Programming Assignment #1 for Biomechanics

Due: March 2, 2021

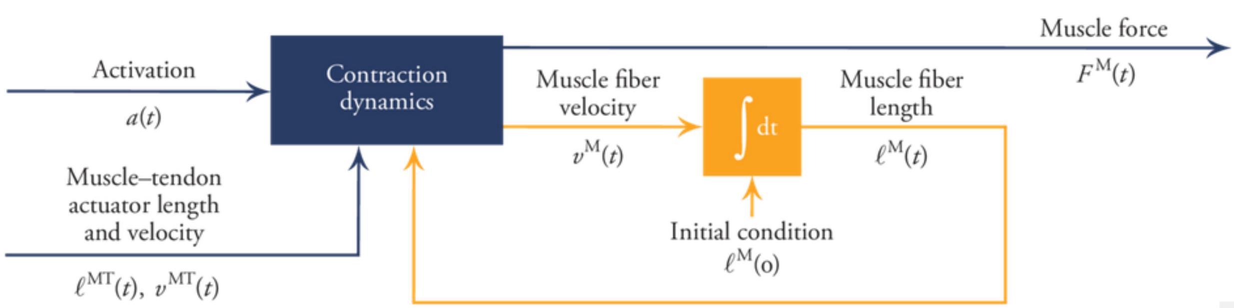
Context:

In generating a movement, the forces muscles generate often accelerate the mass of a limb or the body's center of mass. In the simulation you will first activate a muscle isometrically, and then you will couple a Hill-type muscle model to a mass load and investigate how the series-elasticity influences the power generation in the muscle.

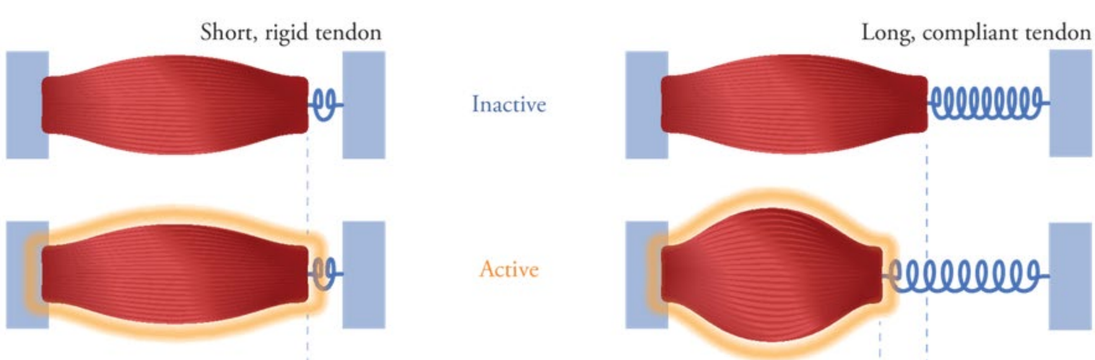
Program:

- The overall objective is simulate the activation of a musculotendon unit to move an initially stationary mass load and determine the power generated by the muscle throughout the movement. The important features to keep in mind are:
  - Activation is not constant, and force scales linearly with activation.
  - Isometric force depends on length and activation.
  - Isometric force scales the force-intercept of the force-velocity curve.
  - Tendon force is equal to muscle force, which is equal to musculotendon force (assume no pennation).
  - The force in the musculotendon unit accelerates the mass.
  - The acceleration, velocity, and position change of the mass is equivalent to the length and velocity of the musculotendon unit.
  - Acceleration can be integrated to get velocity, and velocity can be integrated to get length.

- Use the Muscle Modeling algorithm from *Research Methods in Biomechanics* (Example 9.1) to solve the forces generated by the musculotendon unit. (Note that we will be using different equations from Table 9.1, but the process is the same). This is similar conceptually to the algorithm described in Delp and Uchida, figure 5.12 and pages 126-128:

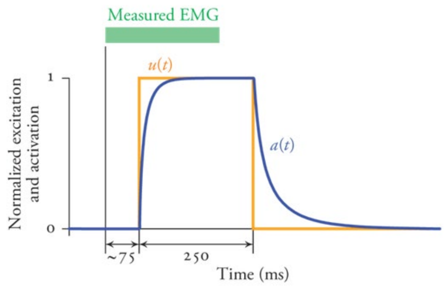


Part 1: Isometric force generation (no mass)



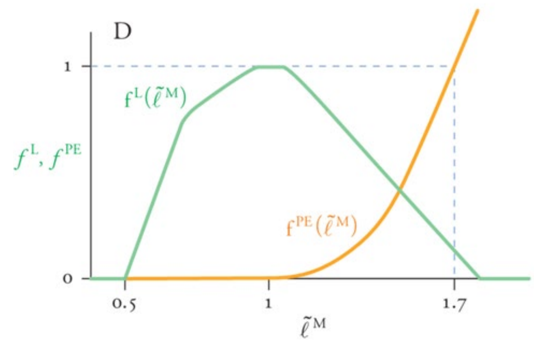
- **Activation** of the initially passive muscle to 100% activation will be modeled by a first-order differential equation with a time constant of τ=0.02 seconds

- Solving the differential equation gives the analytical solution of: $\alpha(t) = 1 - e^{-t/\tau}$ where *α(t)* is the activation level (multiplicative constant for maximal isometric force).
- Activation as shown in Figure 4.16 of Uchida and Delp: Note that here we will use α for activation, because **a** can also stand for acceleration or the Hill's equation constant.

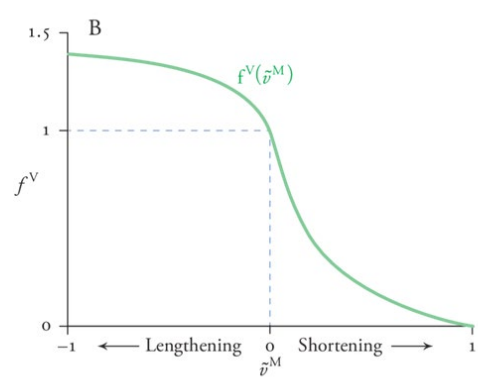


- **The series-elasticity (all lumped in the tendon)** is modeled as a linear spring: $F^T = K_{SE}\Delta L^T$. Perform one simulation with K_{SE}=500 N/m and another with 10000 N/m.

- **The active force-length (FL) curve**: isometric force at each muscle length is defined by the following points:
  - Length = [0.5 0.7 0.8 1 1.5] and Force = [0 0.8 1.0 1.0 0] where length and force are normalized by maximum isometric force (F_o^{max}=20 N) and optimal muscle length (L_o=0.2 m)
  - Interpolate for force values at muscle lengths between the defined points. In Matlab, you can use the command "interp1" to conveniently do the interpolation.
  - For this simulation, muscle length starts at optimal length and then shortens, so the passive force does not come into play.



- **The force-velocity (FV) curve:**
  - Use the constants of a=0.25F_o^{max} and b=2 L_o/second in Hill's equation:
    $v^{muscle} = b * (\alpha F_o(L) - F)/(F + a)$
  - Note that the isometric force in Hill's equation is affected by both activation and length. $\alpha F_o(L)$ is the force normalized by F_o^{max} calculated from the FL curve and multiplied by activation level (α) (so it should be between [0,1]) and *F* is the current muscle force (also normalized by F_o^{max}). v^{muscle} is normalized by L_o. Keep track of whether your calculations are for values that are normalized or unnormalized (this is where commenting your program helps!)



- Setting up your simulation
  - Initial conditions: $L^{muscle}=L_o$, v^{muscle}=0, *F*=0, α=0; tendon is at its slack length so ΔL^T = 0.
  - End conditions: loop while L^{muscle}>0.6*L_o
  - Looping and integration: setup your simulation to loop with a Δt=0.001 seconds
    - Follow steps 1-5 in the "Muscle Model Algorithm" from Robertson et al. Note that some of their terminology is slightly different from what we have been using:
      SEC force, length = F^T, L^T
      CC force, velocity, length = F^M, v^M, L^M
      Muscle length = L^{MT}

- To integrate velocity to get length, use $x(t+\Delta t)=x(t)-v(t)*\Delta t$, where **v is shortening velocity (hence the negative sign)**, x is displacement (or change in length), and Δt is the time step of the simulation

Calculations and plots for part 1 (isometric)

1. Activation as a function of time.
2. Muscle force as a function of time (or tendon force – they should be equal; this is a good check). Do this for two values of K_{SE}. If possible, overlay the two cases on the same graphs.
3. Muscle length as a function of time (another check – the change in tendon length should be opposite to the change in muscle length. Do this for two values of K_{SE}. If possible, overlay the two cases on the same graphs.

Part 2: Adding a mass and calculating power

- **Couple the musculotendon unit to a 5 kg mass**, such that:
  - The force of the muscle causes an acceleration equal to a=F/m, where a is the acceleration, F is the force, and m is the mass
  - To find the displacement of the mass, integrate acceleration twice (i.e., $v(t+\Delta t)=v(t)+a(t)*\Delta t$ and $x(t+\Delta t)=x(t)-v(t)*\Delta t$, where **v is shortening velocity (hence the negative sign)**, x is displacement, and Δt is the time step of the simulation)
  - The displacement of the mass is equal to the length change of the musculotendon unit

  The acceleration, velocity and length calculation should come after calculating tendon force and be used as input to the muscle isometric force and force-velocity calculations.

- Remember that **power generated by the muscle** at each instance of time is $F*v^{muscle}$

Calculations and plots for part 2 (moving a mass)

1. Normalized muscle velocity [units of L_o/s] as a function of time. If possible, overlay the two cases on the same graphs.
2. Normalized power [units $F_o^{max}\bullet L_o/s$] as a function of time for the two cases of K_{SE}. If possible, overlay the two cases on the same graphs.

What to turn in:

1. An algorithm or flow chart of your program, including equations that you used. Another person should be able to read your algorithm and recreate your work.
2. Your commented code
3. Graphs (described in part 1 and part 2).
4. Address the question: How does the series-elasticity help muscle generate power? (2 or 3 sentences should suffice)

Grading:

- The assignment will be worth 40 points
- The assignment will be graded according to the following
  - 12 points for program organization: make sure that your code is easy to follow and has adequate comments (a comment every 3-4 lines on average).
  - 12 points that your program follows the algorithm you describe and that is outlined in this assignment.
  - 16 points for graphs and answer to question: appropriate variables plotted, graphs labeled appropriately; answer uses simulation results in explanation.