ME 193B / 292B: Feedback Control of Legged Robots
HW #6

Problem 1. Event-based Controller

This problem follows on the 3-link walking controller problem from HW #5. We will now design an additional controller around the IO linearizing controller we designed in HW #5. This *event-based controller* will change the desired torso angle θ_3^d at the beginning of every walking step to regulate the average hip velocity. Note that moving the torso forwards (increasing θ_3) causes the center of gravity of the robot to also lean forwards further, which in turn causes the robot to move faster.

The average hip velocity v_{avg} is defined as

$$v_{avg} = \frac{pH_{k+1}^x - pH_k^x}{t_{k+1} - t_k}, \tag{1}$$

where pH_k^x and pH_{k+1}^x are the x-position of the hip at the start of the k^{th} and $k+1^{th}$ step respectively, and similarly, t_{k+1} and t_k are the times at start of the k^{th} and $k+1^{th}$ step respectively. The control structure is illustrated in Figure 1. The event-based controller Π outputs a desired torso angle θ_3^d at the start of every walking step based on the current average hip velocity and desired average velocity at the start of that step. This desired torso angle obtained from the event-based controller Π is then kept constant throughout that walking step and enforced through the IO-Linearizing controller you developed earlier.

Design a simple linear feedback controller for Π to regulate the average hip velocity to 0.5m/s from the initial conditions provided in HW #5 (i.e. $v_{avg}^d = 0.5$). (Note: The controller can take multiple walking steps to converge to the desired average hip velocity). Provide a plot of the average hip velocity vs the number of steps and check for the friction constraints as you did in HW #5. Examples of simple linear feedback controllers include PD, PI, PID, etc. NOTE: You do not have to compute or use the Poincaré map to achieve this.

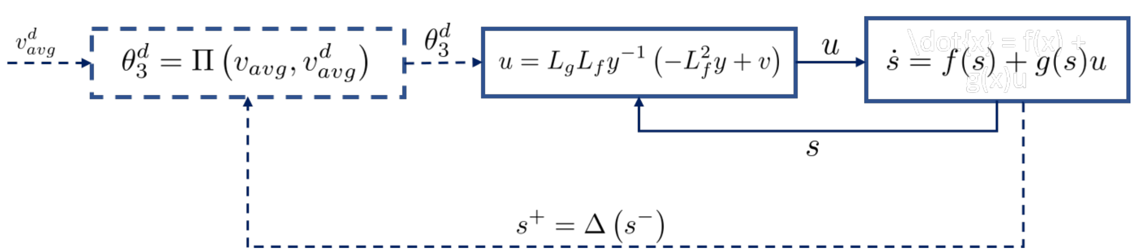

Figure 1: Event-based controller to regulate average hip velocity. Dashed lines represent discrete events taking place at the beginning of a walking step. Refer to HW #5 for definition of various symbols such as $s, f(s), g(s)$, etc.

Problem 2. Raibert Control (This problem is due to Prof. Chris Atkeson at CMU)

In this problem, you will implement a Raibert controller to enable a one legged hopping robot to hop at a particular height and particular speed. We do this through the following:

(a) **Height Control:** Download the Matlab code `part2a.zip`. This code has been designed to allow you to focus on designing control of hopping height by only allowing vertical movement. Execute `main.m` in the given Matlab code to see the hopper.

To enable the hopper to hop at a desired hopping height (approximately), modify the function `control.m` appropriately at the point where it says:
`rest_leg_length = leg_length_default;`
You can set the desired hopping height to values between 0.6 and 1.0 and have the hopping height vary accordingly. To do this, change `main.m` at the line:
`height_desired = 0.6;`
Submit two plots of multiple steps of hopping showing convergence to a hopping height of 0.6 and 1.0 respectively. (You only need to modify one line in `control.m` and one line in `main.m` for this problem.)

(b) **Leg Control:** Download the Matlab code `part2b.zip`. This code has been designed to allow you to focus on designing the control of the leg when it is in the air. The function `simulate.m` has been modified to keep the hopper fixed in the air. You need to modify the gains in the function `control.m`:
`hip_air_k = 10;`
`hip_air_b = 1;`
Modify these gains so that the leg swings quickly without oscillation to the `leg_angle_desired` target in less than 0.3s. Submit a plot illustrating this.

(c) **Full Control:** Download the Matlab code `part2c.zip` that has the full simulator. You need to modify the `control.m` function based on (a) and (b) above that does height control and leg control. You also need to add body angle control (setting the gains `hip_grnd_k` and `hip_grnd_b`. You also need to figure out how to control forward speed by choosing the leg angle at touchdown. You should be able to set the desired speed in `main.m` by changing:
`speed_desired = 1.0;`
Change this between 0 and 1 and have the forward speed vary accordingly. Submit two plots showing convergence of hopping speed to two different speeds that you choose.

Instructions

1. You may submit either a typeset or handwritten solution. In either case, submit a **PDF** version of your solutions on bCourses, with the naming convention: `firstName_lastName_HW6.pdf`.
2. Start each problem on a separate page.
3. You may choose to use a symbolic math package such as the `Symbolic Math Toolbox` (`https://www.mathworks.com/help/symbolic/index.html`) in `MATLAB` or `Mathematica`.
4. Do include all your code, if any.
5. Please submit a single pdf of your HW. (If typset on a computer, please save to pdf. If handwritten, please scan to pdf.)
6. **Honor Code.** You are to do your own work. Discussing the homework with a friend is fine. Sharing results or MATLAB code is not.