ECE 8 Robotics Notes 10-25-22

# Lecture 10

* Introducing : **proportional control**
* 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
  + - Current velocity “speed”
    - Is labeled on the car with an arrow to the right
  + We want to know:
    - Given desired velocity --design an algorithm that guarantees that the vehicle velocity converges to 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
      * Make a program/model that takes “current position” and have it become
      * 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 at its current position
    - We adjust the “gas pedal or the brake pedal” appropriately to attain
      * “ converges to 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 out closed loop system diagram
* A: professor redraws the diagram "velocity model"
* loop: in == velocity model == out
* in : desired speed $v\_d$

- out: current sppeed $v\_k$

- loop out $v\_k$ into in $v\_d$ by using error equations

- $e = v\_d -v\_k$

- in the velocity model:

- recall: $\frac{dp\_x}{dt} = v\_x$

- use forward Euler to create the model

- see next notes to find this

- Red box: PROPORTIONAL CONTROL

- $k(v\_d - v\_k) + u\*$

- wher $u\*$ is an input

- design $u\*$ such that we have the steady state input or fixed input

- the control algorithm will only give a constant input

- Learn how to jump between forward Euler and continuous time model for any system

## Velocity model

- the model we will use is obtained by modeling the change of velocity 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

> $$ v\\_{k+1} = v\_k + \Delta (\frac{-bv\_k +h\_k + u\_k}{m}) $$

where,

- $v\_k$ is the forward velocity "speed" "current"

- $\Delta$ 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 downhil

- if $h\_k<0$ is negative, then we are uphill

- $m$ mass of the vehicle

- $u\_k$ is the force i.e. the control

- if $u\_k>0$ we are increasing the speed

- gass pedal is pressed

- if $u\_k<0$ we are decreasing the speed

- break 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

- Kinetic Energy

- $K = \frac{1}{2}mv^2$

## Handwritten Notes

- professor completes the closed loop diagram by pluggin in the velocity model

- 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)

- Stage one:

- code the work done so far

## Matlab

- simulate the previous system for

- $\Delta = 0.01$

- $b$ = 0.5

- m = 20

- $h\_k = 0 $ for all k

- $u\_k = 0 $ for all k

- $v\_0 = 10$

- $v\_d = 56$

- plot $v\_k$ vs $t$

- for 10 seconds

- horizontal is 0, $\Delta$, $2\Delta$, ... 10 $\Delta$

- we obtained $v\_k$ decreases to zero as expected

- Now, let's make the vehicle reach $v\_d$

## 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 eqn.s in the newest closed loop model the prof has drawn

- Namely, the input $u\_k$ is assigned as follows:

> $$u\_k = k (v\_d - v\_k) + u\*$$

- k : control gain

- $v\_d- v\_k$ is the error

- $u\*$ is the constant that induces $v\_k = v\_d$

- To design $u\*$, notice that if $v\_0 = v\_d$ then we have

> $$v\_{k+1} = v\_k + \Delta (\frac{-bv\_k + u\_k}{m})$$

- where $h\_k = 0$ for all k

- Then for the k=0 case

> $$ v\_1 = v\_d + \Delta (\frac{-bv\_d + u\_0}{m})$$

- if $u\_0 = k (v\_d - v\_0) + u\* = u\*$

> $$v\_1 = v\_d + \Delta(\frac{-bv\_d + u\*}{m})$$

- if we want v\_1 = v\_d then

- vd = vd + delta(-bvd + u\\*/m)

- we want -bvd + u\\*/m = 0

- u\\* = bvd

- the correct choice

- To design k, for now, try k>0 and k<0 to achieve convergence to vd

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: