Homework 08

1. The Matlab file `demowalking.m` demonstrates dynamic walking with active push-off `P` and a torsional spring `Kp` located at the pelvis and acting between the legs. The spring acts to speed the motion of the legs. The code performs a parameter study in which `P` and `Kp` are varied separately, and then plots the speed as a function of collision loss. The code requires and demonstrates the `dynamicWalking` class library (see course GitHub) using Matlab object oriented programming, as well as the publish command.
   A. Use publish to create a report from the file. Note how the command can be used to produce PDF, html, or other output formats. (No need to turn anything in for this step.)
   B. The equation for a simple pendulum under the forces of gravity and a torsional spring is Based on this equation, how do you expect spring stiffness to affect the frequency of pendulum swing?
   C. Examine the Matlab code, and describe how the spring stiffness is applied to the equations of motion.
   D. Based on the parameter study results, explain how increasing push-off affects speed, step length, and step frequency, and how increasing spring stiffness affects the same variables. If each parameter were said to affect mainly step length or step frequency, which would you suggest?
   E. Find the combination of `P` and `Kp` that produces a "normal" human gait of 1.25 m/s with step frequency 1.8 Hz.
   F. Examine your previous experimental results of preferred step length as a function of walking speed. From your simulations, would you say that humans increase speed mainly by pushing off more, by increasing the equivalent stiffness of a pelvis spring, or some combination of the two?
   G. Look at the commands `findgait` and `findgaitspeed`. Write very brief descriptions of what they do. (You can use the class library for your project if it proves relevant. Regardless, you may wish to learn about how object-oriented programming can simplify and organize the simulation studies.)

2. The Matlab file `rununderarmthrowopt.m` demonstrates several variations of the Alexander throwing model with trajectory optimization. Publish the document, using the output format of your choice. This may take several minutes, due to a large number of computations. The file also demonstrates several computations that you will encounter in your project. These include a sanity check to test for energy conservation, a comparison between different models, a parameter study, and different ways to illustrate the model's output. One is a stop-motion cartoon of the model, drawn at intervals of time and space, and the other is an animation that can be viewed on the screen and optionally saved to a movie file.
   A. Examine the file and for each cell, write a one-sentence description of what is performed, as well as a brief summary of the result.
   B. Comment on the difference in maximal throw between the original Alexander model and the best trajectory optimization. How large do you consider the difference, and what do you find interesting about it? Also comment on what which torque model is most helpful for understanding throwing.
   C. Notice that the final optimization (#6) performs similarly to the Alexander model, and less well than the others. What accounts for this difference?

D. A difficulty with optimization over a large number of parameters is the possibility of local minima. One means to test for local minima is to try different initial guesses for the optimization. For one of the optimization methods, try a some different guesses and test their effect on the optimal throw, in terms of the maximum velocity and the torque trajectories (which could be plotted overlaid on top of each other). Comment on the likelihood of local minima.

E. The trajectory optimizations rely on spline functions, each using a number of knot points. Try a different number of knots and comment on that parameter's effect on performance and the trade-off on computation time.

F. A disadvantage of trajectory optimization is that there are a large number of variables to be optimized, and there can be unintended consequences from these degrees of freedom. Comment on how complex a model should be, and the trade-offs involved in making a model more "realistic."

G. Add your own optimization to the file. This could be your own selection of parameter values, spline type, optimization algorithm, constraints, etc. Can you improve on the previous optimizations, preferably without increasing the degrees of freedom?

3. The equations of motion for the underarm and overarm throwing models are in the Mathematica file `AlexanderThrowingModels.nb` . Using that code, try reproducing Alexander's underarm or overarm simulation (your choice). You can use the shotput simulation as a starting point, but there are some differences in the other throws. Then try to reproduce the results plots shown in Alexander's paper. Finally, if you are feeling very confident of your abilities, try using numerical methods to find the optimum delay time for the throw. (You need not match Alexander's results exactly.)