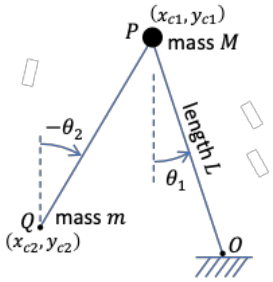
Homework 04

1. Implement the simplest walking model and find a periodic gait. The file `simplestwalkingstep` provides equations of motion, event detection, the step-to-step transition, and even sample animation code.
   A. Produce a simulation similar to the rimless wheel. Use parameter values alpha = 0.3, gamma = 0.03 (3% downward slope), and try initial guess $x = [0.3 - 0.3 - 0.3 - 0.25]$.
   B. Add a test for energy conservation, similar to the rimless wheel.
   C. Using the initial guess, try to find a periodic motion (limit cycle) that includes stance phase ending at heelstrike, and the step-to-step transition to yield the new state.

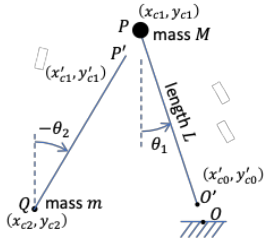
2. Use the embedded constraint method to produce a drop-in replacement `fsimpwalk2Jp` for the equations of motion for the simplest walking model, which are provided in symbolic form in the sample code `testsimpwalk2eom`. The foot O is attached to the ground origin with a hinge joint, and the swing leg also rotates about the pelvis P with a hinge joint. Both legs are of length L, and have orientations measured counter-clockwise with respect to vertical. The corresponding pose Jacobian looks like this:



$$J_p = \begin{bmatrix} -L\cos q_1 & 0 \\ \dots & 0 \\ 1 & 0 \\ \dots & \dots \\ -L\sin q_1 & L\sin q_2 \\ 0 & 1 \end{bmatrix}$$

   The only masses are point masses M and m; the legs have no mass or inertia.
   A. Produce a state-derivative function `fsimpwalk2Jp`, which should have identical inputs and outputs to the sample `fsimpwalk2m`. Demonstrate it by comparing one-step simulations against the symbolic function. Look in `testsimpwalk2eom` for places where you should enter your own code.
   B. Test for your function against the provided `fsimpwalk2m`, as performed in `testsimpwalk2eom`.

3. Use the explicit constraint method to produce a drop-in replacement `fsimpwalk2Jc` for the equations of motion for the simplest walking model. Here the segments should be treated as separate, but then require a constraint Jacobian J_c (also called C in lecture) to enforce the joints. The matrix should look like this:



$$J_c = \begin{bmatrix} 1 & 0 & L\cos q_1 & 0 & 0 & 0 \\ 0 & 1 & \dots & \dots & \dots & \dots \\ -1 & 0 & 0 & 1 & -L\cos q_2 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix}$$

(Note that the centers of mass are located at the ends of the segments, and the constraints describe how the segment ends are connected together.)

   A. Produce a state-derivative function `fsimpwalk2Jc`, which should have identical inputs and outputs to the sample `fsimpwalk2m`. Demonstrate it by comparing one-step simulations against the symbolic function.
   B. When using the constraint Jacobian J_c method, it is helpful to also compute the pose Jacobian J_p. If these two quantities are consistent with each other, the product $J_c \cdot$J_p$ should be zero. Compute the product and verify that this is true. Also explain briefly why this should be the case.
   C. Test for your function against the provided `fsimpwalk2m`, as performed in `testsimpwalk2eom`.

4. The problems above are challenging, and deserve additional clues or clarifications. Please file GitHub issues to ask for hints or corrections.