A. More about the Biomechanical Man

The Biomechanical Man in MATLAB

The previous spreadsheet calculations produced 52 biomechanical landmarks including joint centres that were then used to construct the biomechanical man body model in MATLAB (Figure 1). The example code can be found on the accompanying CD (Appendix A). When `code characters` are used it denotes actual variables or functions used in the MATLAB algorithms.

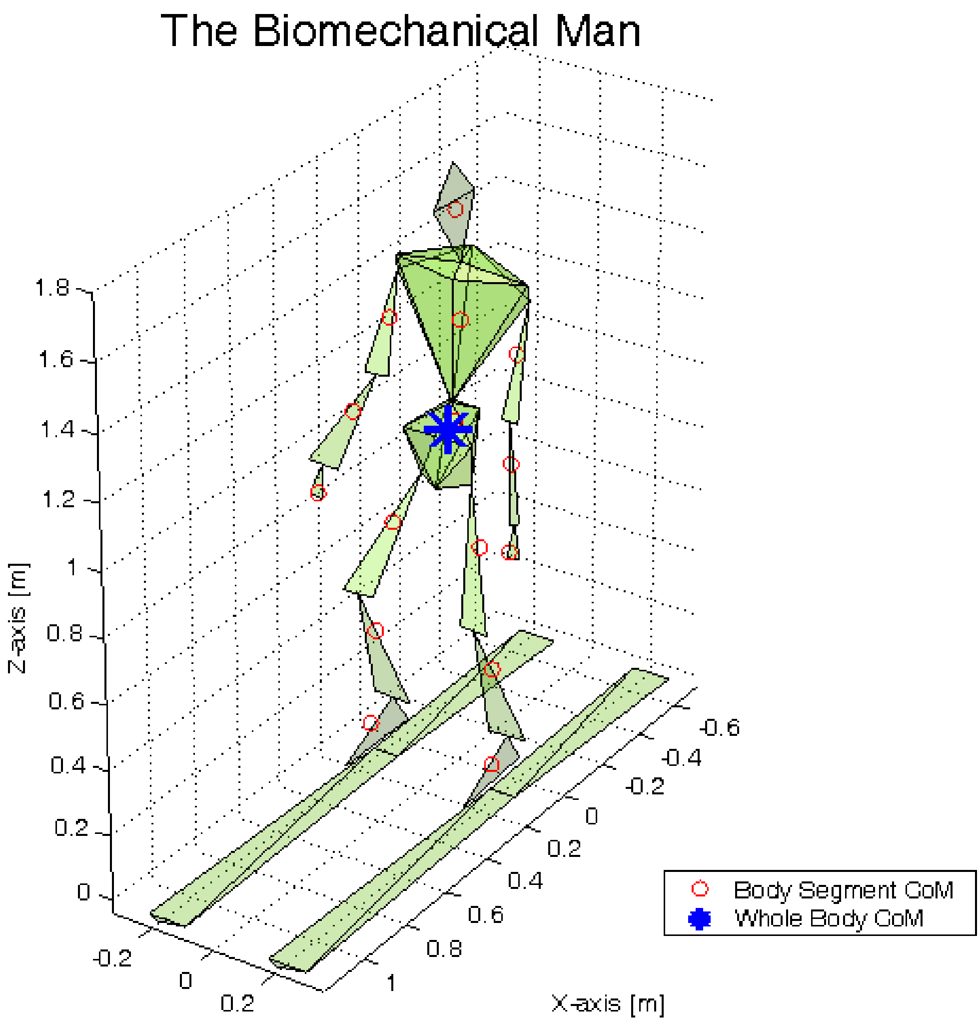

Figure 1: The Biomechanical Man: Visualisation of the athlete's measurements from the 3D anthropometric frame

In the example code the inputs are the coordinates of 52 biomechanical landmarks held in a variable called 'Measurement' in centimetres, and the subjects 'Mass' in kilograms. The outputs are three structures called; **`BSP`**, **`Indicies`** and **`Points_Index`**. The variable BSP contains information about each body segment, numbered from (1-15) in a multiple fields.

For example **`BSP(1).Name`** is the Thorax. **`BSP(2).Points`** is an array of points that define the visual representation of the Head segment in the local limb coordinate system based on the Measurements. **`BSP(3).SCS`** is the left arm segment coordinate system, a 9 element rotation matrix in row vector form that defines the rotation from the 3D anthropometric frame coordinate system to the arm coordinate system. The full list of **`BSP`** fields is provided in the referenced code. The following calculation steps are implemented in the MATLAB code on the accompanying CD to define the body model:

1. The base measurements are converted to metres from centimetres.
2. The points that make up the skis are defined based on actual measurements.
3. Information about each body segment is stored in the **`BSP`** structure including; the name of each bony landmark, the measurements, and which points connect together to form faces, when rendering.
4. The local segment coordinates system is defined ($\mathbf{R}_{GL}$). For example the right arm is defined in the following way: The local y-axis runs from the shoulder to elbow joint centres. The heading of the z-axis runs medial to lateral across the humeral epicondyles. The x-axis is perpendicular to the y-axis and the z-axis heading, and is found by a cross product. Finally to make the system orthogonal, the z-axis is found by the cross product of the x and y axes.
5. The characteristic segment length and segment origin are defined. In general this is defined as the inter-joint centre distance and the proximal joint centre respectively.
6. The biomechanical landmarks of arbitrary body segments (**`BSP(n)`**) measured in the global system (3D anthropometric frame) are transformed into the local coordinate system by subtraction of the origin and a rotation (**Equation 1**).

Equation 1: $$\mathrm{BSP(n).Points} = \mathrm{BSP(n).}\mathbf{R}_{GL} * (\mathrm{BSP(n).Points} - \mathrm{BSP(n).Origin})$$

7. The body segment masses are defined by multiplying the fractional body segment masses by the total body mass. The head and feet are adjusted for the additional mass of the athlete’s helmet and skis. The body segment mass fractions are defined in Dumas’s paper (Dumas et al., 2007).
8. The location of the segment CoM is defined by scaling the parameters suggested by Dumas by the athlete’s characteristic limb length. This is defined in the local limb coordinate system by adding an additional point to the variable **`BSP(n).Points.`**
9. The segment inertia tensor (a three-by-three matrix) is defined in the local limb coordinate system by scaling the parameters suggested by Dumas by the athlete’s characteristic limb length and segment mass. This information is stored in a new field called: **`BSP(n).Tensor`**.

In order to prepare for reconstruction and rendering of the body model (Figure 1), the following steps are then implemented in the code:

1. Define the order in which to reconstruct the body segments and the proximal joint centre to which each distal segment is attached. This is done in a variable called **`Joint_by_Name`** in the referenced code. The order implemented is proximal to distal starting at the C7 joint centre.
2. The points that make up the body segments are compiled into a single n-by-three matrix called **`Global_Points`** that defines the X, Y and Z location of each point.
3. Define the indices that point to the location of specific points within **`Global_Points`** matrix such as: **`Com_Index`**, a vector whose elements point to the individual body segment CoM locations. **`Joint_Index`**, a vector whose elements point toward the proximal joint centres to which the body segments must be attached. **`Points_Index,`** a vector whose elements define the locations of the points representing the body segments. And **`Face_Index`**, a vector whose elements define the locations of the vertices of each triangular face within the **`Global_Points`** matrix.

Reconstruction of the body model is then possible. If data were available from the fusion motion capture algorithms then the body model could be animated. In the example code on the accompanying CD and described below, the position of the athlete in the 3D anthropometric frame is reconstructed, but the process would not change if dynamic data were available (Figure 1). In the example code, to reconstruct the body model the follow steps are implemented:

1. The global origin, or cervical joint centre, of the athlete is supplied externally by a variable called **`Global_Origin`**. In this case the global origin of the athlete in metres is X=0, Y=0 and Z=1.5.
2. The rotation from local limb coordinate frame to the global coordinate frame is supplied by a variable called **`Rotate_LG`**. In this case the rotation is defined by the matrix transpose of the rotation matrix that defines the local limb coordinate system relative to the global coordinate system ($\mathbf{R}_{GL}$).
3. The **`Global_Points`** matrix is compiled by a 'for loop' of code that cycles through each body segment in a predefined order. The global points of an arbitrary segment are the points defined in the local coordinate system rotated to the global coordinate system with the global location of the proximal joint centre added.
4. Extract the location of the body segment CoMs from the **`Global_Points`** matrix, using the predefined **`Com_Index`**, scale, using the predefined body segment mass fractions, and sum (**Equation 2** and **Equation 3**).

Equation 2: $$Com\_Points = Global\_Points(Com\_Index,:)$$

Equation 3: $$Com\_Whole\_Body = \sum_{N}^{1} Com\_Points_n * Mass\_Fraction_n$$