Advanced AI Assignment 2: Kalman Filter

This folder contains the problem statement and the basic code for the assignment. You must attempt this assignment only in python. You are provided with minimum boilerplate code to begin with, and you need to fill up the missing code such that all conditions mentioned in the question are met.

Problem Statement

In an area near the border, you are required to monitor any enemy aircraft that approaches near the vicinity. The RADAR you have can scan a large area, and takes a certain time T to make a full rotation while scanning. You and your team is working on an Autonomous Surface-to-Air Missile-launcher (ASAM) that can fire missiles at the target. It takes input from the RADAR, and it is designed to lock on and fire at a given location as the target enters a specific range. For this purpose, the ASAM must continuously monitor the target. Given the clock speed of the processor, the ASAM takes Δt time to update the target. However, $\Delta t < T$, meaning the the ASAM must continuously update the target's position in multiple time ticks between any two readings of the RADAR. Use the Kalman filter to update the target's state.

The enemy aircraft(target) can be modelled as a discrete dynamic system. For the missiles to lock on, the state of the target must be known with high accuracy. In this question, the state of the target at a given time t is S_t, and is given by

$$S_t = \begin{bmatrix} v \\ x \end{bmatrix}$$

where v is the 3-D velocity vector at the time instant, and x is similarly the 3-D position vector at that given time instant.

With some hacking devices, the acceleration control $u_t = a_t$ acting on the target is also known, which is a 3-D acceleration vector at that given time instant. Needless to say, the equations to deduce the the next corresponding values of the state are given by

$$v_{t+1} - v_t = a_t \times \Delta t$$
$$x_{t+1} - x_t = v_t \times \Delta t$$

Assume for sake of simplicity, that the time taken between each time tick for the ASAM's processor is the same Δt , that is, the value of Δt can be taken as 1.

All RADAR readings is subject to a Gaussian noise η, which is known. Assume for this assignment that the measurements of position, velocity and acceleration are independent and therefore there is no co-variance in error for the observed readings. Also, given that $\Delta t < T$, take

$$k \times \Delta t = T$$

where k is a positive integer constant. This implies that after obtaining a reading from the RADAR, the ASAM needs to predict its target's location exactly k times before it receives another reading.

Part 1

Write the equations for the Kalman filter that would be necessary in this case. Clearly mention all assumptions, and give an example(taking any values of a_t, and initial state $[v_0, x_0]$) and show the states evaluated in 5 such iterations. Explain how the ASAM will update multiple times before every two readings from the RADAR. Write this in a PDF report.

Part 2

Complete the code as per the planned equations. View the error between the observations and the predictions that is generated by the code, and list it in the report as well. The code to be completed is in the file `kalman.py`. Complete the code, and run the python file `tests.py` and mention the obtained output in the PDF report.

Part 3

Consider the case of a falling projectile. It is noted that to have a velocity v_0 at a given position x_0. The only acceleration force acting on this body is the force of gravity, but due to reaching its terminal velocity (a velocity where the air resistance cancels out all downwards acceleration), the total acceleration of the body is 0. Assuming that it will stay that way, rewrite the Kalman filter equations where the state of the object is observed from the RADAR. You can choose to change the structure of the previous state equations. Write these equations in the PDF report, and point out the differences compared to the equations in the first part.

General Guidelines for the assignment

- This assignment is to be done in python. No dependency other than the `numpy` library should be required.
- The sections **Part I, Part III** are to be done in the in the report. The section **Part II** is to be done in the file `kalman.py`, and the output obtained from running `tests.py` is to be included in the report as well.
- Complete the given source code file, and upload it along with the report. (Do not include any additional folders or compress in a the zip file). Upload the following on google classroom

```
├── kalman.py
└── Report.pdf
```

- For evaluation of your assignment, your code will be tested with additional hidden test statements. They will be of the same format as the test cases provided to you. So ensure that the provided test cases are passing in your machine.
- Do not edit the file `tests.py`. Do not add any other file than the ones already present in the boilerplate code.
- Needless to say, you should also not edit the defined name of the `KalmanFilter` class, or the function names. Doing so will ensure that your code will fail the hidden test cases once you submit your code.

- The hidden tests will only use valid types, so there is no need for error handling within the code. There is also no need for any performance optimisation. Your code is expected to return the correct answer for queries on the functions that you need to complete, and you will be graded accordingly.
- All students are expected to maintain their integrity and do the assignment on their own. Any cases of cheating/copying will be awarded 0 marks.
