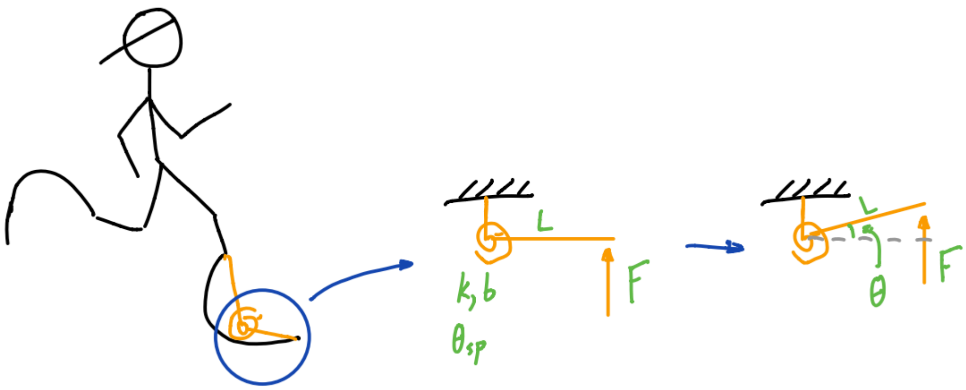
MATLAB PROBLEM 1

A quick note before you start – you will answer all questions for this problem within this Word document. When finished, please save the document as a .pdf, which you will upload directly to Gradescope.

The majority of prosthetic legs today are made from composite springs, which deflect predictably when loads are applied. A common goal when designing a passive leg prosthesis is to reproduce the mechanical behavior of the innate biological limb. You are tasked with designing a new running prosthesis, that behaves as similarly as possible to the biological leg under the loads associated with running.

1a: Build a mathematical model.

The running leg system can be represented according to the model below.



Here, we've modeled the prosthetic running leg as a beam attached to a rotational spring-damper. The spring-damper is defined by a constant stiffness k, a set point θ_{sp} (the angle at which the spring produces zero torque), and a damping ratio b. A force F representing the ground reaction force (GRF) is applied on the leg at a distance L from the spring-damper's center of rotation. Deflection of the spring is defined by the angle θ, measured from the horizontal. To simplify our work, we will assume that the GRF is always purely vertical. We will also assume that the system is massless, and therefore has no inertial dynamics.

Your first task is to find the differential equation that models this system. **To do this, explicitly work through the five steps we learned in class for generating a mathematical model, and <u>write out your process</u>. Then, classify the equation according to type, order, and linearity. You can do this by hand, then take a picture and insert it into this document. Make sure that your work is readable!**

See next page.

1. Identify independent and dependent variables

   Independent: t

   Dependent: θ

2. Choose units

   t in seconds, θ in degrees

3. Articulate the basic principle

   $\tau = Fr \quad \Sigma\tau = I\alpha$

4. Generate mathematical equations by expressing the principle in terms of variables in step 1.

   $\tau_G = FL\cos\theta \quad \tau_k = -k(\theta - \theta_{sp}) \quad \tau_d = -b\omega$

   $\Sigma\tau = FL\cos\theta - k(\theta - \theta_{sp}) - b\omega = 0$

   $FL\cos\theta = b\frac{d\theta}{dt} + k(\theta - \theta_{sp})$

   $\frac{d\theta}{dt} = \frac{F}{b}L\cos\theta - \frac{k}{b}(\theta - \theta_{sp})$

5. Check for sanity.

   The units match up and the edge cases are intuitive.

Type: Ordinary Differential Equation

Order: 1st

Linearity: Nonlinear

1b: Solve the differential equation.

You'll notice that we haven't yet learned any analytical tools for solving this type of equation (welcome to the real world.) Fortunately, computers exist, and we can leverage computational tools to help us find specific solutions.

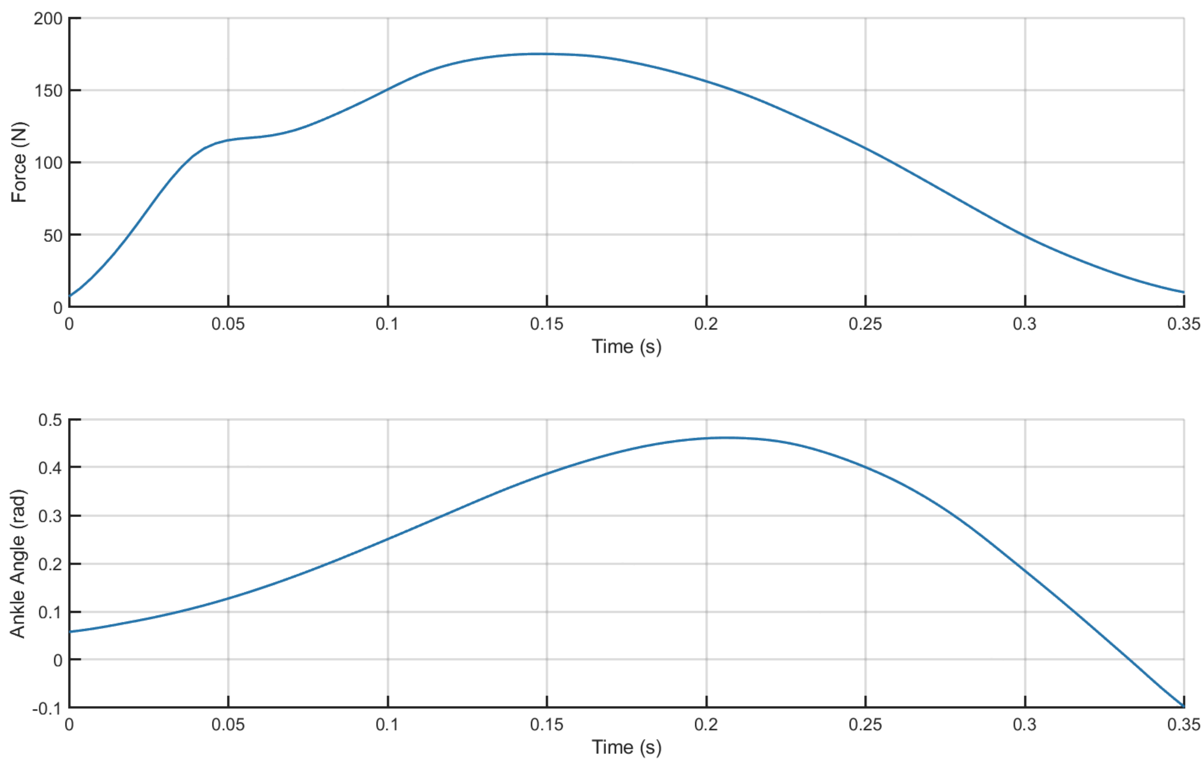
Your goal is to design a prosthesis that closely mimics the human ankle-foot during running. In other words, you need to choose values for k, b, θ_{sp}, and L that result in a deflection trajectory $\theta(t)$ that looks like the deflection trajectory of the biological ankle, when subjected to a load trajectory $F(t)$.

Download Homework1_MatlabProbs.m, passiveProsODEfun.m, and GRFdata.mat. Without changing any of the file names for these three files, put all three files into a single folder. Navigate to that folder in Matlab's "current folder" pane.

Open up Homework1_MatlabProbs.m and passiveProsODEfun.m in MATLAB by double clicking on the file names in the Current Folder pane. You're going to use this code to solve the differential equation you found in 1a for the position trajectory $\theta(t)$. Throughout the code, you'll see double asterisks ("**") – this indicates places where you need to fill in missing code snippets. **Note:** if you run this right away, you will get errors, because the code is not complete. That's where you come in!

Tip: if you use double percent signs ("%%") to split up your code, you can take advantage of the "Run Section" command (or press ctrl-enter) to run only a single section of code. You'll see examples of this in the code already. This is helpful when working on incomplete code, because it prevents MATLAB from throwing errors for the parts that are unfinished.

The first step in this process is to define the force trajectory and the angle trajectory for the biological ankle. I have pulled data from a published journal article on running biomechanics (check it out in the folder with the homework) for a single step, and loaded it in for you. Add some double percent signs to line 34 and run just the top section to see what these data look like. **Save the resultant figure as a .png or .jpg (don't just screenshot it), and insert the image into this document. All figures from MATLAB should be saved and inserted in this way.**



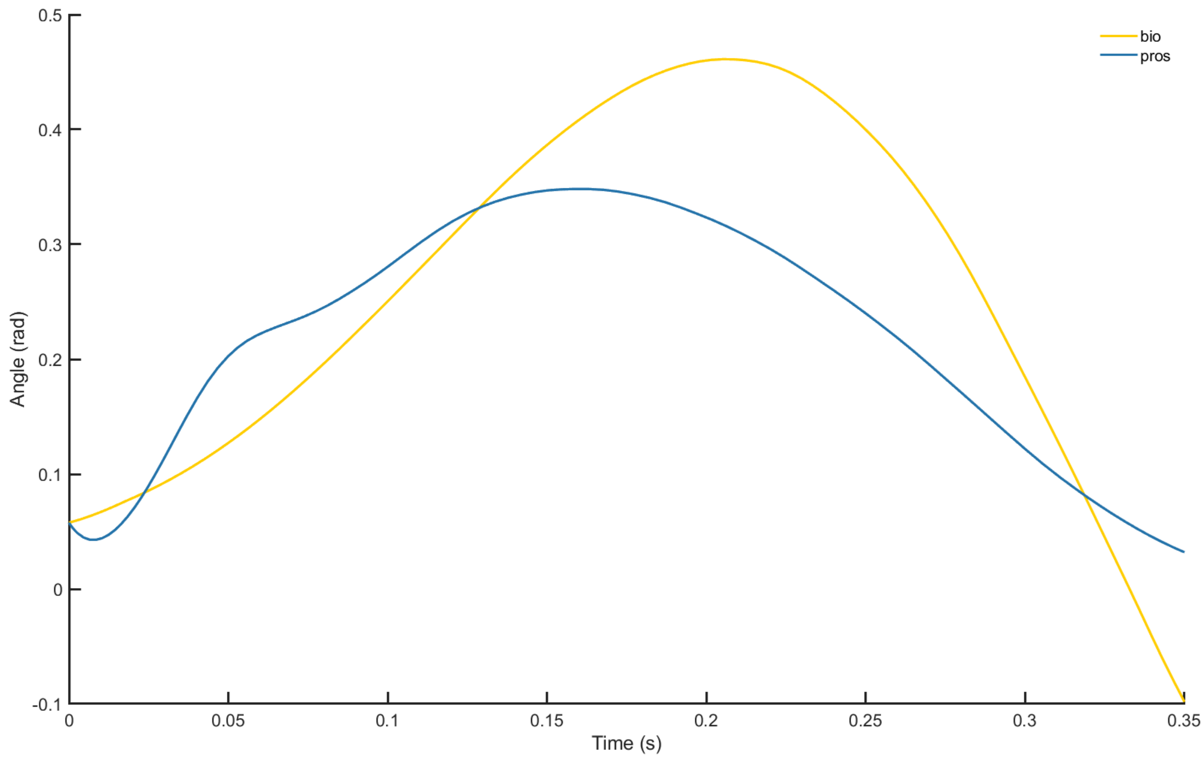
The next step is to define some starting values for the constants, and choose initial conditions. Let's just guess some values, and see how we do. Use the following values for the first pass (starting on line 35).

$$b = 1\frac{Nms}{rad},\ k = 100\frac{Nm}{rad},\ \theta_{sp} = 0\,rad,\ L = 0.2m$$

Follow along in the code as we set initial conditions and give the solver instructions for how to run.

Now comes the magic. Switch over to passiveProsODEfun.m, and fill in the missing code. Before you do this, you should read and understand MATLAB's documentation on ode45: https://www.mathworks.com/help/matlab/ref/ode45.html, The first line of code in passiveProsODEfun.m needs to interpolate into the GRF data to grab the force at each individual timestep. This has to do with ode45's function as variable-step solver (see above-referenced documentation for examples). The next line of code needs to contain the differential equation you derived in 1a. **Hint:** make sure that you solve your differential equation explicitly for the highest-order derivative of theta before filling in this code.

Head back to Homework1_MatlabProbs.m, fill in some code around line 61 to plot your results, and run the whole first section. **Save figure 2 as a .png and insert it into this document.**



Copy-paste your passiveProsODEfun.m code here.

```
function dtheta_dt = passiveProsODEfun(t,theta,b,k,theta_sp,L,GRFvals,GRFtime)

% First, we need to interpolate within our GRF data to get an estimate of

% GRF at the current time point.

F = [interp1(GRFtime,GRFvals,t)]; % **Fill this in with a function that interpolates GRFvals

      % at time t. Hint: look up "interp1".**

dtheta_dt = [F/b*L*cosd(theta)-k/b*(theta-theta_sp)]; %**Fill in your differential equation
from part a. Be sure

      % to use only the variables we've defined in this

      % function: theta, b, k, theta_sp, L and F.
```

Describe what you see in this plot.

The angle decreases slightly at first, but then it increases until at around t=0.15s, where it then decreases.

Does this make it seem like our system is performing well, or not? Why?

Yes, our system seems to be performing well (but could be better) because it follows the shape of the yellow graph. We can adjust some values to make our model better fit data.

Other than being an estimate (we'll talk more about this later on in the course), how is this numerical solution different from the analytical ones we've derived in class? Why might an analytical solution be more valuable?

This numerical solution is different because we made guesses on our constants (b, k, θ_{sp}, and L). An analytical solution might be more valuable because since we made a lot of assumptions, our model is better to be used for identifying patterns and trends. We can use our graph to predict the when and how much the angle will rise/fall by.

1c. Quantify performance.

Before we optimize the parameters governing prosthesis behavior, it is helpful to reduce our evaluation of performance to a single value. In this case, we'll use root mean squared error (RMSE) to compare the solution of our differential equation to the biological ankle angle trajectory. Run section 1c. of the code, and look at the resultant value for RMSEankAng.

What value of RMSE did our first pass generate?

0.0978

What are the units of this RMSE?

Radians

What does this value tell us?

The average amount the modeled angle was off by from the actual angle for each recorded time.

Based on this value of RMSE, do our starting parameter guesses seem good or bad?

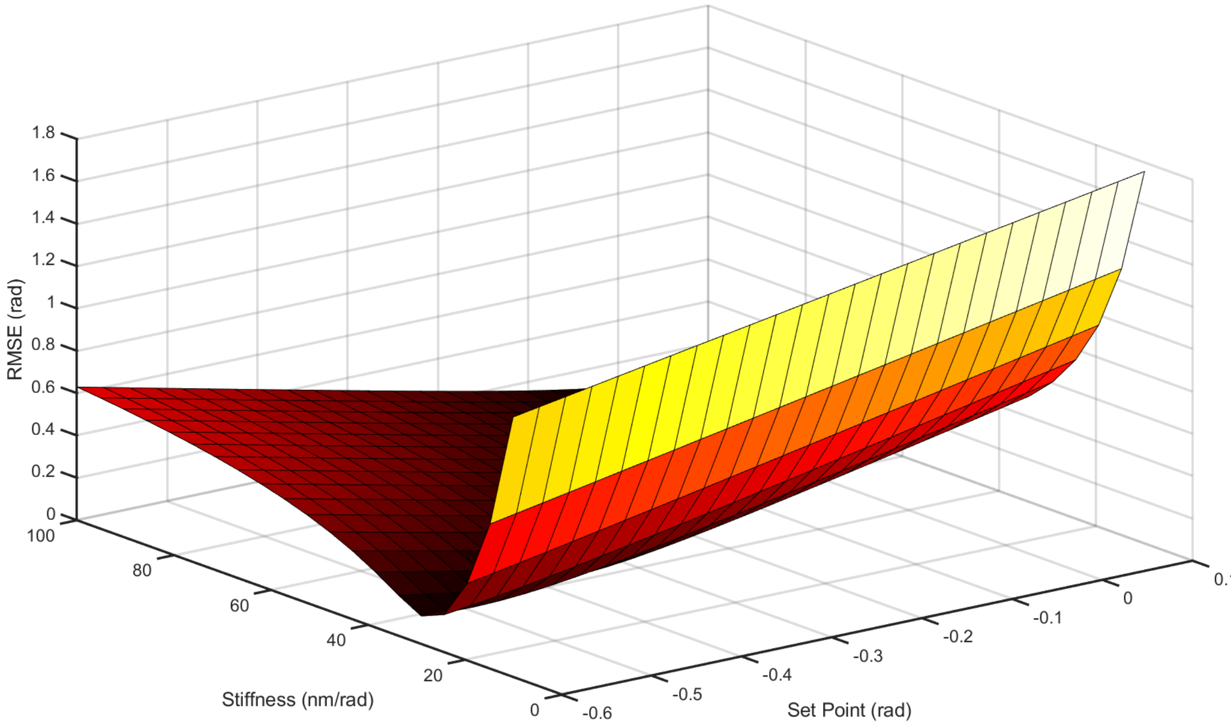
Our starting parameter guesses seem good, because we are off by on average 5.60352724 degrees for each time data recorded. However, we can make our model better.

1d. Optimize the system.

Now comes the fun part. You're going to do a brute-force parameter search to find spring parameters that minimize the error between your prosthetic angle trajectory and the biological ankle angle trajectory, when subjected to the same loads. In our parameter search, we want to try 20 different values of k, spaced evenly between $k = 10\frac{Nm}{rad}$ and $k = 100\frac{Nm}{rad}$. We also want to try 25 values of θ_{sp}, spaced evenly between $\theta_{sp} = -0.6\, rad$ and $\theta_{sp} = 0.1\, rad$. To reduce the dimensionality of this problem, you can leave b and L the same as they were in 1b.

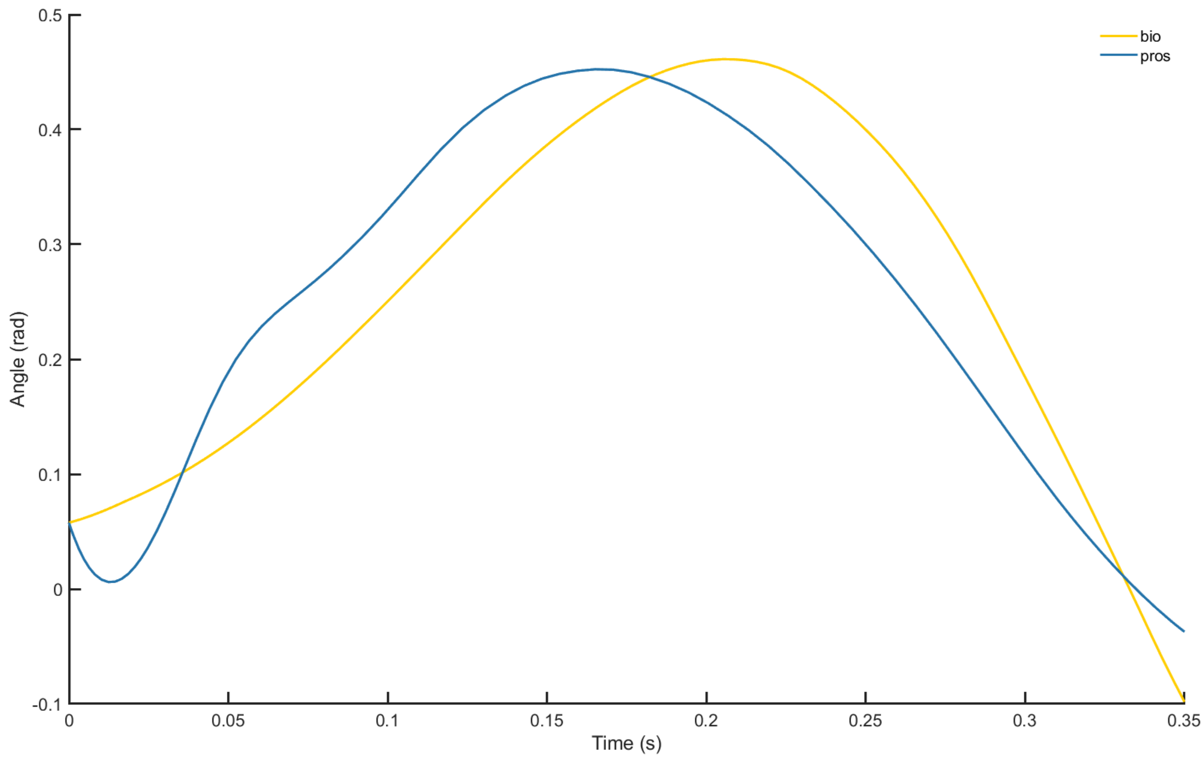
Fill in the missing code in Problem 1d of Homework1_MatlabProbs.m to complete this process. Note: it's worth taking the time to understand each step of this optimization code, because you will likely be asked to reproduce it in the future.

After you fill in the code, you should be able to run the whole script without any errors. **Save the resultant figures 3 and 4 as .png, paste them in below, and answer the following questions.**



Use the rotate 3d tool to look at this surface. What do you notice about its shape? What does this tell you about the relationship between these two parameters and RMSE?

The shape resembles a slightly bent sheet of paper. There is an ultimate minimum for the RMSE somewhere in the graph. As k increases, the RMSE tends to increase. As θ_{sp} increases, RMSE tends to decrease.



What is the minimum RMSE value?

0.0672

What values of k and θ_{sp} produce this minimum RMSE value?

k = 62.105263157894740

θ_{sp} = -0.104166666666667

How does the optimized performance compare to the performance from our initial parameter guesses?

The blue graph follows the yellow graph more closely. The shape is more similar. The RMSE is now 0.0306 less.

Why do you think there is a limit on how well we can reproduce human ankle performance with a spring-damper system? Hint: think about what produces torque at the ankle joint when humans run.

We kept our L and b values the same as in part b. We could potentially improve our model by modifying those values. Also, there are other factors that take into account the angle change that we did not include in our model, such as environmental variability and material behavior.