Assignment 3: Fire control (ring buffers, plot updating)

https://drive.google.com/drive/folders/1Ps9knVaecdyNUfCVwXRI3SSSv5gl1va-?usp=sharing — Use this Google Drive link to access the provided files.

Due:

- m-file due by 11:00 AM U.S. Mountain Standard Time, Friday May 14, 2021

Background:

For this assignment, you will write a program that decides when a simulated weapon is to fire based on a history of the orientation and position of a moving target, which I imagine as a quad-rotor drone. Your task comes down to writing the contents of a function called `fire_decision.m`, which is called by the main program every 0.1s with updated target information. I have provided a "template" version of this function. You will write Matlab code in this function that accomplishes the tasks detailed below. You can run the entire assignment, using your own "smart" decision software, by typing

```
  fire_control;
```

in the command window, and compare the "dumb" fire control decision (which just blindly fires every 10 seconds) by typing

```
  fire_control('dumb');
```

Other invocations that may be useful for debugging:

```
  % Arguments: smart/dumb, random "seed", speedup factor
  fire_control('smart',200,1); % same as fire_control;
  fire_control('dumb', 200,1); % same as fire_control('dumb');
  fire_control('smart',200,3); % 'smart' decision, triple speed
  fire_control('dumb' ,200,2); % 'dumb'  decision, double speed
```

If your program is successful, it will perform better than the 'dumb' decision.

You will write a Matlab program that will do the following:

- Store the newest target orientation in a ring buffer, which I have initialized for you.
- From the ring buffer, extract an array of the most recent 20 orientations in order.
- Smoothly update figure 2 with the time-ordered target orientation data.
- Your program will instruct the gun to fire only under the following conditions:
  - The target is at a range of less than 500 meters.
  - The target has just made a sharp turn (oldest 3 orientations have a range greater than 60 degrees), and now follows a mostly-linear course (newest 17 orientations have a range of less than 5 degrees).

You MUST follow the requirements below to make automated grading possible:[*]

- Use the provided library functions demonstrated in class.
- Don't add any functions requiring a mouse click, key press, or any other user input.
- Don't add any figures or graphics other than what was provided.
- Use Matlab's default fonts, colors, linestyles, and all other plot properties.

[*] *Without requiring the invention of Generalized Artificial Intelligence for this course, which I admit I am working on, but doubt I can solve this semester.*