Legged Robotics, HW 7
Homework due on 03-11-2021, Topic: Learning a controller from simulation. Email solutions to pranav@uic.edu.

1. **Walker with controlled foot placement.**
Model: Figure 1 shows point mass model of walker with massless legs of length ℓ, hip mass M, walking down a ramp of slope γ. Gravity is g and points downwards. The angle between the stance leg and the normal to the ramp is θ. The states of the walker are θ and $\dot{\theta}$ and the control input is the angle between the legs, ϕ. The swing leg dynamics are neglected so that the foot placement angle can be set instantaneously fast.
Control: Assume the Poincaré section is at $\theta_i = 0$ where i is the step number. The Poincaré map may be written as $\dot{\theta}_{i+1} = F(\dot{\theta}_i, \phi_i)$. The goal is to find foot placement angle ϕ_i such that the robot can track a given reference (or desired) velocity at the Poincaré section ϕ_i^{des}.

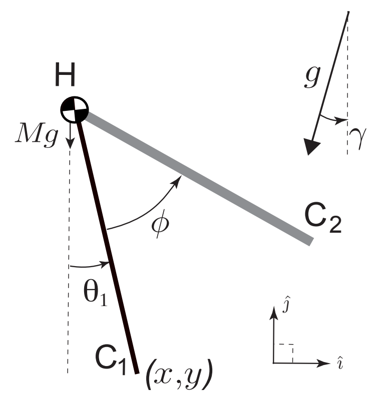

Figure 1: Walker with controllable foot placement angle ϕ

Controller design: You will do the controller design in two stages.

(a) In stage 1, you will use the file *walker_data_generation.m* for data generation. You will input $\dot{\theta}_i$ and ϕ_i which will produce the output $\dot{\theta}_{i+1}$. Note that there will be some inputs that will lead to failure or the output $\dot{\theta}_{i+1}$ will be incorrect. You can monitor these cases by checking the output of *flag*. If it has all zeros, you can trust the output else not.

(b) In stage 2, you will use *walker_main.m* to define the reference (or desired) velocity $\dot{\theta}_{i+1}^{des}$ (see walker.thetadot_des and note that for forward walking these values are negative) and then write a controller in the file *controller.m* which has its input θ_i (see thetadot in controller.m) and $\dot{\theta}_{i+1}^{des}$ (see thetadot_des in controller.m) to produce the output ϕ_i (see phi in controller.m). That is $\phi_i = f(\dot{\theta}_i, \dot{\theta}_{i+1}^{des})$. In particular you are going to follow two approaches: (1) a look-up table (2) an explicit function f such as a polynomial, neural network, gaussian process regression.

Compare the results you get with respect to velocity tracking for both approaches.