ECE 8 Robotics Notes 10-25-22

Lecture 10

- Introducing : **<u>proportional control – Not Included in midterm</u>**
- a variation of the pervious class (see 10-20-22)

Example 7 : Cruise Control System

- See professors Handwritten notes:
  - Find the diagram of the car approaching a hill, this is where lecture begins
  - v_k
    - Current velocity "speed"
    - Is labeled on the car with an arrow to the right
  - We want to know:
    - Given desired velocity v_d --design an algorithm that guarantees that the vehicle velocity v_k converges to v_d even in the presence of a terrain change. In particular see diagram, where the hill is the terrain change.
      - Converge: we get as close to the desired value as possible (it may never actually reach it, but it can be 0.00000000000000001 away or closer but never zero.
    - The incline
      - Initially the incline is zero
      - Going up the hill we have a positive incline
      - Going down the hill we have a negative incline
    - In short:
      - Given a target v_d
      - Make a program/model that takes v_k "current position" and have it become $v_k \approx v_d$
      - i.e., "where were away from the door, but my model helped me get to the door"
- To solve this problem:
  - FIND THE MODEL+ALGORITHM! That helps the car move over the hill!
  - Solution:
    - We design an algorithm that measures velocity v_k at its current position
    - We adjust the "gas pedal or the brake pedal" appropriately to attain
      - $v_k \approx v_d$ "v_k converges to v_d i.e., gets as close as possible"
    - A model is given in a later diagram
    - MATLAB code is provided as the first steps to solving this problem

- Prof. Question: Guess what would be placed in our closed loop system diagram
    - Answer: professor redraws the diagram "velocity model"
    - See Diagram: the following are equations of interest
    - Velocity model has inputs and outputs:
        - Input: Desired speed: v_d
            - input because we are trying to tell the car to go somewhere we desire
        - output: Current speed: v_k
            - output because this is what the car does after we tell it how fast to go
    - Error equations:
        - $e = v_d - v_k$
        - error is equal to the speed we want less the current speed
            - e.g.: if we want to go 5 mph and we are currently at 2mph then
                - error = 5 mph – 2 mph = 3 mph
                - we are 3 mph off from what we should be
    - Recall the Equations of motion/Newton's equations
        - $\frac{dp_x}{dt} = v_x$
        - "the derivative of position is equal to the speed"
        - this is a dynamic equation for continuous time t
        - To get this equation into the discrete time k we use the Forward Euler Model:
            - Forward Euler always has a recursive $k, k+1$ variable
                - $v_{x,k+1} = v_k + \Delta\frac{(-bv_k + h_k + u_k)}{m}$
                - this is the discrete version of the previous Newton model
- Proportional control Model:
    - $k(v_d - v_k) + u^*$
    - This equation is in the red box, in the professors diagram
        - u^*= force input that controls the speed
            - we design this variable such that we have a steady state input for a fixed input
            - meaning we have a constant force moving the vehicle
- It is important to know how to jump between continuous and discrete systems

Velocity model

- the model we will use is obtained by modeling the change of velocity (Δv_x) as a function of several parameters playing a role in the problem set up
- in this case we will use the following discrete time model:
    - $v_{x,k+1} = v_k + \Delta \frac{(-bv_k + h_k + u_k)}{m}$

where,

- v_k is the forward velocity "speed" "current"
- Δ step size
- h_k represents the incline of the road
    - so if h_k = 0 the road is flat
    - if h_k > 0, is positive, then we are in downhill
        - downhill means v_x is positive moving to the right
        - so if $b = 0$ and $u_k = 0$ then h_k should be positive ??
    - if h_k < 0,is negative, then we are uphill
        - v_x is negative? because its slowing down as it moves uphill?
        - so the incline is negative?? the prof made no sense here to-be-honest
- $m =$ mass of the vehicle
- u_k is the force i.e. the control
- if $u_k > 0$ we are increasing the speed
    - gas pedal is pressed
- if $u_k < 0$ we are decreasing the speed
    - brake pedal is pressed
- if $u_k = 0$ then no effect on speed
- $-bv_k$ some sort of resistance
    - the value $-b$ will remove speed until the vehicle stops moving, unless we have a stronger force that overcomsit
    - for now, we will assume b is a constant
    - friction
        - e.g.:
            - contact with ground
            - air affects
            - water
            - grit from the road
- Recall from Physics: Kinetic Energy
    - $K = \frac{1}{2} mv^2$

Handwritten Notes

- professor completes the closed loop diagram by plugging in the velocity model we wrote above
    - we can only control the u_k so we call it the control input
    - disturbance variables: h_k incline
- this is shown as an input but it does not loop, it's only in (not out)

MATLAB

- See files from 10/25/22 for full examples of using the previous models

Control system

- To achieve convergence of $v_k \; to \; v_d$ we need to properly design u_k.
    - One way is through proportional control (P control)
    - the structure is the given equations in the newest closed loop model – the professor has drawn a diagram representing the models
- Namely, the input u_k is assigned as follows:
-

➢ $u_k = k(v_d - v_k) + u^*$: Givens(k, v_d, v_k) We **NEED** to determine u^*

- k : control gain
- $v_d - v_k$ is the error
- u^* is the constant that induces the convergence $v_k = v_d$
    - To design u^*, notice that if $v_0 = v_d$, where v_0 is our initial velocity then we have,
    - (all equations after this is for the case: if $v_0 = v_d$

➢ $v_{k+1} = v_k + \Delta \frac{(-bv_k + u_k)}{m}$

- where $h_k = 0$ we assigned to be constant for all values of k
- Then for the $k = 0$ case: (plug in $k = 0$)

➢ $v_1 = v_d + \Delta \frac{(-bv_d + u_0)}{m}$

- then we consider $u_0 = k(v_d - v_0) + u^*$ since $v_d = v_0$ then $u_0 = u^*$
- We can then change the equation for v_1

➢ $v_1 = v_d + \Delta \frac{(-bv_d + u^*)}{m}$

- Now, if we let $v_1 = v_d$ to be true we change the previous equation again
    ➢ $v_d = v_d + \Delta \frac{(-bv_d + u^*)}{m}$
- We WANT: $v_d = v_d$ so we NEED the 2nd term to be zero, since Δ is never zero we have,
    ➢ $\frac{(-bv_d + u^*)}{m} = 0$ or If we solve for our control input u^*
    ➢ $u^* = bv_d$!!! The correct choice for our u^*

End of day Announcements

Thursday: Review for 1hour, recorded

End of Thur, Quiz on previous lecture

- familiarize the closed loop diagram

  Tuesday: Midterm, just bring a pen/pencil

  Review: lectures, be capable to write code, discrete to continous models, trajectory, inputs, sensors, quizes

  Friday: