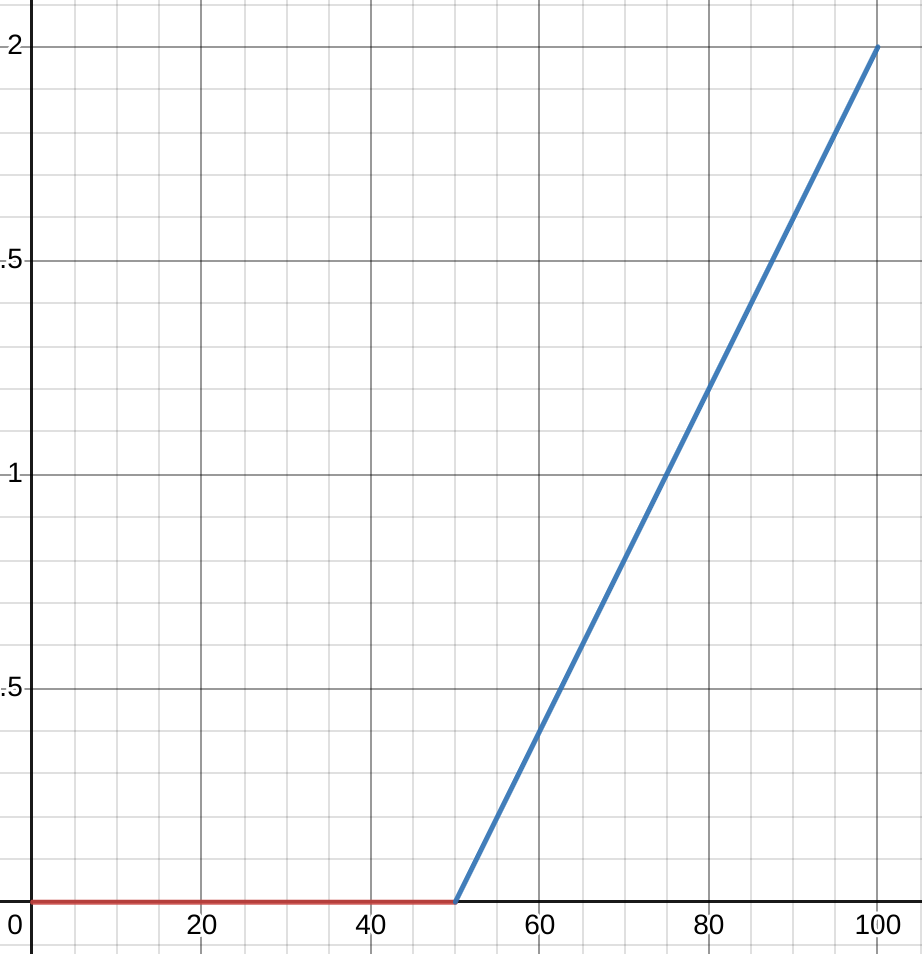
* 1. B
  2. A
  3. C? - just the diagonal elements would be non-zero
  4. G
  5. D

1. 1. 1. 
      2. By using a linear neural network, the robot will act upon the assumption that the true dynamics of the environment is a linear combination of its x and y position, which is clearly not the case. The linear function will overestimate the change in x in the valley and underestimate the change in x above the valley (where there is wind). If the drone explores just the valley, it will assume that there is no wind and rise to (75, 50), then stop. Otherwise, the higher the drone explores in the y direction, the more likely it will end up to the left of the target, due to it fitting a linear function that is more likely to overestimate the change in x inside the valley. For an interactive proof, see: <https://www.desmos.com/calculator/ebjnuwqrxf> (k represents the gradient of the linear model).
      3. Model predictive control works by continuously planning for the next steps and executing just one single step, then starting planning again. This will allow it to compensate for the fact that the predicted deltaX will be larger than the true value. Ultimately, the drone will reach the position at (125, 60), and will likely cut out its engine 10 units above the landing target.
      4. 1. DAgger may not be practical as it requires a large number of demonstrations.
         2. It also requires a robot to stop halfway through a task, which may not be practical in some situations
      5. C. Like most imitation learning algorithms, the robot will learn only to imitate the action of the provided demonstrations, not use predictions or heuristics to calculate the best path forward. Since the starting points sit directly in the middle of the demonstrated areas, the robot will be equally likely to go left or right. If the robot is still in training mode, it will likely request another demonstration, as it is uncertain about the next action.