ECE 8 Robotics Notes 10-13-22

prof. = professor

- professor recants that the previous lecture was just a "look into the future" not required material
- recall:
    - Discrete Time models – the topic of today
- Prof. notes:
    - one week ago we had notes on discrete time models
    - see equations of position, velocity, delta Δ
- Question given:
    - why is it discrete time? or what makes it discrete?
        - because it is expressed in k
            - and k is countable $1,2,3,4$, and so on
    - In contrast : continuous time t
        - we can express t as 1, 1.01, 1.02, etc. (it can be in decimals)
- professor referenced a graph from last week
    - the graph shows an example of discrete time steps (i.e., dots along the line)
    - see last weeks notes

Today: Trajectories to discrete time model

- Our first discrete time model
  - is given by the evolution of position p_x at discrete times that is associated to the brick on the ground

$$\frac{dp_x}{dt} = v_x \rightarrow p_{\{x,k+1\}} = p_{\{x,k\}} + \Delta v_{\{x,k\}}$$

  - where:
    position at $k+1$ "new position" = position at discrete time k + step size * velocity at discrete time k

- Q: what is needed to calculate the rest of our positions i.e. the ones we generate($p_{\{x,k+1\}}$)?
  - we need the starting position, where is it located?
  - we need the starting velocity, how fast is it moving and in what direction?
  - the step size, or in general the full model “equation”
    - how does current position given the previous position evolve, what equation shows the relationship?
    - A: It is given by some physics law i.e. model/equation
- summary of previous question:
  - We Need:
    - Initial conditions/parameters (temperature, position, etc.)
      - e.g.:
      - the brick starts at the origin $(x, y) = (0,0)$
      - and it is not moving so $(v_x, v_y) = (0,0)$ these are the initial conditions of the brick
    - velocity (the input from some force)
      - sometimes the force only happens at the beginning but sometimes the force can be constant and affect the speed of the object.
        - e.g.: Falling objects experience the constant force of gravity
    - model of the system
      - equations
        - e.g.: The spring force equation, the equation for a moving pendulum, (Equations of motion)
    - **this solves the evolution of the key variable as it moves forward in time**

Prof noted:

- Today we practice 1D motion but we will eventually move up to 2D, 3D motion

Hand written notes

- for this discrete-time model we can compute the values of $p_{x1}, p_{x2}, \ldots$ and so on, but only if we know
  - p_{x0} what we call "the initial position"
  - $v_{x0}, v_{x1}, \ldots$ and so on (the velocity input from a force)
  - the **relationship** between key variables p_x and inputv_x (in this case), and this is what we call the **model**.

- General robotic systems (not just position)
  - we can treat a robotics system (robot) as a system with inputs and outputs
  - Prof. draws:
    - a square (could be any shape but square is convention)
    - the square has arrows pointing in and out with the box, such that the box is representing the model
    - The box:
      - contains an initial condition
      - for the case of p_x:
        - inputs are $v_{x0}, v_{x1}, \ldots$
        - the model is $p_{\{x,k+1\}} = p_{\{x,k\}} + \Delta v_{\{x,k\}}$
        - the output is $p_{\{x,k+1\}}$
        - given the initial condition p_{x0} can get $p_{x1}, p_{x2}, \ldots$ and so on
        - see MATLAB code from lect. 5 for formal example of this
          - we also use it in the following examples in class
- How does this relate to Quadrotor (quadcopter)
  - the model for quadrotor depends on thrust, the user, the computer board that generates the voltages
    - basically a bunch of little things are in the inputs
  - **Overall: we create the input** and the **motion** of the quadrotor **is the output**

Prof. Returns to MATLAB

- professor hints at using loops to generate many repetitive steps
  - Code: for loops, while loops
    - If cannot recall how to use loops consider using the help command:
      - command: help for
        - examples can be found here
    - Example of for loop:
      - for loop : is a program structure that allows you to repeat a desired code a finite amount of times
        - **for k = 1:10**

          disp(k)

          end
        - this will display **1,2,3,4,5,6,7,8,9,10**

NOTE: the code for the MATLAB example will be posted in canvas

Errors

- professor emphasized when you run a code if you get a **red error** in the terminal
    - read it:
        - it gives the **file with the issue** and **the line where the code is giving a problem**

Code: file: PositionDiscreteTime2.m

- We spent time attempting to create a for loop and generate a graph
    - the professor adds the code to the file and states he will keep the convention of the notes
    - We spent the remainder of the class just trouble shooting MATLAB loops
    - see professors files for better examples and completed code
        - to better understand any loops see the lab that was posted with tutorials
        - or google:
            - MATLAB for loop
            - MATLAB while loop
            - these should give better examples if ever stuck

TUESDAY

- Today we did 1D (1 dimension) p_x next week we will go up to 2D and then 3D
    - this means we will consider p_y, p_z
- Professors Question:
    - Can we plot all of them?
        - we will try this next week it's not required but it's suggested to attempt it over the weekend
        - Recall: we can see everything up to 3 dimensions, so we should be able to graph the new variables next week
- Next Week:
    - We will learn how to design our velocity and position vectors so that we can create the control algorithm "the algorithm that tells the robot to go to a desired location, or do a desired action"
- Quiz on Tuesday
    - see announcement for more information