Body Segment Inertial Parameters

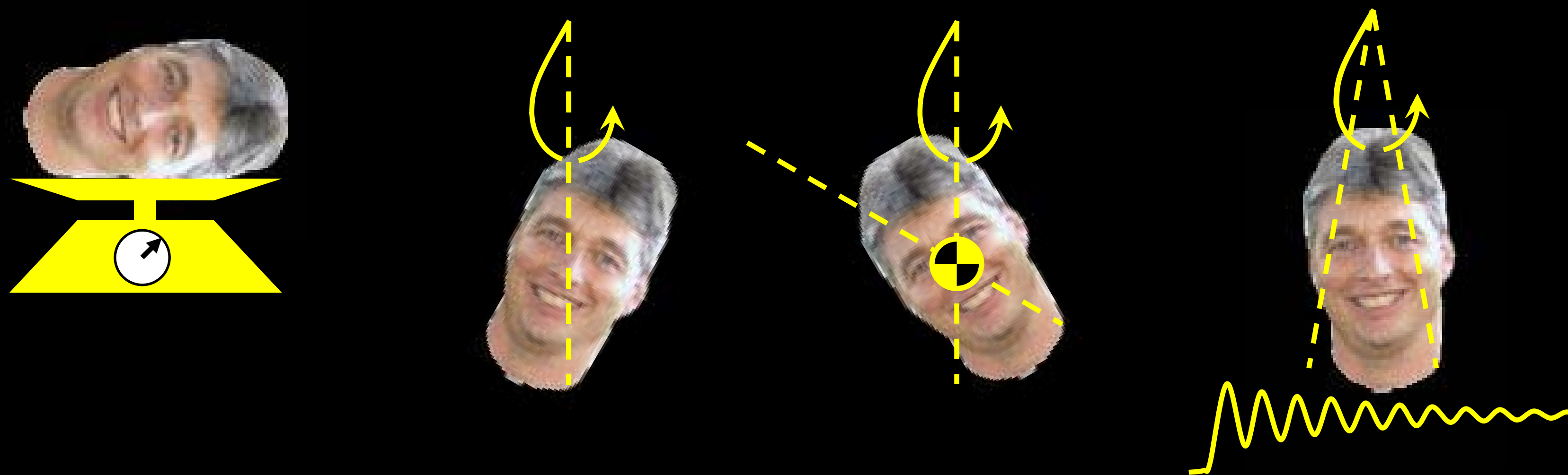
Jeff Reinbolt

BioE215

April 27, 2007

Question of the Day

What is the mass, center of mass, and moment of inertia for Paul's head?



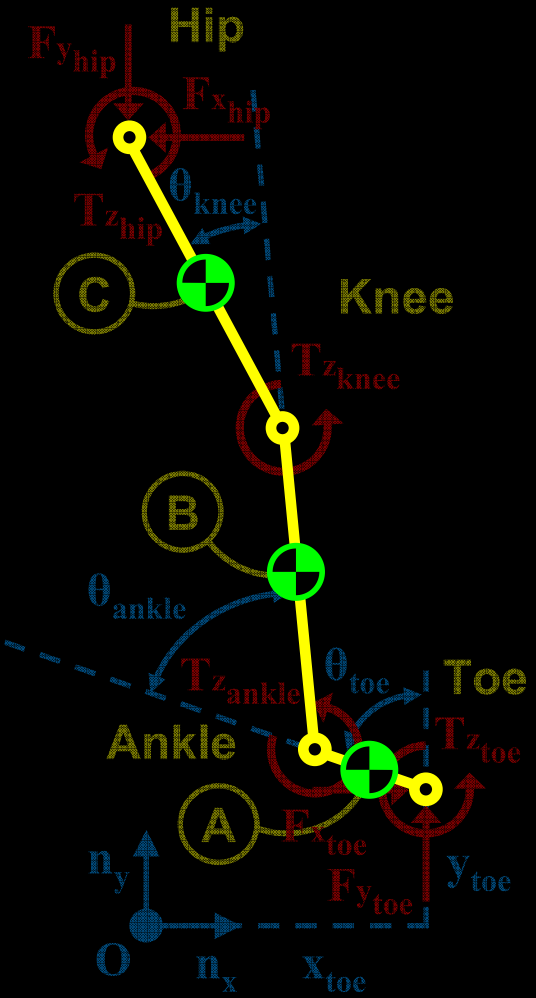
$$m_{head} = P_{head} m_{total}$$
$$m_{head} = 0.081 \times 86.0$$
$$m_{head} = 6.97 \text{ kg}$$

$$r_{head} = R_{proximal(head)} l_{head}$$
$$r_{head} = 1.00 \times 0.229$$
$$r_{head} = 0.229 \text{ m}$$

$$I_{zz}^{head/cm} = m_{head} (K_{cm(head)} l_{head})^2$$
$$I_{zz}^{head/cm} = 6.97 \times (0.495 \times 0.299)^2$$
$$I_{zz}^{head/cm} = 0.089 \text{ kg} \cdot \text{m}^2$$

Why Do We Care?

$F = ma$



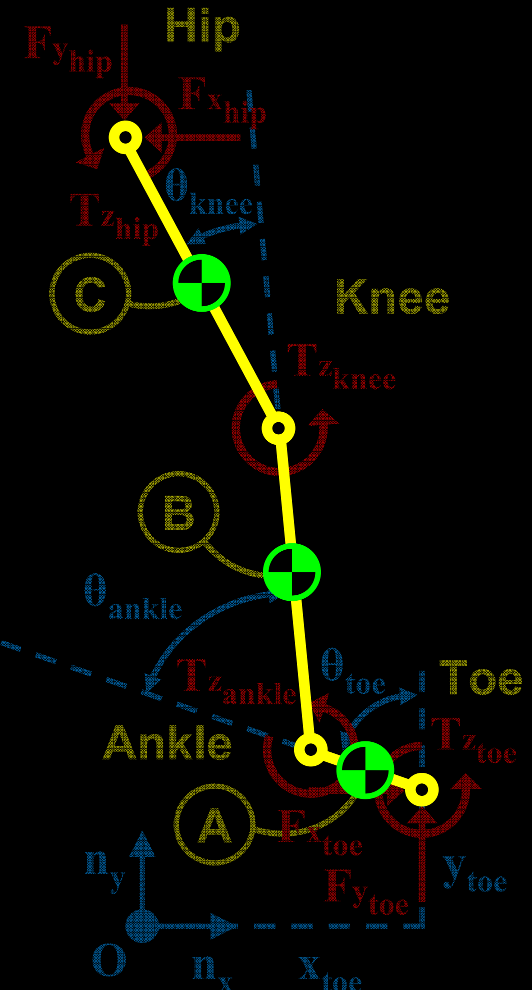
Inertial Properties

- mass
- center of mass
- moment of inertia

Why Do We Care?

$F = ma$

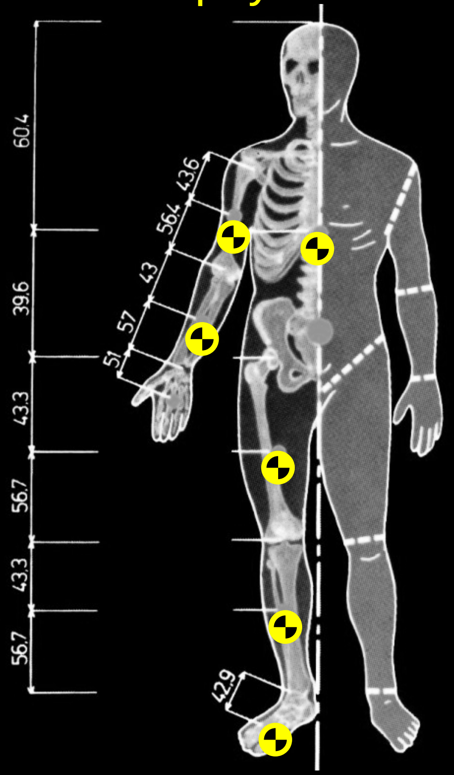
```
Fr1 = comA * mA * sin(q3) * u3^2 + mB * (IA * sin(q3) * u3^2 + comB * sin(q3 + q4) * (u3 + u4)^2) + mC * (IA * sin(q3) * u3^2 + IB * sin(q3 + q4) * (u3 + u4)^2 + comC * sin(q3 + q4 + q5) * (u3 + u4 + u5)^2) + (mA + mB + mC) * u1' - F1 - comC * mC * cos(q3 + q4 + q5) * u5' - (comB * mB * cos(q3 + q4) + mC * (IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u4' - (comA * mA * cos(q3) + mB * (IA * cos(q3) + comB * cos(q3 + q4)) + mC * (IA * cos(q3) + IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u3'

Fr2 = g * mA + g * mB + g * mC + (mA + mB + mC) * u2' - F2 - comA * mA * cos(q3) * u3^2 - mB * (IA * cos(q3) * u3^2 + comB * cos(q3 + q4) * (u3 + u4)^2) - mC * (IA * cos(q3) * u3^2 + IB * cos(q3 + q4) * (u3 + u4)^2 + comC * cos(q3 + q4 + q5) * (u3 + u4 + u5)^2) - comC * mC * sin(q3 + q4 + q5) * u5' - (comB * mB * sin(q3 + q4) + mC * (IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u4' - (comA * mA * sin(q3) + mB * (IA * sin(q3) + comB * sin(q3 + q4)) + mC * (IA * sin(q3) + IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u3'

Tr3 = comB * IA * mB * sin(q4) * (u3^2 - (u3 + u4)^2) + mC * (IA * IB * sin(q4) * u3^2 + comC * IA * sin(q4 + q5) * u3^2 + comC * IB * sin(q5) * (u3 + u4)^2 - IA * IB * sin(q4) * (u3 + u4)^2 - comC * IB * sin(q5) * (u3 + u4 + u5)^2 - comC * IA * sin(q4 + q5) * (u3 + u4 + u5)^2) + (IC + comC * mC * (comC + IB * cos(q5) + IA * cos(q4 + q5))) * u5' + (IB + IC + comB * mB * (comB + IA * cos(q4)) + mC * (comC^2 + IB^2 + IA * IB * cos(q4) + 2 * comC * IB * cos(q5) + comC * IA * cos(q4 + q5))) * u4' + (IA + IB + IC + mA * comA^2 + mB * (comB^2 + IA^2 + 2 * comB * IA * cos(q4)) + mC * (comC^2 + IA^2 + IB^2 + 2 * comC * IB * cos(q5) + 2 * IA * IB * cos(q4) + 2 * comC * IA * cos(q4 + q5))) * u3' + (IA * cos(q3) + IB * cos(q3 + q4) + IC * cos(q3 + q4 + q5)) * (comA * mA * sin(q3) * u3^2 + mB * (IA * sin(q3) * u3^2 + comB * sin(q3 + q4) * (u3 + u4)^2) + mC * (IA * sin(q3) * u3^2 + IB * sin(q3 + q4) * (u3 + u4)^2 + comC * sin(q3 + q4 + q5) * (u3 + u4 + u5)^2) + (mA + mB + mC) * u1' - F1 - comC * mC * cos(q3 + q4 + q5) * u5' - (comB * mB * cos(q3 + q4) + mC * (IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u4' - (comA * mA * cos(q3) + mB * (IA * cos(q3) + comB * cos(q3 + q4)) + mC * (IA * cos(q3) + IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u3') + (IA * sin(q3) + IB * sin(q3 + q4) + IC * sin(q3 + q4 + q5)) * (g * mA + g * mB + g * mC + (mA + mB + mC) * u2' - F2 - comA * mA * cos(q3) * u3^2 - mB * (IA * cos(q3) * u3^2 + comB * cos(q3 + q4) * (u3 + u4)^2) - mC * (IA * cos(q3) * u3^2 + IB * cos(q3 + q4) * (u3 + u4)^2 + comC * cos(q3 + q4 + q5) * (u3 + u4 + u5)^2) - comC * mC * sin(q3 + q4 + q5) * u5' - (comB * mB * sin(q3 + q4) + mC * (IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u4' - (comA * mA * sin(q3) + mB * (IA * sin(q3) + comB * sin(q3 + q4)) + mC * (IA * sin(q3) + IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u3') - T3 - comA * g * mA * sin(q3) - g * mB * (IA * sin(q3) + comB * sin(q3 + q4)) - g * mC * (IA * sin(q3) + IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5)) - (comA * mA * sin(q3) + mB * (IA * sin(q3) + comB * sin(q3 + q4)) + mC * (IA * sin(q3) + IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u2' - (comA * mA * cos(q3) + mB * (IA * cos(q3) + comB * cos(q3 + q4)) + mC * (IA * cos(q3) + IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u1'

T5 = T3 + comA * g * mA * sin(q3) + g * mB * (IA * sin(q3) + comB * sin(q3 + q4)) + g * mC * (IA * sin(q3) + IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5)) + comC * mC * (IA * sin(q4 + q5) * u3^2 + IB * sin(q5) * (u3 + u4)^2) + (IC + mC * comC^2) * u5' + (IC + comC * mC * (comC + IB * cos(q5))) * u4' + (IC + comC * mC * (comC + IB * cos(q5) + IA * cos(q4 + q5))) * u3' + (comA * mA * sin(q3) + mB * (IA * sin(q3) + comB * sin(q3 + q4)) + mC * (IA * sin(q3) + IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u2' + (comA * mA * cos(q3) + mB * (IA * cos(q3) + comB * cos(q3 + q4)) + mC * (IA * cos(q3) + IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u1' + IC * cos(q3 + q4 + q5) * (comA * mA * sin(q3) * u3^2 + mB * (IA * sin(q3) * u3^2 + comB * sin(q3 + q4) * (u3 + u4)^2) + mC * (IA * sin(q3) * u3^2 + IB * sin(q3 + q4) * (u3 + u4)^2 + comC * sin(q3 + q4 + q5) * (u3 + u4 + u5)^2) + (mA + mB + mC) * u1' - F1 - comC * mC * cos(q3 + q4 + q5) * u5' - (comB * mB * cos(q3 + q4) + mC * (IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u4' - (comA * mA * cos(q3) + mB * (IA * cos(q3) + comB * cos(q3 + q4)) + mC * (IA * cos(q3) + IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u3') + IC * sin(q3 + q4 + q5) * (g * mA + g * mB + g * mC + (mA + mB + mC) * u2' - F2 - comA * mA * cos(q3) * u3^2 - mB * (IA * cos(q3) * u3^2 + comB * cos(q3 + q4) * (u3 + u4)^2) - mC * (IA * cos(q3) * u3^2 + IB * cos(q3 + q4) * (u3 + u4)^2 + comC * cos(q3 + q4 + q5) * (u3 + u4 + u5)^2) - comC * mC * sin(q3 + q4 + q5) * u5' - (comB * mB * sin(q3 + q4) + mC * (IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u4' - (comA * mA * sin(q3) + mB * (IA * sin(q3) + comB * sin(q3 + q4)) + mC * (IA * sin(q3) + IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u3') - comC * g * mC * sin(q3 + q4 + q5) - comB * IA * mB * sin(q4) * (u3^2 - (u3 + u4)^2) - mC * (IA * IB * sin(q4) * u3^2 + comC * IA * sin(q4 + q5) * u3^2 + comC * IB * sin(q5) * (u3 + u4)^2 - IA * IB * sin(q4) * (u3 + u4)^2 - comC * IB * sin(q5) * (u3 + u4 + u5)^2 - comC * IA * sin(q4 + q5) * (u3 + u4 + u5)^2) - comC * mC * sin(q3 + q4 + q5) * u2' - comC * mC * cos(q3 + q4 + q5) * u1' - (IC + comC * mC * (comC + IB * cos(q5) + IA * cos(q4 + q5))) * u5' - (IB + IC + comB * mB * (comB + IA * cos(q4)) + mC * (comC^2 + IB^2 + IA * IB * cos(q4) + 2 * comC * IB * cos(q5) + comC * IA * cos(q4 + q5))) * u4' - (IA + IB + IC + mA * comA^2 + mB * (comB^2 + IA^2 + 2 * comB * IA * cos(q4)) + mC * (comC^2 + IA^2 + IB^2 + 2 * comC * IB * cos(q5) + 2 * IA * IB * cos(q4) + 2 * comC * IA * cos(q4 + q5))) * u3' - (IA * cos(q3) + IB * cos(q3 + q4) + IC * cos(q3 + q4 + q5)) * (comA * mA * sin(q3) * u3^2 + mB * (IA * sin(q3) * u3^2 + comB * sin(q3 + q4) * (u3 + u4)^2) + mC * (IA * sin(q3) * u3^2 + IB * sin(q3 + q4) * (u3 + u4)^2 + comC * sin(q3 + q4 + q5) * (u3 + u4 + u5)^2) + (mA + mB + mC) * u1' - F1 - comC * mC * cos(q3 + q4 + q5) * u5' - (comB * mB * cos(q3 + q4) + mC * (IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u4' - (comA * mA * cos(q3) + mB * (IA * cos(q3) + comB * cos(q3 + q4)) + mC * (IA * cos(q3) + IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u3') - (IA * sin(q3) + IB * sin(q3 + q4) + IC * sin(q3 + q4 + q5)) * (g * mA + g * mB + g * mC + (mA + mB + mC) * u2' - F2 - comA * mA * cos(q3) * u3^2 - mB * (IA * cos(q3) * u3^2 + comB * cos(q3 + q4) * (u3 + u4)^2) - mC * (IA * cos(q3) * u3^2 + IB * cos(q3 + q4) * (u3 + u4)^2 + comC * cos(q3 + q4 + q5) * (u3 + u4 + u5)^2) - comC * mC * sin(q3 + q4 + q5) * u5' - (comB * mB * sin(q3 + q4) + mC * (IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u4' - (comA * mA * sin(q3) + mB * (IA * sin(q3) + comB * sin(q3 + q4)) + mC * (IA * sin(q3) + IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u3')

T4 = T3 + comA * g * mA * sin(q3) + g * mB * (IA * sin(q3) + comB * sin(q3 + q4)) + g * mC * (IA * sin(q3) + IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5)) + comB * IA * mB * sin(q4) * u3^2 + mC * (IA * IB * sin(q4) * u3^2 + comC * IA * sin(q4 + q5) * u3^2 + comC * IB * sin(q5) * (u3 + u4)^2 - comC * IB * sin(q5) * (u3 + u4 + u5)^2) + (IC + comC * mC * (comC + IB * cos(q5))) * u5' + (IB + IC + mB * comB^2 + mC * (comC^2 + IB^2 + 2 * comC * IB * cos(q5))) * u4' + (comA * mA * sin(q3) + mB * (IA * sin(q3) + comB * sin(q3 + q4)) + mC * (IA * sin(q3) + IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u2' + (comA * mA * cos(q3) + mB * (IA * cos(q3) + comB * cos(q3 + q4)) + mC * (IA * cos(q3) + IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u1' + (IB + IC + comB * mB * (comB + IA * cos(q4)) + mC * (comC^2 + IB^2 + IA * IB * cos(q4) + 2 * comC * IB * cos(q5) + comC * IA * cos(q4 + q5))) * u3' + (IB * cos(q3 + q4) + IC * cos(q3 + q4 + q5)) * (comA * mA * sin(q3) * u3^2 + mB * (IA * sin(q3) * u3^2 + comB * sin(q3 + q4) * (u3 + u4)^2) + mC * (IA * sin(q3) * u3^2 + IB * sin(q3 + q4) * (u3 + u4)^2 + comC * sin(q3 + q4 + q5) * (u3 + u4 + u5)^2) + (mA + mB + mC) * u1' - F1 - comC * mC * cos(q3 + q4 + q5) * u5' - (comB * mB * cos(q3 + q4) + mC * (IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u4' - (comA * mA * cos(q3) + mB * (IA * cos(q3) + comB * cos(q3 + q4)) + mC * (IA * cos(q3) + IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u3') + (IB * sin(q3 + q4) + IC * sin(q3 + q4 + q5)) * (g * mA + g * mB + g * mC + (mA + mB + mC) * u2' - F2 - comA * mA * cos(q3) * u3^2 - mB * (IA * cos(q3) * u3^2 + comB * cos(q3 + q4) * (u3 + u4)^2) - mC * (IA * cos(q3) * u3^2 + IB * cos(q3 + q4) * (u3 + u4)^2 + comC * cos(q3 + q4 + q5) * (u3 + u4 + u5)^2) - comC * mC * sin(q3 + q4 + q5) * u5' - (comB * mB * sin(q3 + q4) + mC * (IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u4' - (comA * mA * sin(q3) + mB * (IA * sin(q3) + comB * sin(q3 + q4)) + mC * (IA * sin(q3) + IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u3') - comB * g * mB * sin(q3 + q4) - g * mC * (IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5)) - comB * IA * mB * sin(q4) * (u3^2 - (u3 + u4)^2) - mC * (IA * IB * sin(q4) * u3^2 + comC * IA * sin(q4 + q5) * u3^2 + comC * IB * sin(q5) * (u3 + u4)^2 - IA * IB * sin(q4) * (u3 + u4)^2 - comC * IB * sin(q5) * (u3 + u4 + u5)^2 - comC * IA * sin(q4 + q5) * (u3 + u4 + u5)^2) - (IC + comC * mC * (comC + IB * cos(q5) + IA * cos(q4 + q5))) * u5' - (comB * mB * sin(q3 + q4) + mC * (IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u2' - (comB * mB * cos(q3 + q4) + mC * (IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u1' - (IB + IC + comB * mB * (comB + IA * cos(q4)) + mC * (comC^2 + IB^2 + IA * IB * cos(q4) + 2 * comC * IB * cos(q5) + comC * IA * cos(q4 + q5))) * u4' - (IA + IB + IC + mA * comA^2 + mB * (comB^2 + IA^2 + 2 * comB * IA * cos(q4)) + mC * (comC^2 + IA^2 + IB^2 + 2 * comC * IB * cos(q5) + 2 * IA * IB * cos(q4) + 2 * comC * IA * cos(q4 + q5))) * u3' - (IA * cos(q3) + IB * cos(q3 + q4) + IC * cos(q3 + q4 + q5)) * (comA * mA * sin(q3) * u3^2 + mB * (IA * sin(q3) * u3^2 + comB * sin(q3 + q4) * (u3 + u4)^2) + mC * (IA * sin(q3) * u3^2 + IB * sin(q3 + q4) * (u3 + u4)^2 + comC * sin(q3 + q4 + q5) * (u3 + u4 + u5)^2) + (mA + mB + mC) * u1' - F1 - comC * mC * cos(q3 + q4 + q5) * u5' - (comB * mB * cos(q3 + q4) + mC * (IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u4' - (comA * mA * cos(q3) + mB * (IA * cos(q3) + comB * cos(q3 + q4)) + mC * (IA * cos(q3) + IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u3') - (IA * sin(q3) + IB * sin(q3 + q4) + IC * sin(q3 + q4 + q5)) * (g * mA + g * mB + g * mC + (mA + mB + mC) * u2' - F2 - comA * mA * cos(q3) * u3^2 - mB * (IA * cos(q3) * u3^2 + comB * cos(q3 + q4) * (u3 + u4)^2) - mC * (IA * cos(q3) * u3^2 + IB * cos(q3 + q4) * (u3 + u4)^2 + comC * cos(q3 + q4 + q5) * (u3 + u4 + u5)^2) - comC * mC * sin(q3 + q4 + q5) * u5' - (comB * mB * sin(q3 + q4) + mC * (IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u4' - (comA * mA * sin(q3) + mB * (IA * sin(q3) + comB * sin(q3 + q4)) + mC * (IA * sin(q3) + IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u3')
```

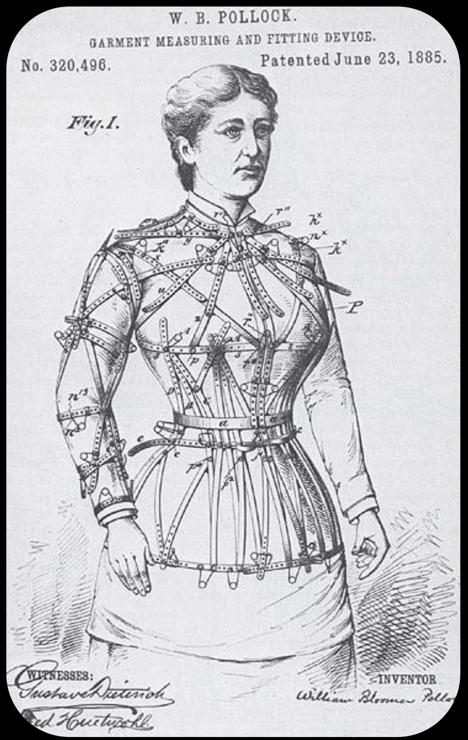


How Do We Determine Body Segment Parameters?

Anthropometry (Greek, *anthrōpos-* + *-metriā* = human measure)

- Discipline concerned with the measurement of the physical characteristics of humans

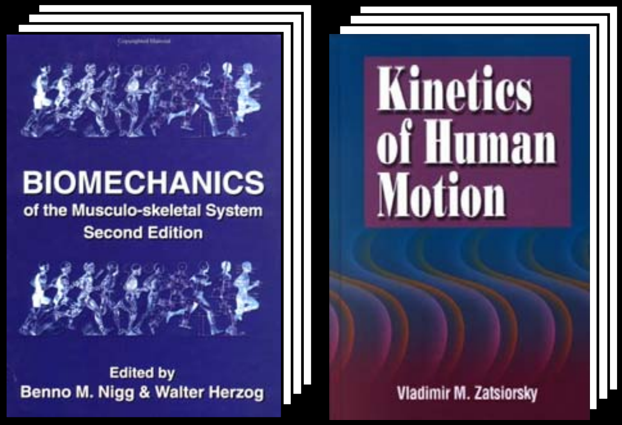




Biomechanists are mainly interested in the inertial properties of the body and its segments

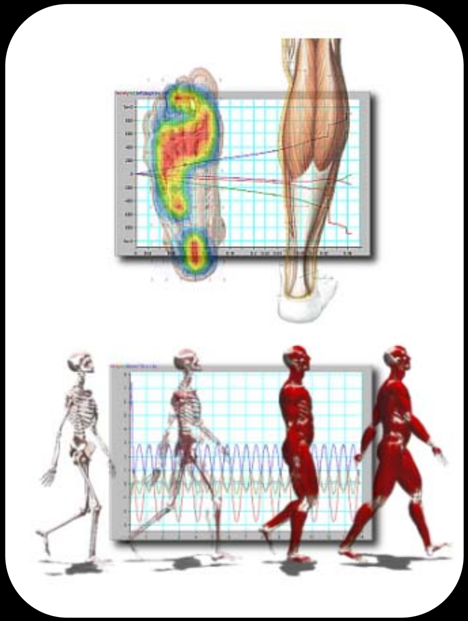
- May need segment lengths, circumferences, etc.

Disclaimer



This lecture is *NOT* meant to be a detailed account of the study of anthropometry or body segment parameters

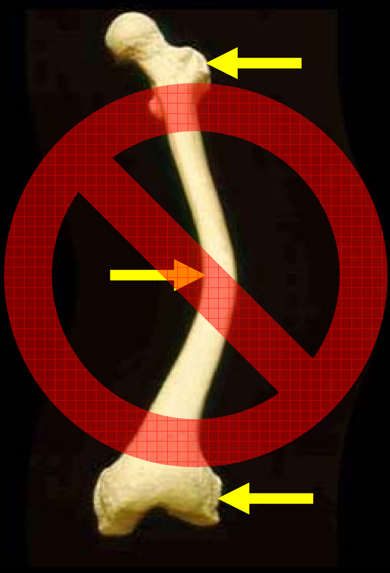
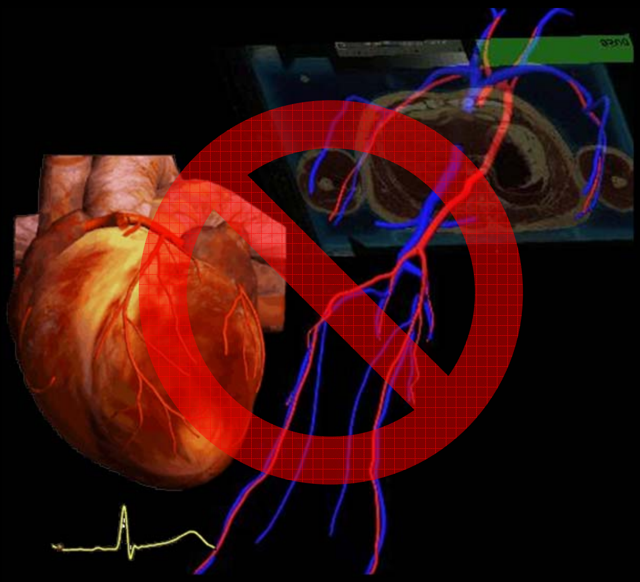
- Described elsewhere (Contini 1972, Drillis, Contini, and Bluestein 1964, Nigg and Herzog 1994, Zatsiorsky 2002)



The purpose of this lecture is to present background material on determining inertial properties to use in your simulations

Biomechanical Assumptions

- Body segments behave as rigid bodies during movement
  - Ignores the fact that bones bend, blood flows, and muscles contract


Biomechanical Assumptions

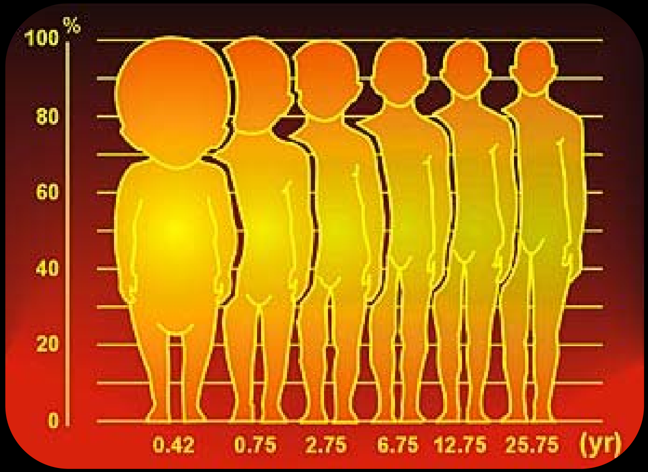
- Body segments behave as rigid bodies during movement
  - Ignores the fact that bones bend, blood flows, and muscles contract
- Single rigid body can be used to lump several body segments
  - Ignores the fact that hands, feet, and torso have several joints

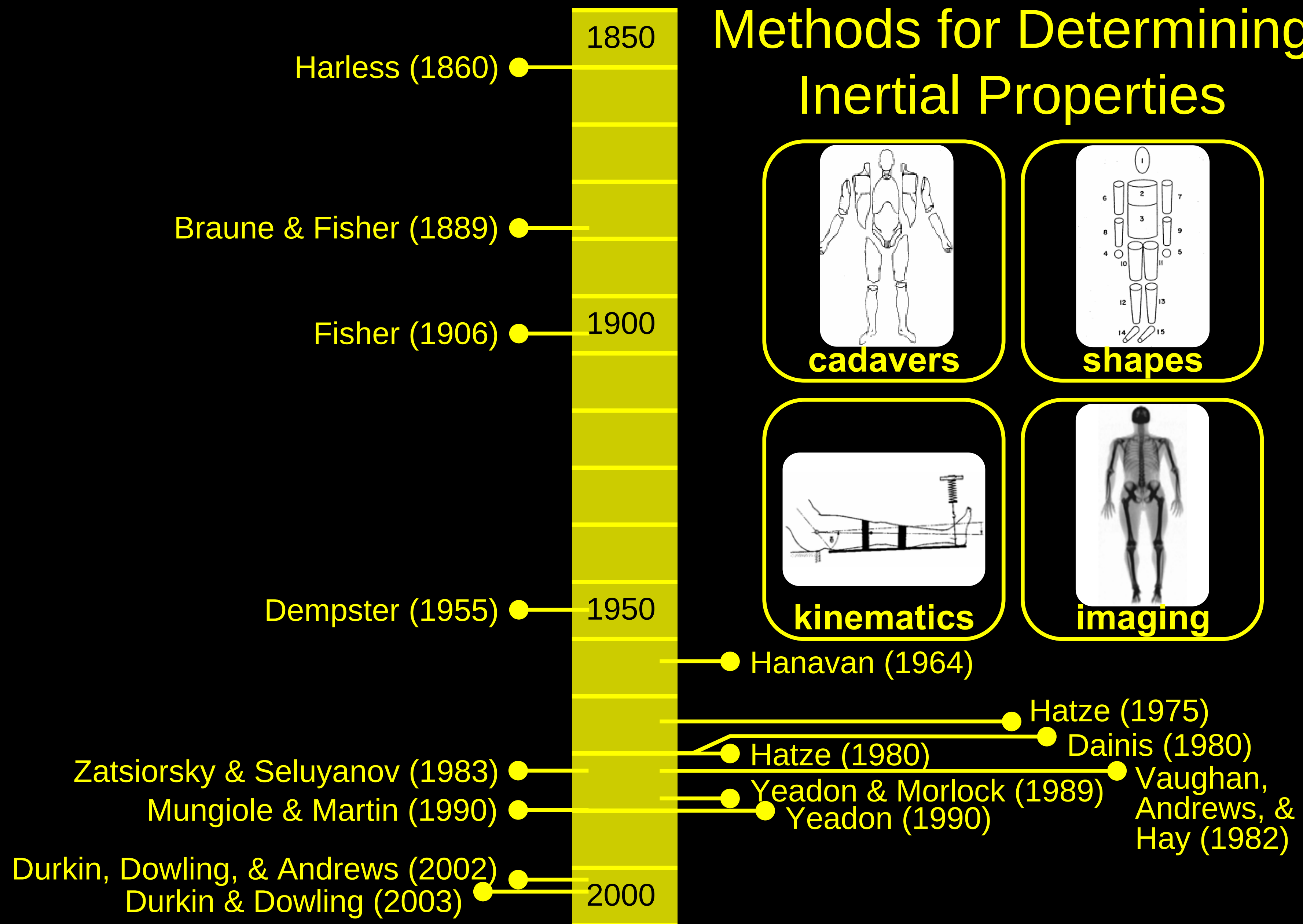
Biomechanical Assumptions

- Body segments behave as rigid bodies during movement
  - Ignores the fact that bones bend, blood flows, and muscles contract
- Single rigid body can be used to lump several body segments
  - Ignores the fact that hands, feet, and torso have several joints
- Rigid body assumptions simplify a complex musculoskeletal system
  - Eliminates the need to quantify mass distribution changes caused by tissue deformation and movement of bodily fluids

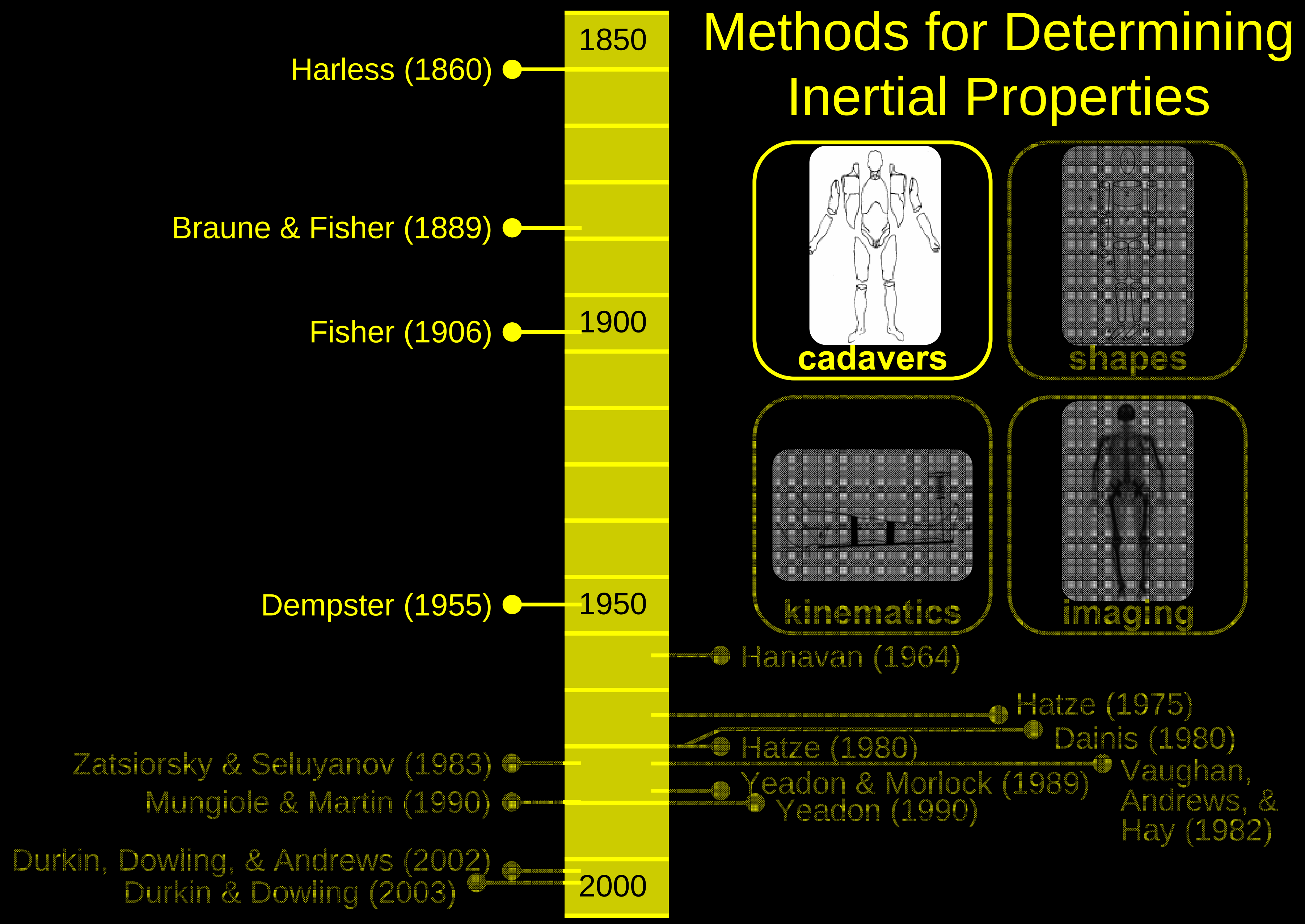
Biomechanical Assumptions

- Body segments behave as rigid bodies during movement
  - Ignores the fact that bones bend, blood flows, and muscles contract
- Single rigid body can be used to lump several body segments
  - Ignores the fact that hands, feet, and torso have several joints
- Rigid body assumptions simplify a complex musculoskeletal system
  - Eliminates the need to quantify mass distribution changes caused by tissue deformation and movement of bodily fluids
- Body segment mass distribution scales across “reasonable” sizes
  - Allows individual’s inertial properties to be estimated based on averages from a sample population



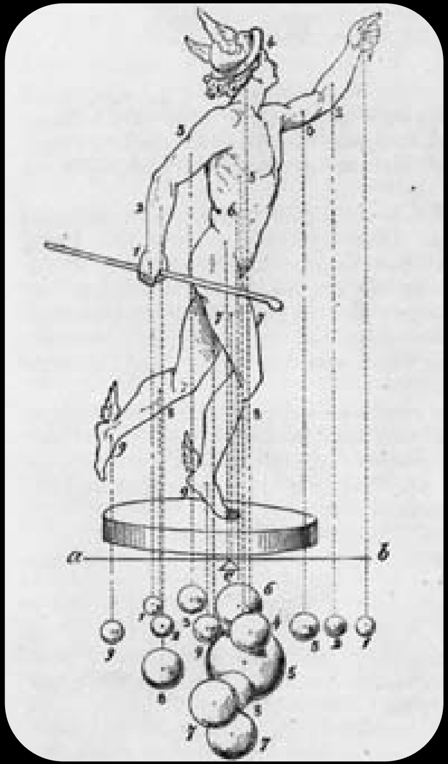


Methods for Determining Inertial Properties
1850
1900
1950
2000
Harless (1860)
Braune & Fisher (1889)
Fisher (1906)
Dempster (1955)
Hanavan (1964)
Hatze (1975)
Dainis (1980)
Hatze (1980)
Vaughan, Andrews, & Hay (1982)
Zatsiorsky & Seluyanov (1983)
Yeadon & Morlock (1989)
Yeadon (1990)
Mungiole & Martin (1990)
Durkin, Dowling, & Andrews (2002)
Durkin & Dowling (2003)
cadavers
shapes
kinematics
imaging


Methods for Determining Inertial Properties
1850
1900
1950
2000
Harless (1860)
Braune & Fisher (1889)
Fisher (1906)
Dempster (1955)
cadavers
shapes
kinematics
imaging
Hanavan (1964)
Hatze (1975)
Dainis (1980)
Hatze (1980)
Vaughan, Andrews, & Hay (1982)
Zatsiorsky & Seluyanov (1983)
Yeadon & Morlock (1989)
Yeadon (1990)
Mungiole & Martin (1990)
Durkin, Dowling, & Andrews (2002)
Durkin & Dowling (2003)

Cadaver Measurements

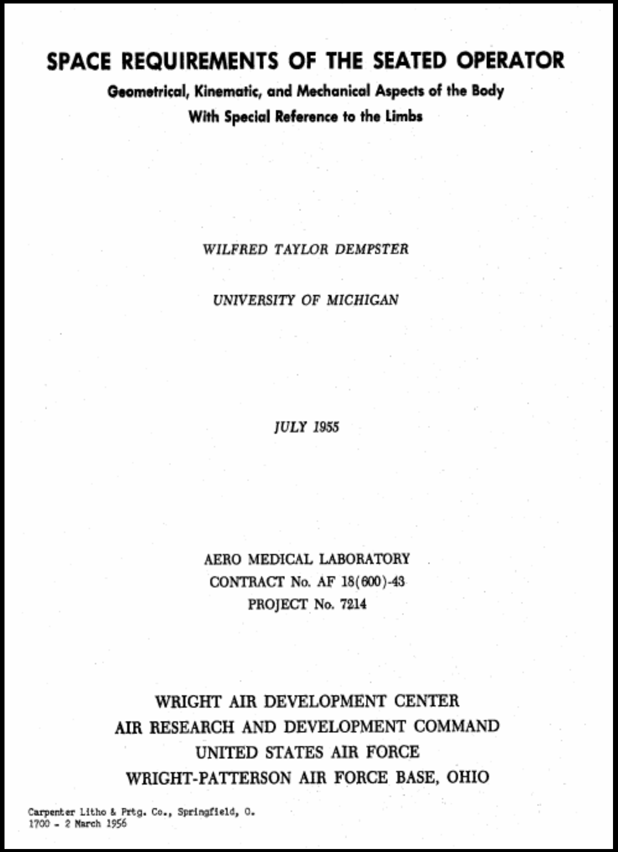
- Inertial properties are difficult to directly determine in living persons
- Scaling methods indirectly estimate inertial properties using non-invasive measurements (i.e., body mass, height, and segment lengths)
- Earliest attempts
  - Harless (1860)
    - Estimated mass (scales), center of mass (balance plate), and volume (principle of Archimedes)
    - 2 executed (beheaded) criminals separated into 18 segments
  - Braune and Fischer (1889)
    - Estimated mass, center of mass (hanging from 3 different axes), and volume
    - 3 dead German soldiers separated at joints
  - Fischer (1906)
    - Estimated moment of inertia (?)
    - 2 additional cadavers



Cadaver Measurements

Landmark work (currently in use) created by the U.S. Air Force in *Space Requirements of the Seated Operator* (Dempster 1955)

- Outlined procedures for measuring inertial properties of 8 cadavers (mean age = 69)
- Recorded segment lengths, masses, and volumes
- Calculated segment centers of mass (balancing technique)
- Calculated segment moment of inertia (pendulum technique)



SPACE REQUIREMENTS OF THE SEATED OPERATOR

Geometrical, Kinematic, and Mechanical Aspects of the Body
With Special Reference to the Limbs

WILFRED TAYLOR DEMPSTER

UNIVERSITY OF MICHIGAN

JULY 1955

AERO MEDICAL LABORATORY
CONTRACT No. AF 18(600)-43
PROJECT No. 7214

WRIGHT AIR DEVELOPMENT CENTER
AIR RESEARCH AND DEVELOPMENT COMMAND
UNITED STATES AIR FORCE
WRIGHT-PATTERSON AIR FORCE BASE, OHIO

Carpenter Litho & Prtg. Co., Springfield, O.
1700 - 2 March 1956

Cadaver Measurements

Landmark work (currently in use) created by the U.S. Air Force in *Space Requirements of the Seated Operator* (Dempster 1955)

- Included tables for proportionally determining inertial properties
  - Segment masses as proportions of the total body mass
  - Centers of mass and radii of gyration as proportions of segment lengths

TABLE 10

MASS OF BODY PARTS

Weights in grams; percentages are ratios to total body weight.

Cadaver Number	Body Weight	Trunk Minus Limbs	%	Trunk Minus Shoulders	%	Both Shoulders	%
14815	51364	31363	61.1	26818	52.2	4310	8.4
15059	58409	32955	56.4	26705	45.7	6535	11.2
15062	58409	(34558)	59.1	(27670)	47.3	6888	11.8
15095	49886	29300	58.7	24431	49.0	5743	11.5
15097	72500	40568	56.0	33409	46.1	8039	11.1
15168	71364	38369	53.8	33377	46.8	7229	10.1
15250	60455	31558	52.2	25909	42.9	5708	9.4
15251	55909	30341	54.3	25341	45.3	4942	8.8
Mean %			56.5		46.9		10.3

Cadaver Number	Head and Neck	%	Thorax	%	Abdomen Plus Pelvis	%
14815	----	---	----	----	-----	----
15059	3797	6.5	4803	8.2	18182	31.1
15062	5227	8.9	6136	10.5	16364	28.0
15095	4348	8.7	5341	10.7	14515	29.1
15097	5337	7.4	8754	12.1	19187	22.0
15168	4850	6.8	9053	12.7	17237	24.1
15250	4371	7.2	6620	10.9	(14918)	24.7
15251	4340	7.8	6637	11.9	(14364)	25.7
Mean %		7.9		11.0		26.4

TABLE 11

MASS, UPPER EXTREMITY

Weights are in grams; percentages represent ratios to total body weight.

Cadaver Number	Entire Upper Extremity	%	Arm	%	Forearm and Hand	%	Forearm	%	Hand	%
					Left Side					
14815	2720	5.3	1157	2.3	1290	2.5	850	1.7	445	0.9
15059	2770	4.7	1541	2.6	1256	2.2	934	1.6	325	0.6
15062	2485	4.3	1373	2.4	1080	1.8	747	1.3	332	0.6
15095	2132	4.3	1133	2.3	1003	2.0	703	1.4	317	0.6
15097	3899	5.4	2199	3.0	1691	2.3	1191	1.6	500	0.7
15168	3453	4.8	1909	2.7	1515	2.1	1104	1.5	417	0.6
15250	3080	5.1	1663	2.8	1400	2.3	1002	1.7	390	0.6
15251	2459	4.4	1315	2.4	1140	2.0	780	1.4	339	0.6
Mean %		4.8		2.6		2.1		1.5		0.6
					Right Side					
14815	2641	5.1	1212	2.4	1342	2.6	865	1.7	457	0.9
15059	3277	5.6	1920	3.3	1340	2.3	995	1.7	352	0.6
15062	2695	4.6	1528	2.6	1134	1.9	815	1.4	311	0.5
15095	2125	4.3	1123	2.3	1024	2.1	710	1.4	317	0.6
15097	3947	5.4	2171	3.0	1777	2.5	1250	1.7	517	0.7
15168	3673	5.1	1970	2.8	1699	2.4	1265	1.8	452	0.6
15250	3035	5.0	1614	2.7	1414	2.3	1021	1.7	400	0.7
15251	2394	4.3	1372	2.5	1017	1.8	713	1.3	295	0.5
Mean %		4.9		2.7		2.2		1.6		0.6

TABLE 12

MASS, LOWER EXTREMITY

Weights are in grams; percentages represent ratios to total body weight.

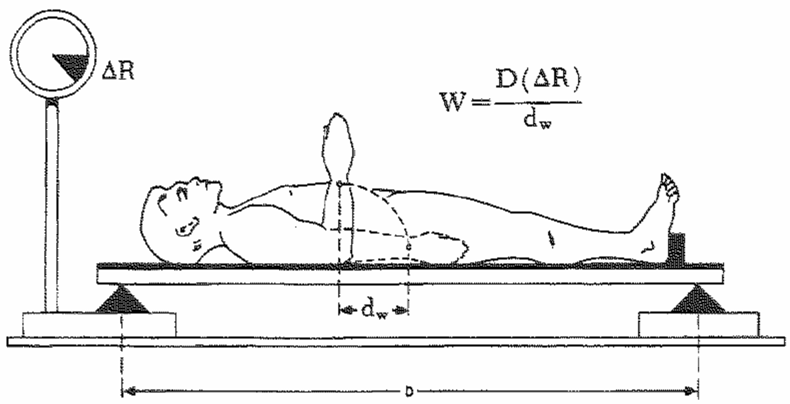
Cadaver Number	Entire Lower Extremity	%	Thigh	%	Leg and Foot	%	Leg	%	Foot	%
					Left Side					
14815	6255	12.1	3495	6.8	2602	5.1	1961	3.8	725	1.4
15059	9855	16.9	6482	11.1	3384	5.8	2629	4.5	760	1.3
15062	8390	14.4	5520	9.5	2835	4.9	2080	3.6	754	1.3
15095	8313	16.7	5285	10.5	3041	6.1	2218	4.4	814	1.6
15097	11907	16.4	7093	9.8	4846	6.7	3860	5.3	967	1.3
15168	11111	15.6	6258	8.8	4812	6.7	3552	5.0	1209	1.7
15250	11337	18.8	7700	12.7	4045	6.7	2991	4.9	949	1.6
15251	(8092)	15.0	4660	8.3	3432	6.1	2564	4.6	796	1.4
Mean %		15.7		9.7		6.0		4.5		1.4
					Right Side					
14815	6176	12.0	3385	6.6	2613	5.1	1963	3.8	655	1.3
15059	9580	16.4	6115	10.5	3472	5.9	2674	4.6	800	1.4
15062	8303	14.2	5370	9.2	2907	5.0	2165	3.7	746	1.3
15095	7715	15.5	4770	9.6	2878	5.8	2205	4.4	767	1.5
15097	11920	16.4	7155	9.9	4825	6.7	3899	5.4	924	1.3
15168	11904	16.7	6902	9.7	4765	6.7	3606	5.1	1095	1.5
15250	11791	19.5	7215	11.9	3955	6.5	2954	4.9	865	1.4
15251	(8457)	15.1	5135	9.2	3322	5.9	2459	4.4	808	1.4
Mean %		15.7		9.6		5.9		4.5		1.4

Cadaver Measurements

Many studies have been conducted since Dempster's 1955 work

(2 are noteworthy because they defined segments using palpable bony landmarks)

- Clauser, McConville, and Young (1969)
  - Measured mass, center of mass, and volume
  - 13 male cadavers dissected into 14 segments



Determination of Forearm-Hand Weight

W - Weight of Forearm-Hand

ΔR - Difference Between Scale Readings

D - Distance Between Supports

d_w - Displacement of Center of Mass of Forearm-Hand

Figure 1. Estimation of a Segment's Weight by the Method of Reaction Change.

Cadaver Measurements

Many studies have been conducted since Dempster's 1955 work

(2 are noteworthy because they defined segments using palpable bony landmarks)

- Clauser, McConville, and Young (1969)
  - Measured mass, center of mass, and volume
  - 13 male cadavers dissected into 14 segments
- Chandler *et al*. (1975)
  - Measured mass, center of mass, principal moments of inertia, and volume
  - 6 male cadavers dissected into 14 segments

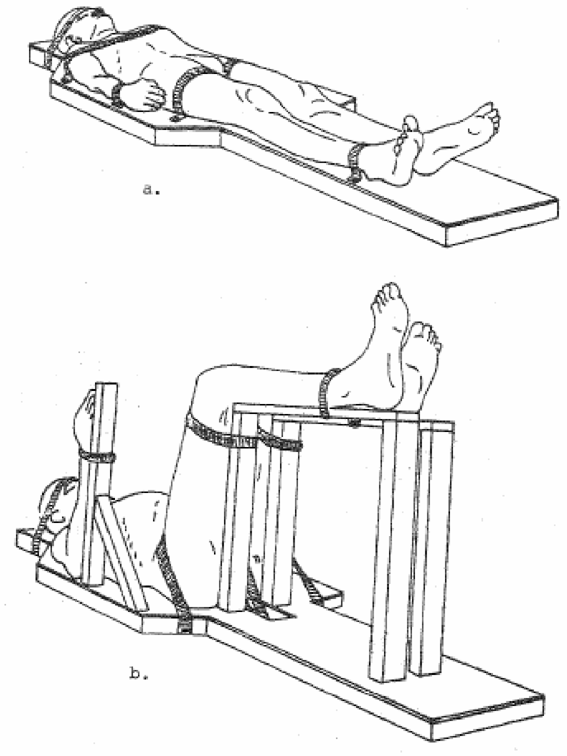


FIGURE 16. STANDING (a) AND SEATED (b) SPECIMEN POSITIONING BOARD WITH SPECIMEN IN PLACE.

Cadaver Measurements

Subsequent studies expanded on methods of Chandler *et al*.

- Hinrichs (1985)
  - Applied regression equations to the moments of inertia
- Vaughan, Davis, and O’Connor (1992)
  - Created regression equations that included anthropometrics such as calf and mid-thigh circumference

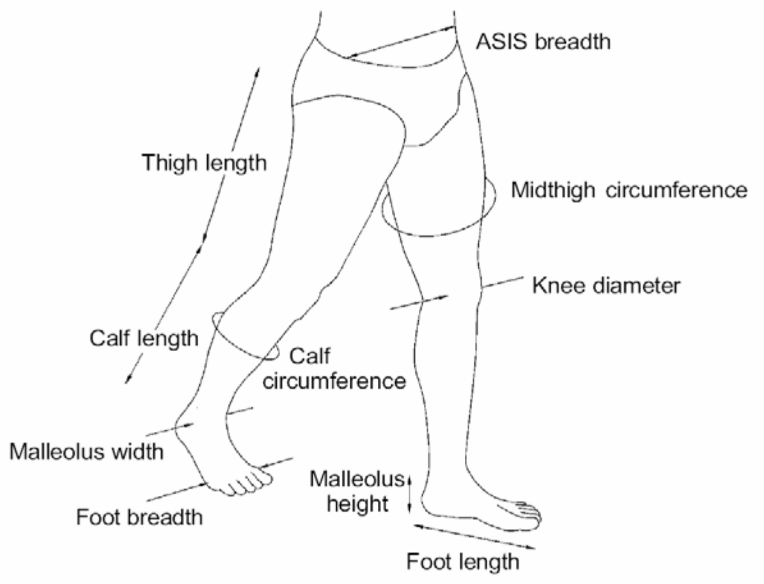
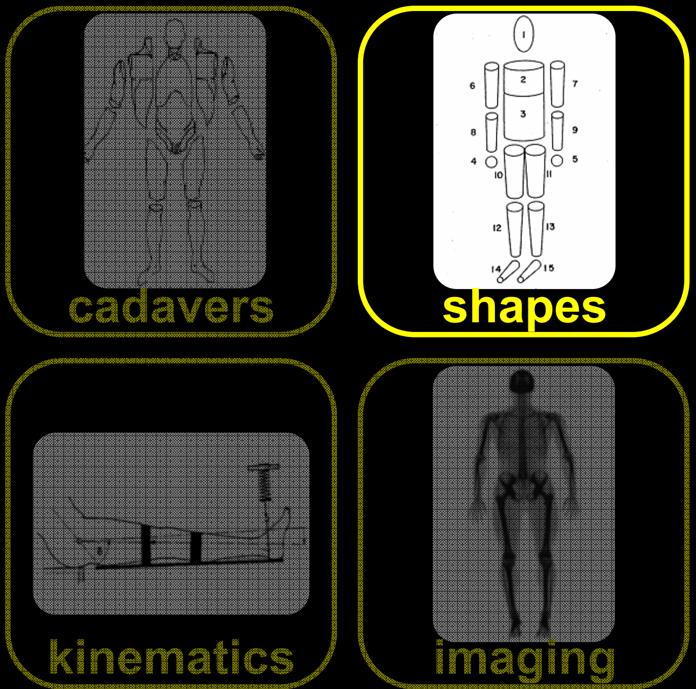


Figure 3.1 The anthropometric measurements of the lower extremity that are required for the prediction of body segment parameters (masses and moments of inertia).

Methods for Determining Inertial Properties

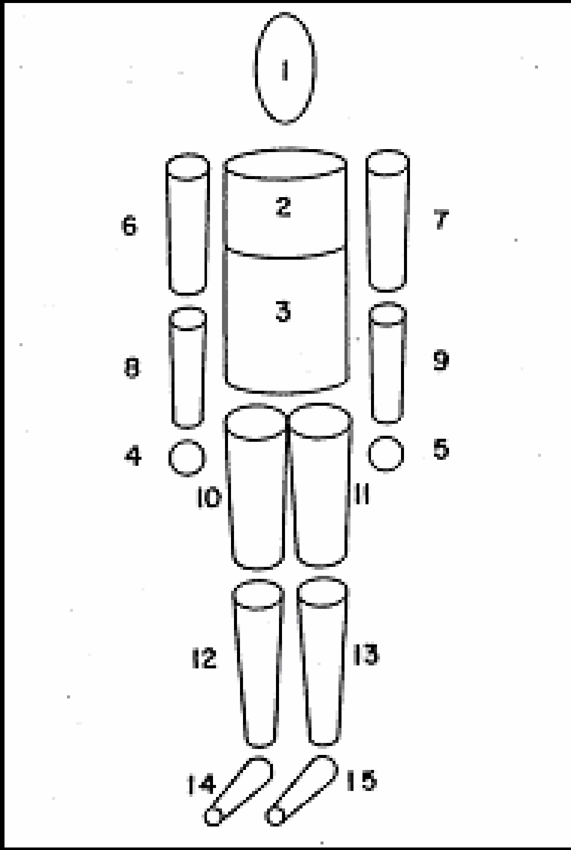


1850
Harless (1860)
Braune & Fisher (1889)
1900
Fisher (1906)
1950
Dempster (1955)
Hanavan (1964)
Hatze (1975)
Dainis (1980)
Hatze (1980)
Vaughan, Andrews, & Hay (1982)
Zatsiorsky & Seluyanov (1983)
Yeadon & Morlock (1989)
Mungiole & Martin (1990)
Yeadon (1990)
Durkin, Dowling, & Andrews (2002)
2000
Durkin & Dowling (2003)

Shape Estimates

Determine inertial properties from a 3D geometric model using uniform density cylinders, ellipsoids, etc.

- Hanavan (1964)
  - Modeled most segments as frusta
  - Modeled hands as spheres, head as an ellipsoid, and trunk as 2 elliptical cylinders
  - Computed moments of inertia using additional anthropometric measures such as mid-thigh circumference, malleolus height, knee diameter, and biacromial breadth



Shape Estimates

Determine inertial properties from a 3D geometric model using uniform density cylinders, ellipsoids, etc.

- Hanavan (1964)
  - Modeled most segments as frusta
  - Modeled hands as spheres, head as an ellipsoid, and trunk as 2 elliptical cylinders
  - Computed moments of inertia using additional anthropometric measures such as mid-thigh circumference, malleolus height, knee diameter, and biacromial breadth
- Hatze (1980)
  - Enhanced Hanavan's method to include more segments and direct measurements
  - Used 242 measures to define a 42-DOF, 17-segment model of the body

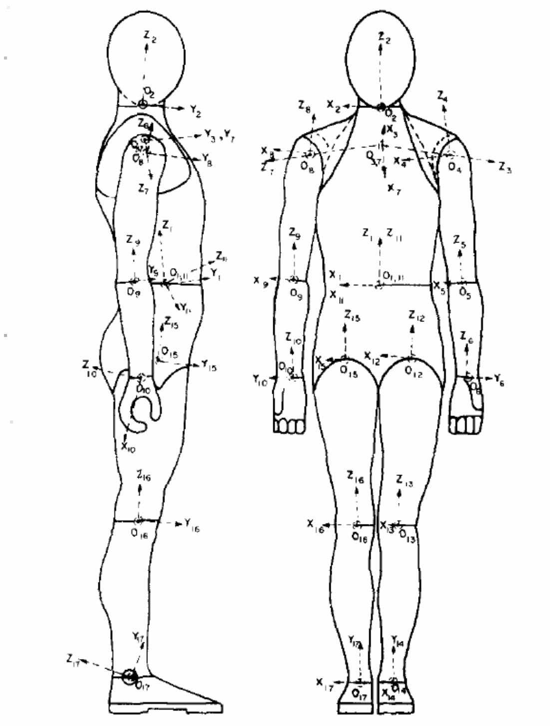
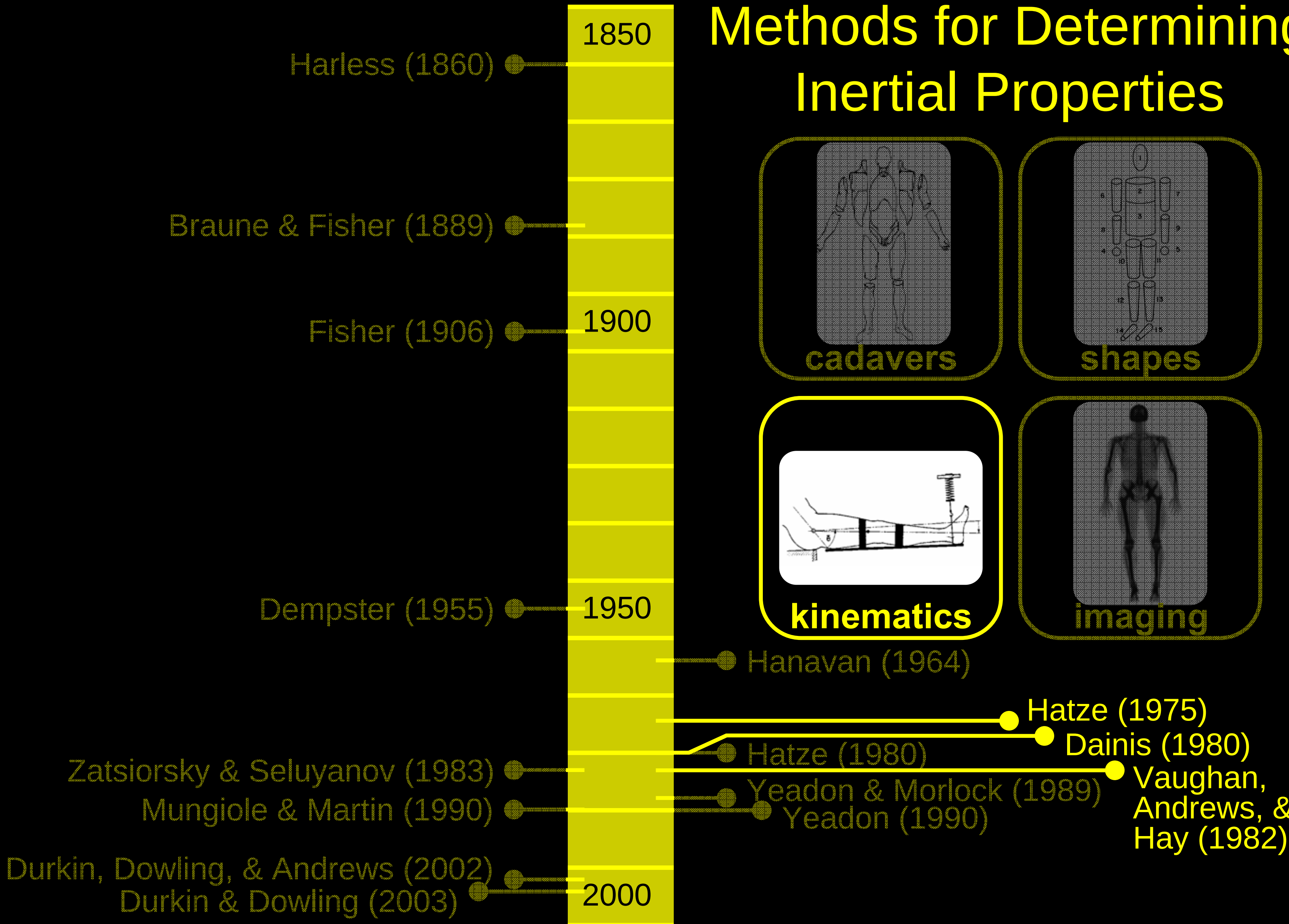


Fig. 1. Lateral and anterior view of 17-segment anthropomorphic model. The shapes of the segments, as depicted here, accurately reflect the morphologies of the model segments. The local (segment-fixed) coordinate systems are also shown.

Methods for Determining Inertial Properties



Kinematic Measurements

Determine inertial properties from *in vivo* kinematic characteristics

- Hatze (1975)
  - Estimated center of mass and moment of inertia using an oscillation technique

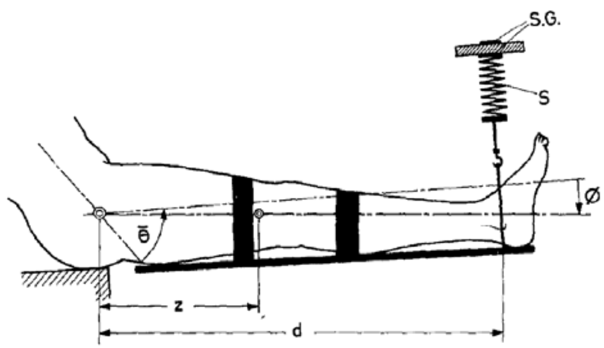


Fig. 1. Schematic representation of the measuring technique. The body segment is suspended horizontally on a spring arrangement *S*, which itself is mounted on a steel bar carrying strain gauges *S.G.* The other symbols are explained in the text

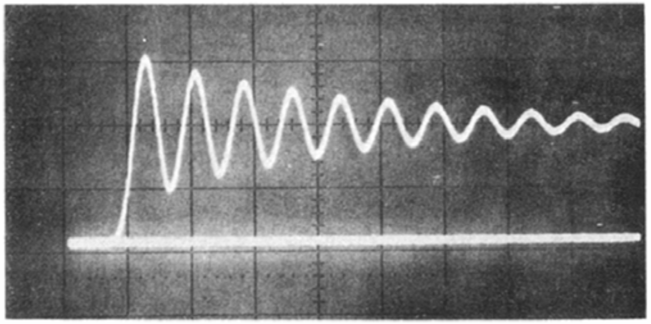


Fig. 4. Photograph of a typical oscillogram of a measurement performed on the total right leg of the subject. One horizontal division represents 0.5 sec while one vertical division corresponds to 30.79 N. The thick trace on the bottom and the x-axis correspond to zero load and equilibrium load (leg resting on the spring arrangement), respectively

Kinematic Measurements

Determine inertial properties from *in vivo* kinematic characteristics

- Hatze (1975)
  - Estimated center of mass and moment of inertia using an oscillation technique
- Drillis, Contini, and Bluestein (1964)
  - Estimated inertial properties using the quick-release method

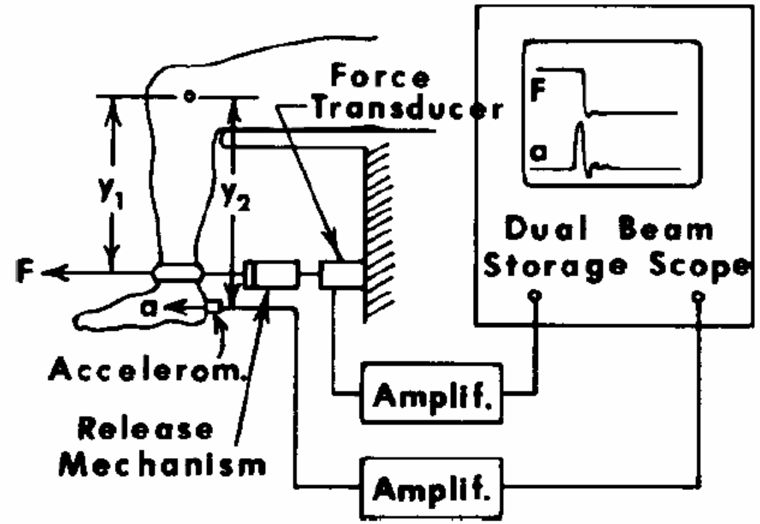
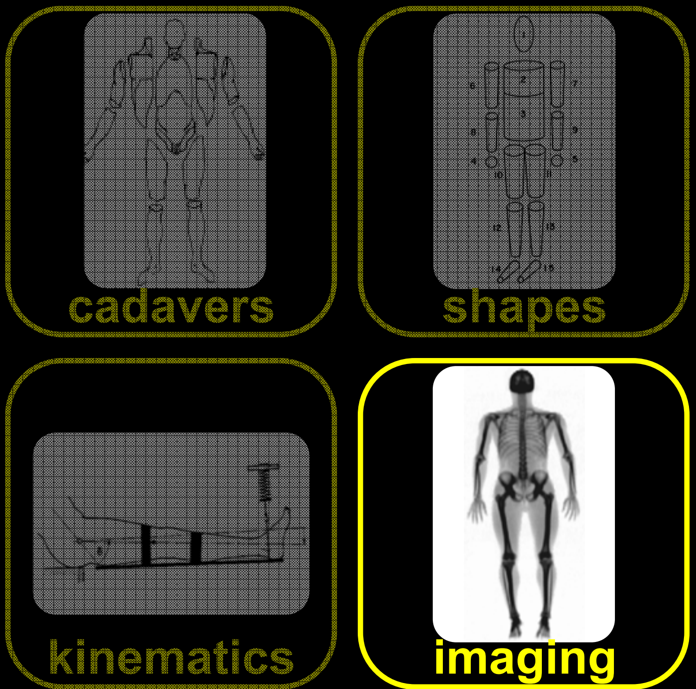


Figure 2: Sketch of the setup used for estimation of the mass moment of inertia of distal segments by the quick release method.

Methods for Determining Inertial Properties



1850

Harless (1860)

Braune & Fisher (1889)

1900

Fisher (1906)

1950

Dempster (1955)

Hanavan (1964)

Hatze (1975)

Dainis (1980)

Hatze (1980)

Vaughan, Andrews, & Hay (1982)

Zatsiorsky & Seluyanov (1983)

Yeadon & Morlock (1989)

Yeadon (1990)

Mungiole & Martin (1990)

Durkin, Dowling, & Andrews (2002)

Durkin & Dowling (2003)

2000

Imaging and Meshing Techniques

Determine inertial properties from imaging with radiation

- Zatsiorsky and Seluyanov (1983)
  - Computed mass distribution by quantifying the density of incremental slices of each segment obtained with gamma mass scanning techniques
  - 100 male and 15 female living subjects (included young subjects)
  - Applied regression equations to customize inertial properties
- Durkin and Dowling (2003) & Durkin, Dowling, and Andrews (2002)
  - Used dual energy x-ray absorptiometry (DEXA) to measure density and geometry
  - 25 males and 25 females (19-30 yrs), 25 males and 25 females (55+ yrs)
  - Developed regression equations for upper and lower extremity segments

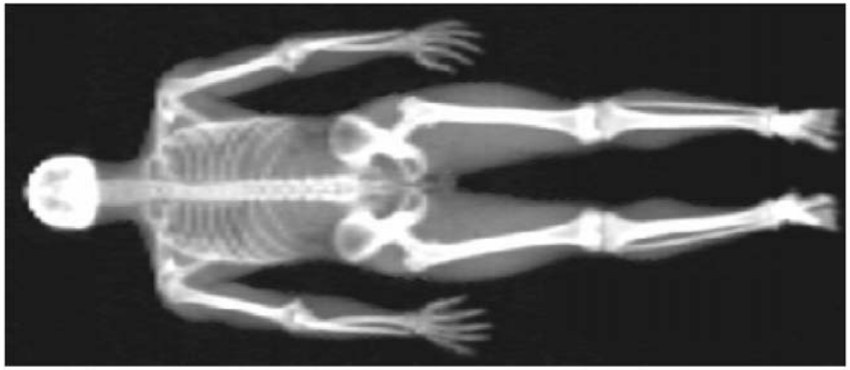


Fig. 1 Density image of a whole body DEXA scan of a human male. Materials that are more dense appear more white, enabling the user to see the skeletal system and more accurately digitize body segments and select segment end-points.

Imaging and Meshing Techniques

Determine inertial properties from imaging without radiation

- Jensen (1978)
  - Combined shapes and photos to find principal axes of ellipses that matched front and side view images of subjects
  - 3 boys of different body types (thin, average, and fat)
- Mungiole and Martin (1990)
  - Estimated inertial properties from MRI data
    - Transverse slices 2.5 cm apart along longitudinal axis
    - Manually segmented into bone, muscle, and fat
    - Used the densities reported by Clauser *et al*. 1969
  - 12 adult male distance runners

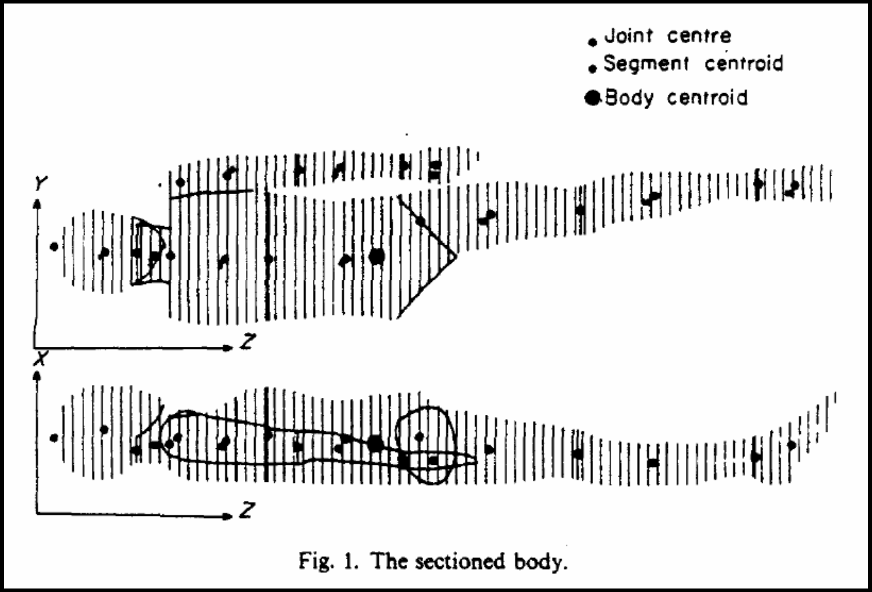
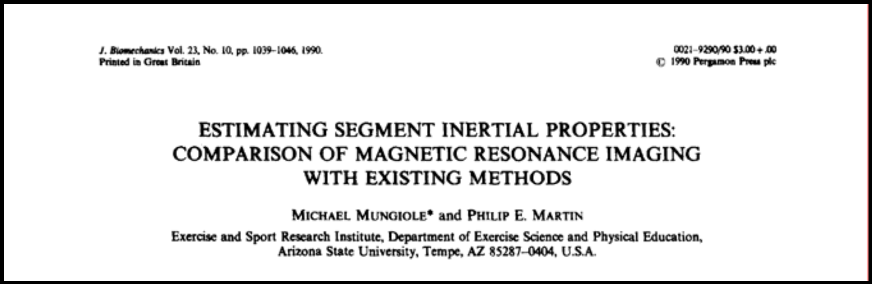


Fig. 1. The sectioned body.


J. Biomechanics Vol. 23, No. 10, pp. 1039–1046, 1990.
Printed in Great Britain

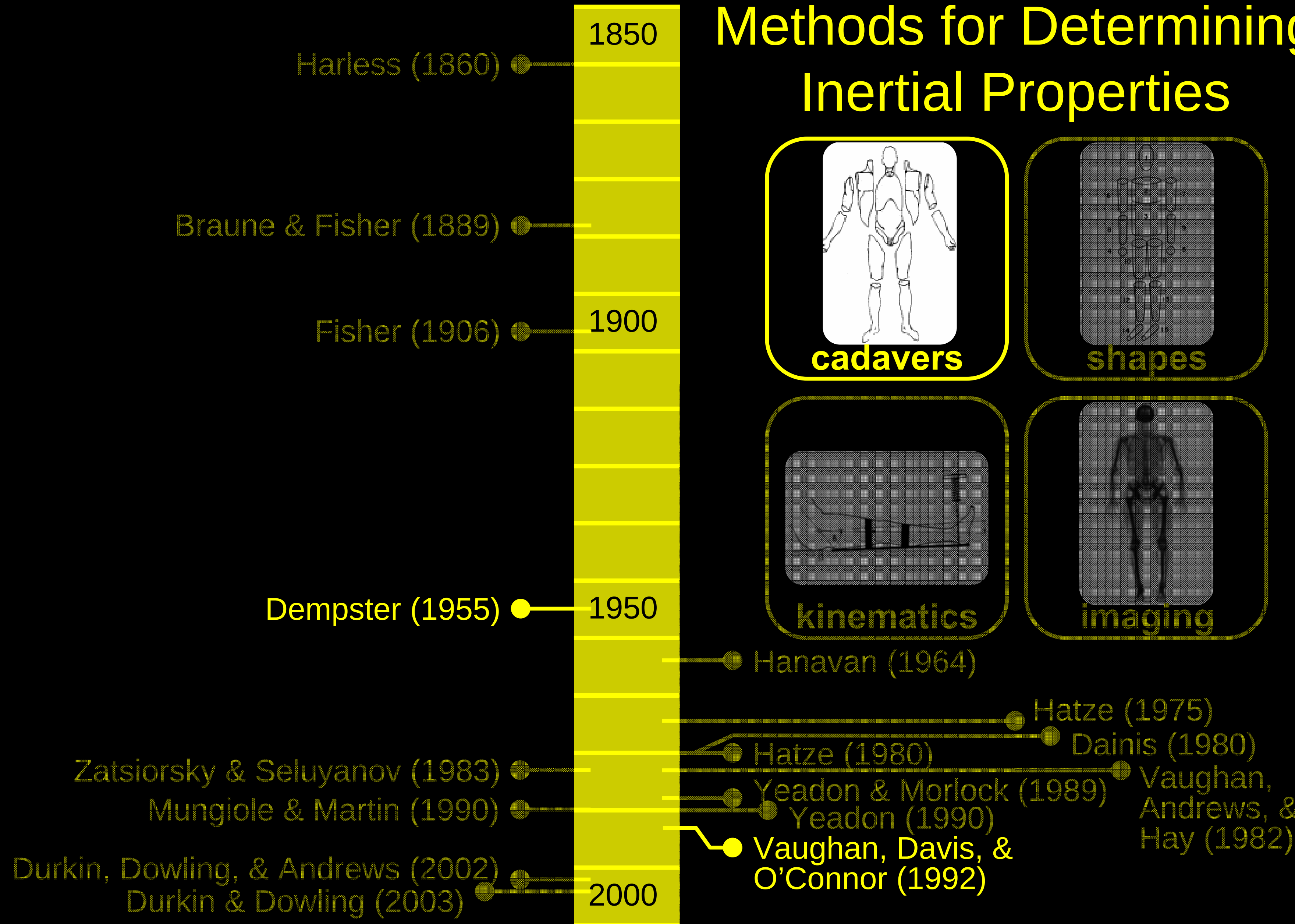
0021-9290/90 $3.00 + .00
© 1990 Pergamon Press plc

ESTIMATING SEGMENT INERTIAL PROPERTIES: COMPARISON OF MAGNETIC RESONANCE IMAGING WITH EXISTING METHODS

MICHAEL MUNGIOLE* and PHILIP E. MARTIN

Exercise and Sport Research Institute, Department of Exercise Science and Physical Education, Arizona State University, Tempe, AZ 85287–0404, U.S.A.

Methods for Determining Inertial Properties



2D Example
mass

Use the proportions in Dempster (1955) to calculate the thigh mass for a 90.0 kg person.

Dempster's Inertial Properties

Segment	Endpoints (proximal to distal)	Segmental mass/ total mass	Center of mass/ segment length		Radius of gyration/ segment length		
		P	$R_{proximal}$	R_{distal}	K_{cm}	$K_{proximal}$	K_{distal}
Thigh	Hip to knee center	0.100	0.433	0.567	0.323	0.540	0.653

$$m_{thigh} = P_{thigh} m_{total}$$
$$m_{thigh} = 0.100 \times 90.0$$
$$m_{thigh} = 9.00 \text{ kg}$$

2D Example

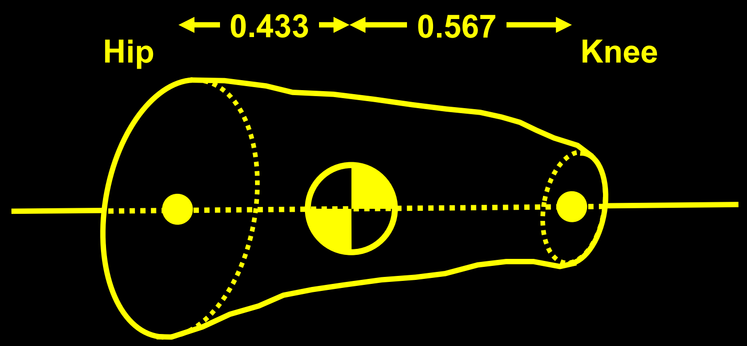
center of mass

Use the proportions in Dempster (1955) to calculate the thigh center of mass given the thigh length is 0.463 m.

Dempster's Inertial Properties

Segment	Endpoints (proximal to distal)	Segmental mass/ total mass	Center of mass/ segment length		Radius of gyration/ segment length		
		P	$R_{proximal}$	R_{distal}	K_{cm}	$K_{proximal}$	K_{distal}
Thigh	Hip to knee center	0.100	0.433	0.567	0.323	0.540	0.653

$$r_{thigh} = R_{proximal(thigh)} l_{thigh}$$
$$r_{thigh} = 0.433 \times 0.463$$
$$r_{thigh} = 0.201\,\text{m}$$

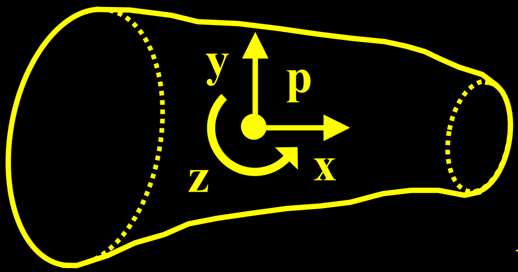


2D Example
moment of inertia

Use the proportions in Dempster (1955) to calculate the thigh moment of inertia (I_{zz}) about its center of mass for a 90.0 kg person given the thigh length is 0.463 m.

Dempster's Inertial Properties

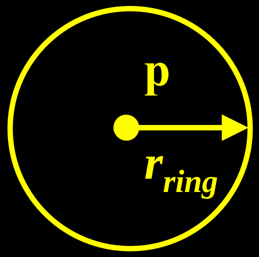
Segment	Endpoints (proximal to distal)	Segmental mass/ total mass	Center of mass/ segment length		Radius of gyration/ segment length		
		P	$R_{proximal}$	R_{distal}	K_{cm}	$K_{proximal}$	K_{distal}
Thigh	Hip to knee center	0.100	0.433	0.567	0.323	0.540	0.653



$$m_{thigh} = m_{ring}$$

$$I_{zz}^{thigh/p} = I_{zz}^{ring/p}$$

$$I_{zz}^{thigh/p} = m_{thigh}(k_{zz}^{thigh/p})^2$$



$$r_{ring} = k_{zz}^{thigh/p}$$

2D Example
moment of inertia

Use the proportions in Dempster (1955) to calculate the thigh moment of inertia about its center of mass for a 90.0 kg person given the thigh length is 0.463 m.

Dempster's Inertial Properties

Segment	Endpoints (proximal to distal)	Segmental mass/ total mass	Center of mass/ segment length		Radius of gyration/ segment length		
		P	$R_{proximal}$	R_{distal}	K_{cm}	$K_{proximal}$	K_{distal}
Thigh	Hip to knee center	0.100	0.433	0.567	0.323	0.540	0.653

$$k_{zz}^{thigh/cm} = K_{cm(thigh)} l_{thigh}$$
$$k_{zz}^{thigh/cm} = 0.323 \times 0.463$$
$$k_{zz}^{thigh/cm} = 0.1495\ \text{m}$$

$$I_{zz}^{thigh/cm} = m_{thigh} (k_{zz}^{thigh/cm})^2$$
$$I_{zz}^{thigh/cm} = 9.00 \times 0.1495^2$$
$$I_{zz}^{thigh/cm} = 0.201\ \text{kg} \cdot \text{m}^2$$

3D Example
mass

Use the regression equations in Vaughan, Davis, and O'Connor (1992) to calculate the thigh mass for a 90.0 kg person given the thigh length is 0.463 m and mid-thigh circumference is 0.445 m.

Mass of thigh = (0.1032)(Total body mass)
+ (12.76)(Thigh length)(Midthigh circumference)²
+ (-1.023) (3.4)

$$m_{thigh} = 0.1032\, m_{total} + 12.76\, l_{thigh} (c_{thigh})^2 - 1.023$$

$$m_{thigh} = (0.1032 \times 90.0) + (12.76 \times 0.463 \times 0.445^2) - 1.023$$

$$m_{thigh} = 9.44 \text{ kg}$$

3D Example
mass

Study	m_{thigh}	m_{total}	l_{thigh}	c_{thigh}
Dempster (1955)	9.00	90.0	-	-
Vaughan *et al*. (1992)	9.44	90.0	0.463	0.445
	9.00	↓**85.8**	↓**0.291**	↓**0.353**

$$m_{thigh} = 0.1032\, m_{total} + 12.76\, l_{thigh} (c_{thigh})^2 - 1.023$$

$$m_{total} = [m_{thigh} + 1.023 - 12.76\, l_{thigh} (c_{thigh})^2] / 0.1032 = \mathbf{85.8} \text{ kg}$$

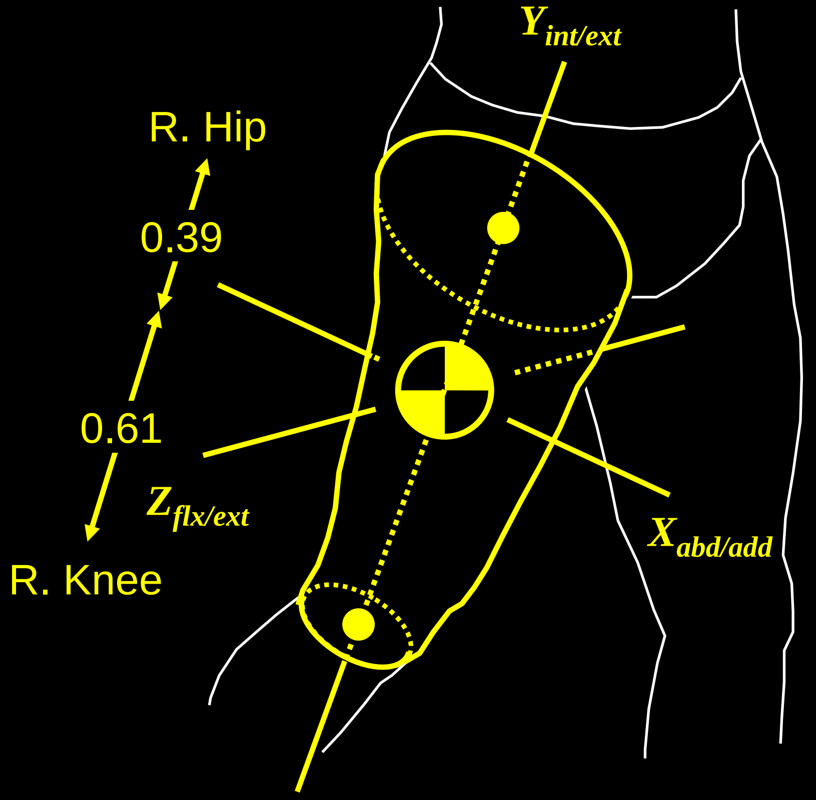
$$l_{thigh} = (m_{thigh} + 1.023 - 0.1032\, m_{total}) / [12.76\, (c_{thigh})^2] = \mathbf{0.291} \text{ m}$$

$$c_{thigh} = [(m_{thigh} + 1.023 - 0.1032\, m_{total}) / (12.76\, l_{thigh})]^{0.5} = \mathbf{0.353} \text{ m}$$

3D Example
center of mass

Use the proportions in Vaughan, Davis, and O'Connor (1992) to calculate the thigh center of mass given the thigh length is 0.463 m.

$$r_{thigh} = R_{proximal(thigh)} l_{thigh}$$
$$r_{thigh} = 0.390 \times 0.463$$
$$r_{thigh} = 0.181 \text{m}$$



3D Example

center of mass

Study	r_{thigh}	$R_{proximal(thigh)}$	l_{thigh}
Dempster (1955)	0.201	0.433	0.463
Vaughan *et al*. (1992)	0.181	0.390	0.463
	0.201	↑**0.433**	↑**0.515**

$$r_{thigh} = R_{proximal(thigh)} l_{thigh}$$

$$R_{proximal(thigh)} = r_{thigh} / l_{thigh} = \mathbf{0.433}$$

$$l_{thigh} = r_{thigh} / R_{proximal(thigh)} = \mathbf{0.515}\text{ m}$$

3D Example

moment of inertia

Use the regression equations in Vaughan, Davis, and O'Connor (1992) to calculate the thigh moments of inertia about its center of mass for a 90.0 kg person given the thigh length is 0.463 m and mid-thigh circumference is 0.445 m.

Moment of inertia of thigh about the flexion/extension axis=
(0.00762)(Total body mass) x
[(Thigh length)^2 + 0.076 (Midthigh circumference)^2] + 0.0115

(3.11)

$$I_{zz}^{thigh/cm} = 0.00762\ m_{total}[(l_{thigh})^2 + 0.076(c_{thigh})^2] + 0.01153$$

$$I_{zz}^{thigh/cm} = 0.00762 \times 90.0 \times (0.463^2 + 0.076 \times 0.445^2) + 0.01153$$
$$I_{zz}^{thigh/cm} = 0.169\ \text{kg} \cdot \text{m}^2$$

3D Example
moment of inertia

Study	$I_{flex(cm(thigh))}$	M_{total}	L_{thigh}	C_{thigh}
Dempster (1955)	0.201	90.0	0.463	-
Vaughan *et al*. (1992)	0.169	90.0	0.463	0.445
	0.201	↑**108**	↑**0.511**	↑**0.903**

$$I_{zz}^{thigh/cm} = 0.00762\, m_{total}[(l_{thigh})^2 + 0.076(c_{thigh})^2] + 0.01153$$

$$m_{total} = (I_{zz}^{thigh/cm} - 0.01153)/\{0.00762[(l_{thigh})^2 + 0.076(c_{thigh})^2]\} = \mathbf{108}\text{ kg}$$

$$l_{thigh} = [(I_{zz}^{thigh/cm} - 0.01153)/(0.0762\, m_{total}) - 0.076(c_{thigh})^2]^{0.5} = \mathbf{0.511}\text{ m}$$

$$c_{thigh} = \{[(I_{zz}^{thigh/cm} - 0.01153)/(0.0762\, m_{total}) - (l_{thigh})^2]/0.076\}^{0.5} = \mathbf{0.903}\text{ m}$$

3D Example
moment of inertia

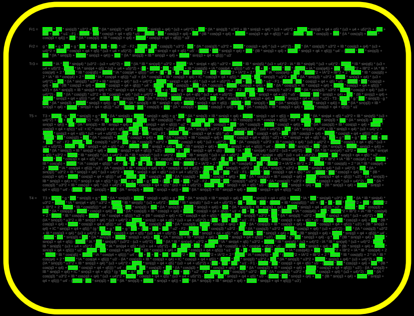
Use the regression equations in Vaughan, Davis, and O'Connor (1992) to calculate the thigh moments of inertia about its center of mass for a 90.0 kg person given the thigh length is 0.463 m and mid-thigh circumference is 0.445 m.

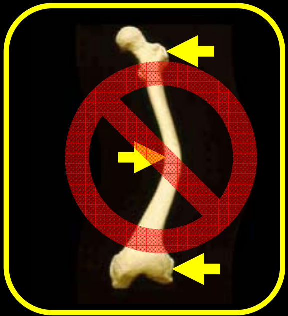
$$I_{zz}^{thigh/cm} = 0.00762\, m_{total}[(l_{thigh})^2 + 0.076(c_{thigh})^2] + 0.01153$$
$$I_{zz}^{thigh/cm} = 0.00762 \times 90.0 \times (0.463^2 + 0.076 \times 0.445^2) + 0.01153$$
$$I_{zz}^{thigh/cm} = 0.169\ \text{kg} \cdot \text{m}^2$$

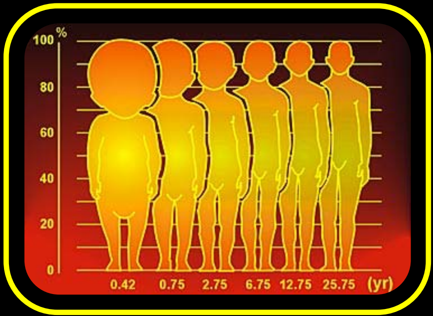
$$I_{xx}^{thigh/cm} = 0.00726\, m_{total}[(l_{thigh})^2 + 0.076(c_{thigh})^2] + 0.01186$$
$$I_{xx}^{thigh/cm} = 0.00726 \times 90.0 \times (0.463^2 + 0.076 \times 0.445^2) + 0.01186$$
$$I_{xx}^{thigh/cm} = 0.162\ \text{kg} \cdot \text{m}^2$$

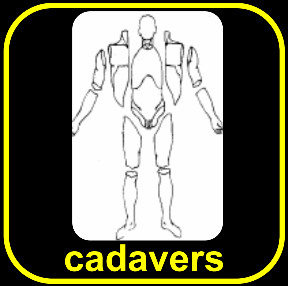
$$I_{yy}^{thigh/cm} = 0.00151\, m_{total}(c_{thigh})^2 + 0.00305$$
$$I_{yy}^{thigh/cm} = 0.00151 \times 90.0 \times 0.445^2 + 0.00305$$
$$I_{yy}^{thigh/cm} = 0.030\ \text{kg} \cdot \text{m}^2$$

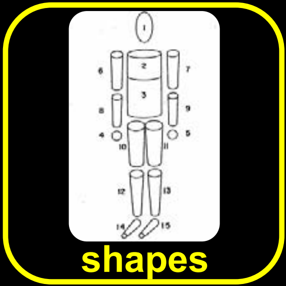
Summary

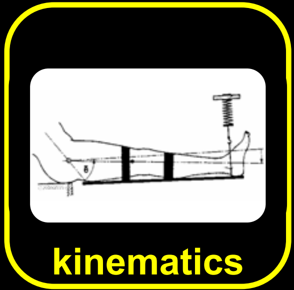


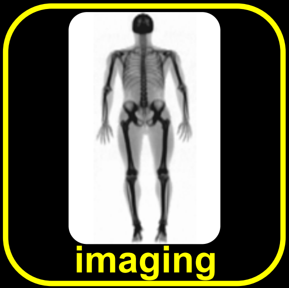












- Inertial properties appear often in equations of motion
- Rigid body assumptions simplify a complex musculoskeletal system
- Inertial properties can be scaled across certain body sizes
- Four categories of methods for determining inertial properties

Why Do We Care?

- Accurate inertial properties are desirable

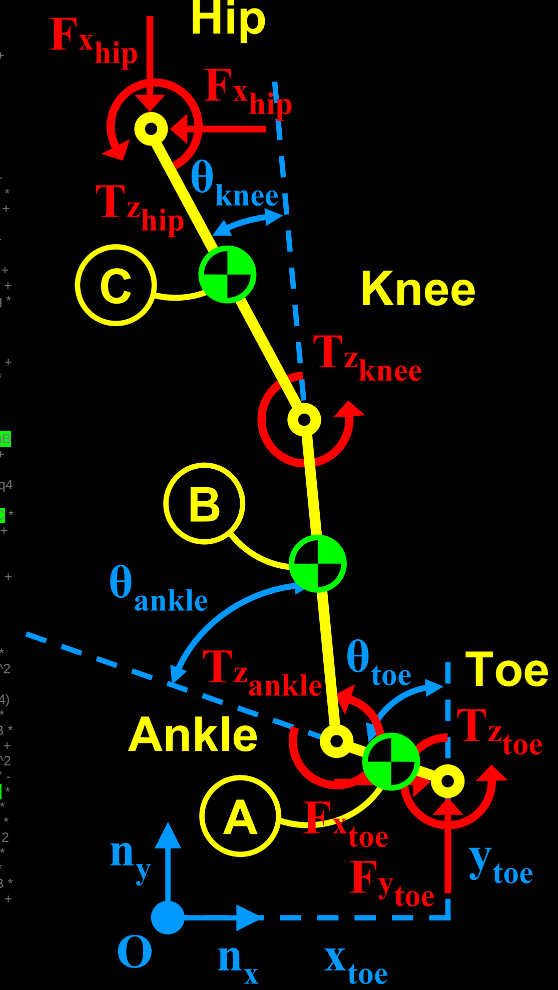
But…

- Errors in inertial properties may have little effect on kinetic measurements (e.g., joint moments or powers)
  - Especially when the body is in contact with the environment
  - The relative magnitudes of inertial forces ($-ma$) and especially moments ($-I\alpha$) are small compared with the moments caused by ground reactions

Should We Care?

$F = ma$

```
Fr1 = comA * mA * sin(q3) * u3^2 + mB * (IA * sin(q3) * u3^2 + comB * sin(q3 + q4) * (u3 + u4)^2) + mC * (IA * sin(q3) * u3^2 + IB * sin(q3 + q4) * (u3 + u4)^2 + comC * sin(q3 + q4 + q5) * (u3 + u4 + u5)^2) + (mA + mB + mC) * u1' - F1 - comC * mC * cos(q3 + q4 + q5) * u5' - (comB * mB * cos(q3 + q4) + mC * (IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u4' - (comA * mA * cos(q3) + mB * (IA * cos(q3) + comB * cos(q3 + q4)) + mC * (IA * cos(q3) + IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u3'

Fr2 = g * mA + g * mB + g * mC + (mA + mB + mC) * u2' - F2 - comA * mA * cos(q3) * u3^2 - mB * (IA * cos(q3) * u3^2 + comB * cos(q3 + q4) * (u3 + u4)^2) - mC * (IA * cos(q3) * u3^2 + IB * cos(q3 + q4) * (u3 + u4)^2 + comC * cos(q3 + q4 + q5) * (u3 + u4 + u5)^2) - comC * mC * sin(q3 + q4 + q5) * u5' - (comB * mB * sin(q3 + q4) + mC * (IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u4' - (comA * mA * sin(q3) + mB * (IA * sin(q3) + comB * sin(q3 + q4)) + mC * (IA * sin(q3) + IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u3'

Tr3 = comB * IA * mB * sin(q4) * (u3^2 - (u3 + u4)^2) + mC * (IA * IB * sin(q4) * u3^2 + comC * IA * sin(q4 + q5) * u3^2 + comC * IB * sin(q5) * (u3 + u4)^2 - IA * IB * sin(q4) * (u3 + u4)^2 - comC * IB * sin(q5) * (u3 + u4 + u5)^2 - comC * IA * sin(q4 + q5) * (u3 + u4 + u5)^2) + (IC + comC * mC * (comC + IB * cos(q5) + IA * cos(q4 + q5))) * u5' + (IB + IC + comB * mB * (comB + IA * cos(q4)) + mC * (comC^2 + IB^2 + IA * IB * cos(q4) + 2 * comC * IB * cos(q5) + comC * IA * cos(q4 + q5))) * u4' + (IA + IB + IC + mA * comA^2 + mB * (comB^2 + IA^2 + 2 * comB * IA * cos(q4)) + mC * (comC^2 + IA^2 + IB^2 + 2 * comC * IB * cos(q5) + 2 * IA * IB * cos(q4) + 2 * comC * IA * cos(q4 + q5))) * u3' + (IA * cos(q3) + IB * cos(q3 + q4) + IC * cos(q3 + q4 + q5)) * (comA * mA * sin(q3) * u3^2 + mB * (IA * sin(q3) * u3^2 + comB * sin(q3 + q4) * (u3 + u4)^2) + mC * (IA * sin(q3) * u3^2 + IB * sin(q3 + q4) * (u3 + u4)^2 + comC * sin(q3 + q4 + q5) * (u3 + u4 + u5)^2) + (mA + mB + mC) * u1' - F1 - comC * mC * cos(q3 + q4 + q5) * u5' - (comB * mB * cos(q3 + q4) + mC * (IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u4' - (comA * mA * cos(q3) + mB * (IA * cos(q3) + comB * cos(q3 + q4)) + mC * (IA * cos(q3) + IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u3') + (IA * sin(q3) + IB * sin(q3 + q4) + IC * sin(q3 + q4 + q5)) * (g * mA + g * mB + g * mC + (mA + mB + mC) * u2' - F2 - comA * mA * cos(q3) * u3^2 - mB * (IA * cos(q3) * u3^2 + comB * cos(q3 + q4) * (u3 + u4)^2) - mC * (IA * cos(q3) * u3^2 + IB * cos(q3 + q4) * (u3 + u4)^2 + comC * cos(q3 + q4 + q5) * (u3 + u4 + u5)^2) - comC * mC * sin(q3 + q4 + q5) * u5' - (comB * mB * sin(q3 + q4) + mC * (IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u4' - (comA * mA * sin(q3) + mB * (IA * sin(q3) + comB * sin(q3 + q4)) + mC * (IA * sin(q3) + IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u3') - T3 - comA * g * mA * sin(q3) - g * mB * (IA * sin(q3) + comB * sin(q3 + q4)) - g * mC * (IA * sin(q3) + IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5)) - (comA * mA * sin(q3) + mB * (IA * sin(q3) + comB * sin(q3 + q4)) + mC * (IA * sin(q3) + IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u2' - (comA * mA * cos(q3) + mB * (IA * cos(q3) + comB * cos(q3 + q4)) + mC * (IA * cos(q3) + IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u1'

T5 = T3 + comA * g * mA * sin(q3) + g * mB * (IA * sin(q3) + comB * sin(q3 + q4)) + g * mC * (IA * sin(q3) + IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5)) + comC * mC * (IA * sin(q4 + q5) * u3^2 + IB * sin(q5) * (u3 + u4)^2) + (IC + mC * comC^2) * u5' + (IC + comC * mC * (comC + IB * cos(q5))) * u4' + (IC + comC * mC * (comC + IB * cos(q5) + IA * cos(q4 + q5))) * u3' + (comA * mA * sin(q3) + mB * (IA * sin(q3) + comB * sin(q3 + q4)) + mC * (IA * sin(q3) + IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u2' + (comA * mA * cos(q3) + mB * (IA * cos(q3) + comB * cos(q3 + q4)) + mC * (IA * cos(q3) + IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u1' + IC * cos(q3 + q4 + q5) * (comA * mA * sin(q3) * u3^2 + mB * (IA * sin(q3) * u3^2 + comB * sin(q3 + q4) * (u3 + u4)^2) + mC * (IA * sin(q3) * u3^2 + IB * sin(q3 + q4) * (u3 + u4)^2 + comC * sin(q3 + q4 + q5) * (u3 + u4 + u5)^2) + (mA + mB + mC) * u1' - F1 - comC * mC * cos(q3 + q4 + q5) * u5' - (comB * mB * cos(q3 + q4) + mC * (IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u4' - (comA * mA * cos(q3) + mB * (IA * cos(q3) + comB * cos(q3 + q4)) + mC * (IA * cos(q3) + IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u3') + IC * sin(q3 + q4 + q5) * (g * mA + g * mB + g * mC + (mA + mB + mC) * u2' - F2 - comA * mA * cos(q3) * u3^2 - mB * (IA * cos(q3) * u3^2 + comB * cos(q3 + q4) * (u3 + u4)^2) - mC * (IA * cos(q3) * u3^2 + IB * cos(q3 + q4) * (u3 + u4)^2 + comC * cos(q3 + q4 + q5) * (u3 + u4 + u5)^2) - comC * mC * sin(q3 + q4 + q5) * u5' - (comB * mB * sin(q3 + q4) + mC * (IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u4' - (comA * mA * sin(q3) + mB * (IA * sin(q3) + comB * sin(q3 + q4)) + mC * (IA * sin(q3) + IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u3') - comC * g * mC * sin(q3 + q4 + q5) - comB * IA * mB * sin(q4) * (u3^2 - (u3 + u4)^2) - mC * (IA * IB * sin(q4) * u3^2 + comC * IA * sin(q4 + q5) * u3^2 + comC * IB * sin(q5) * (u3 + u4)^2 - IA * IB * sin(q4) * (u3 + u4)^2 - comC * IB * sin(q5) * (u3 + u4 + u5)^2 - comC * IA * sin(q4 + q5) * (u3 + u4 + u5)^2) - comC * mC * sin(q3 + q4 + q5) * u2' - comC * mC * cos(q3 + q4 + q5) * u1' - (IC + comC * mC * (comC + IB * cos(q5) + IA * cos(q4 + q5))) * u5' - (IB + IC + comB * mB * (comB + IA * cos(q4)) + mC * (comC^2 + IB^2 + IA * IB * cos(q4) + 2 * comC * IB * cos(q5) + comC * IA * cos(q4 + q5))) * u4' - (IA + IB + IC + mA * comA^2 + mB * (comB^2 + IA^2 + 2 * comB * IA * cos(q4)) + mC * (comC^2 + IA^2 + IB^2 + 2 * comC * IB * cos(q5) + 2 * IA * IB * cos(q4) + 2 * comC * IA * cos(q4 + q5))) * u3' - (IA * cos(q3) + IB * cos(q3 + q4) + IC * cos(q3 + q4 + q5)) * (comA * mA * sin(q3) * u3^2 + mB * (IA * sin(q3) * u3^2 + comB * sin(q3 + q4) * (u3 + u4)^2) + mC * (IA * sin(q3) * u3^2 + IB * sin(q3 + q4) * (u3 + u4)^2 + comC * sin(q3 + q4 + q5) * (u3 + u4 + u5)^2) + (mA + mB + mC) * u1' - F1 - comC * mC * cos(q3 + q4 + q5) * u5' - (comB * mB * cos(q3 + q4) + mC * (IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u4' - (comA * mA * cos(q3) + mB * (IA * cos(q3) + comB * cos(q3 + q4)) + mC * (IA * cos(q3) + IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u3') - (IA * sin(q3) + IB * sin(q3 + q4) + IC * sin(q3 + q4 + q5)) * (g * mA + g * mB + g * mC + (mA + mB + mC) * u2' - F2 - comA * mA * cos(q3) * u3^2 - mB * (IA * cos(q3) * u3^2 + comB * cos(q3 + q4) * (u3 + u4)^2) - mC * (IA * cos(q3) * u3^2 + IB * cos(q3 + q4) * (u3 + u4)^2 + comC * cos(q3 + q4 + q5) * (u3 + u4 + u5)^2) - comC * mC * sin(q3 + q4 + q5) * u5' - (comB * mB * sin(q3 + q4) + mC * (IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u4' - (comA * mA * sin(q3) + mB * (IA * sin(q3) + comB * sin(q3 + q4)) + mC * (IA * sin(q3) + IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u3')

T4 = T3 + comA * g * mA * sin(q3) + g * mB * (IA * sin(q3) + comB * sin(q3 + q4)) + g * mC * (IA * sin(q3) + IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5)) + comB * IA * mB * sin(q4) * u3^2 + mC * (IA * IB * sin(q4) * u3^2 + comC * IA * sin(q4 + q5) * u3^2 + comC * IB * sin(q5) * (u3 + u4)^2 - comC * IB * sin(q5) * (u3 + u4 + u5)^2) + (IC + comC * mC * (comC + IB * cos(q5))) * u5' + (IB + IC + mB * comB^2 + mC * (comC^2 + IB^2 + 2 * comC * IB * cos(q5))) * u4' + (comA * mA * sin(q3) + mB * (IA * sin(q3) + comB * sin(q3 + q4)) + mC * (IA * sin(q3) + IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u2' + (comA * mA * cos(q3) + mB * (IA * cos(q3) + comB * cos(q3 + q4)) + mC * (IA * cos(q3) + IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u1' + (IB + IC + comB * mB * (comB + IA * cos(q4)) + mC * (comC^2 + IB^2 + IA * IB * cos(q4) + 2 * comC * IB * cos(q5) + comC * IA * cos(q4 + q5))) * u3' + (IB * cos(q3 + q4) + IC * cos(q3 + q4 + q5)) * (comA * mA * sin(q3) * u3^2 + mB * (IA * sin(q3) * u3^2 + comB * sin(q3 + q4) * (u3 + u4)^2) + mC * (IA * sin(q3) * u3^2 + IB * sin(q3 + q4) * (u3 + u4)^2 + comC * sin(q3 + q4 + q5) * (u3 + u4 + u5)^2) + (mA + mB + mC) * u1' - F1 - comC * mC * cos(q3 + q4 + q5) * u5' - (comB * mB * cos(q3 + q4) + mC * (IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u4' - (comA * mA * cos(q3) + mB * (IA * cos(q3) + comB * cos(q3 + q4)) + mC * (IA * cos(q3) + IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u3') + (IB * sin(q3 + q4) + IC * sin(q3 + q4 + q5)) * (g * mA + g * mB + g * mC + (mA + mB + mC) * u2' - F2 - comA * mA * cos(q3) * u3^2 - mB * (IA * cos(q3) * u3^2 + comB * cos(q3 + q4) * (u3 + u4)^2) - mC * (IA * cos(q3) * u3^2 + IB * cos(q3 + q4) * (u3 + u4)^2 + comC * cos(q3 + q4 + q5) * (u3 + u4 + u5)^2) - comC * mC * sin(q3 + q4 + q5) * u5' - (comB * mB * sin(q3 + q4) + mC * (IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u4' - (comA * mA * sin(q3) + mB * (IA * sin(q3) + comB * sin(q3 + q4)) + mC * (IA * sin(q3) + IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u3') - comB * g * mB * sin(q3 + q4) - g * mC * (IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5)) - comB * IA * mB * sin(q4) * (u3^2 - (u3 + u4)^2) - mC * (IA * IB * sin(q4) * u3^2 + comC * IA * sin(q4 + q5) * u3^2 + comC * IB * sin(q5) * (u3 + u4)^2 - IA * IB * sin(q4) * (u3 + u4)^2 - comC * IB * sin(q5) * (u3 + u4 + u5)^2 - comC * IA * sin(q4 + q5) * (u3 + u4 + u5)^2) - (IC + comC * mC * (comC + IB * cos(q5) + IA * cos(q4 + q5))) * u5' - (comB * mB * sin(q3 + q4) + mC * (IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u2' - (comB * mB * cos(q3 + q4) + mC * (IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u1' - (IB + IC + comB * mB * (comB + IA * cos(q4)) + mC * (comC^2 + IB^2 + IA * IB * cos(q4) + 2 * comC * IB * cos(q5) + comC * IA * cos(q4 + q5))) * u4' - (IA + IB + IC + mA * comA^2 + mB * (comB^2 + IA^2 + 2 * comB * IA * cos(q4)) + mC * (comC^2 + IA^2 + IB^2 + 2 * comC * IB * cos(q5) + 2 * IA * IB * cos(q4) + 2 * comC * IA * cos(q4 + q5))) * u3' - (IA * cos(q3) + IB * cos(q3 + q4) + IC * cos(q3 + q4 + q5)) * (comA * mA * sin(q3) * u3^2 + mB * (IA * sin(q3) * u3^2 + comB * sin(q3 + q4) * (u3 + u4)^2) + mC * (IA * sin(q3) * u3^2 + IB * sin(q3 + q4) * (u3 + u4)^2 + comC * sin(q3 + q4 + q5) * (u3 + u4 + u5)^2) + (mA + mB + mC) * u1' - F1 - comC * mC * cos(q3 + q4 + q5) * u5' - (comB * mB * cos(q3 + q4) + mC * (IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u4' - (comA * mA * cos(q3) + mB * (IA * cos(q3) + comB * cos(q3 + q4)) + mC * (IA * cos(q3) + IB * cos(q3 + q4) + comC * cos(q3 + q4 + q5))) * u3') - (IA * sin(q3) + IB * sin(q3 + q4) + IC * sin(q3 + q4 + q5)) * (g * mA + g * mB + g * mC + (mA + mB + mC) * u2' - F2 - comA * mA * cos(q3) * u3^2 - mB * (IA * cos(q3) * u3^2 + comB * cos(q3 + q4) * (u3 + u4)^2) - mC * (IA * cos(q3) * u3^2 + IB * cos(q3 + q4) * (u3 + u4)^2 + comC * cos(q3 + q4 + q5) * (u3 + u4 + u5)^2) - comC * mC * sin(q3 + q4 + q5) * u5' - (comB * mB * sin(q3 + q4) + mC * (IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u4' - (comA * mA * sin(q3) + mB * (IA * sin(q3) + comB * sin(q3 + q4)) + mC * (IA * sin(q3) + IB * sin(q3 + q4) + comC * sin(q3 + q4 + q5))) * u3')
```



Should We Care?
example

$F = ma$