinates of
                                   LASI at 16 frames ahead of the current, which is 18+16=34 -
                                   which is the first valid frame *}
b=RASI[(ref - Sample)]
c=LPSI[(ref - Sample)]

fixed = [(a+b)/2,a-b,a-c,yzx]              {* fixed segment creation based on the first frame where LASI,
                                           RASI, LPSI are all visible *}
```

Introduction to Kinetic Analysis

Introduction to Kinetic Analysis

Kinetics – The study of forces which cause the motion of objects.

Kinetic Outputs in BodyBuilder:

1. **Joint Forces**
    a. Net Joint Forces expressed in the local reference system of each rigid body segment
    b. Units: [N/Kg]
2. **Joint Moments**
    a. Net Joint Moments estimated solving the equations of motion for the segments of the lower limbs – excluding the Pelvis (Ramakrishnan et al, 1987)
    b. Anthropometric Measurements from published tables or entered manually
    c. External Forces' convention is used: a GRF that would cause extension produces a positive extension moment
    d. Net Joint Moments ordered as Flexion, Abduction, Rotation
    e. Units: [Nmm/Kg]
3. **Joint Powers**
    a. Scalar product between Joint Moments and Joint Angular Velocities
    b. Joint Powers can be expressed as a scalar or as three components
    c. Units: [W/Kg]

NOTE - For more complete definitions of the above, see the manual, BodyBuilder_v1_2.pdf

Introduction to Kinetic Analysis

Kinetic Considerations

1.If Force Platforms (FPs) are present, the Ground Reaction Forces (GRFs) are added automatically to the processing workflow

2.Joint kinetics are calculated using the inverse dynamics procedure

3.In order to apply the inverse dynamics procedure, a segment hierarchy must be defined

Linking Segments

For those models using the kinetic functions, it is necessary to link the segments in a kinetic chain, 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

Introduction to Kinetic Analysis – Creating Hierarchies review

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]

Integration of inertial characteristics in segment definition – direct method

- An expression of inertial characteristics can be made in the same expression as the basic kinematic and hierarchical definitions, or added as an extension to a previously defined segment.

- The mass properties must the last part of the definition, whether the whole definition is made in one expression, or in parts.

- The syntax for extending a previously defined segment is:

```
segmentS=[segmentS,SegmentMass,CentreOfMassPoint,Inertia]
```

  - segmentS - is the segment name

  - SegmentMass - is a scalar quantity expressing the mass of the segment (in kg)

  - CentreOfMassPoint - is the location of the centre of mass of the segment, in the
  - local co-ordinate system of that segment

  - Inertia - is defined using three terms, like a point, which correspond to the
  - components of the moment of inertia

Integration of inertial characteristics in segment definition – Anthropometry table

The inertial properties can be defined using a table of anthropometric data based on standard values for the mass and inertial properties of various body segments.

Each entry in the table must include a name and four numbers. The numbers are:

- The segment mass as a proportion of total body mass
- The location of the centre of mass as a proportion of the length of the pricipal (long) axis of the segment
- Transverse radius of gyration around the centre of mass
- Longitudinal radius of gyration around the centre of mass
    - Note 1 – The radii of gyration are given in proportion of segment length
    - Note 2 – Origin of segment assumed to be distal end
    - Note 3 – First axes assumed to be along length of segment
    - Note 4 - The longitudinal radius of gyration is used for the inertia calculations around the principal (long) axis, and the transverse value, around the other two axes.
    - Note 5 – Bodymass must be given somewhere in the model or MP file as
        - Bodymass = # (Kg)

Integration of inertial characteristics in segment definition – Anthropometry table syntax

The table must begin with the word AnthropometricData and end with the word EndAnthropometricData

```
AnthropometricData
DefaultFemur 0.1 0.567 0.323 0
DefaultTibia 0.0465 0.567 0.302 0
DefaultFoot 0.0145 0.5 0.475 0
EndAnthropometricData
```

Integration of inertial characteristics in segment definition – Anthropometry table syntax

The segment length defaults to the distance from the segment origin, to the attachment point on the parent segment, a distance which may vary slightly from frame to frame because of skin movement, measurement noise, or inappropriate joint centre estimation, but which will in most cases be nearly constant.

A fixed length may be defined using a value from the parameter file:

segmentS=[segmentS,EntryName,<Segment Length>]

- EntryName reference to the table (e.g. DefaultFemur in the example above)
- SegmentLength - is an expression for the segment length (expressed in the units in which the original measurements were made, normally millimeters).

Session 6 - Assignment

1. Create both fixed and floating Euler angles for a few segments of the body and compare the outputs.

2. Add an IF statement to your code that will look to see weather there is a medial knee marker present on the right side, called RMKNE, and calculate the knee joint center using this marker and a lateral knee marker, RKNE. Then add an else statement that calculated the knee joint center using the CHORD function if the RMKNE is not present.

3. Put the example from slide 22 in our model to calculate a fixed segment in space and see what this looks like.

4. Add a parent to each of your segment definitions to create a hierarchy.

5. Create a anthro table for at least a few segments