Durham University

Assignment

Contemporary Computer Science - Virtual and Augmented Reality

This assignment should be submitted via DUO. The deadline is 2pm on Friday March 15 (Week 19). Submit Python or Matlab source code in addition to a short report in .pdf format (not longer than 3 pages including images). Please ensure that your source code has an introductory comment that describes how to run it if special care is needed. ***L3 students need to implement problems 1-5 (total 90 marks). L4 students need to also implement the additional problem 6 (30 more marks, total 120).*** The marks available for correct implementations/answers to each question are indicated. Partial credit will be given for good attempts. Your marked work will be returned no later than May 10th 2019. Clearly state which version of Python/Matlab you have used used. Do not use existing libraries for sensor fusion. You can use any library you want for data reading, visualization, etc. (numpy, matplotlib, ...).

The year is 2021. The VR revolution has finally happened. Thousands of new VR companies are being established each year. You have just been hired by a Virtual Reality headset manufacturer start-up in the North East. The company has several teams working on research and development (R&D). You are part of the team that is developing the headset's orientation tracker. Your team's mission is to provide the most comfortable experience to their users, i.e., provide reliable tracking.

The company has only received limited start-up funding and is on a very tight budget. As such, the developed headset will have an incredibly cheap Inertial Measurement Unit (IMU). This unit, suffers from gyroscope drift and low measurement precision. Making matters worse, the unit does not have an embedded firmware that combines (fuses) the tri-axial gyroscope, accelerometer and magnetometer readings to calculate a motion vector. As such, the current device driver only provides raw gyroscope, accelerometer and magnetometer readings to the game engine.

You have been tasked with developing a prototype IMU firmware emulator (a mockup) in Python or Matlab to investigate the performance of a few sensor fusion algorithms that you found in an academic paper [1] (provided in DUO coursework folder). If you succeed, your implementation will be embedded in the device driver of the headset and you will get a promotion.

PROBLEM 1 - 20 MARKS:

The hardware integration team provided a sample dataset (available in DUO coursework folder). The dataset (6959 records) is acquired from the IMU that is sequentially rotated from 0 degrees, to +90 degrees and then to -90 degrees around the X, Y and Z axes. IMU observations were recorded at a rate of 256Hz: Time in seconds, Tri-axial velocity (rotational rate) in deg/s, tri-axial acceleration in g (m/s^2), and tri-axial magnetometer flux readings in Gauss (G). Read and import the provided (.csv) dataset (5 marks). Convert rotational rate to radians/sec and normalize the magnitude of both the accelerometer and magnetometer values taking special care of NaN divisions (5 marks). Implement methods to (i) convert Euler angle readings (radians) to quaternions (2 marks), (ii) calculate Euler angles from a quaternion (2 marks), (iii) convert a quaternion to its conjugate (inverse rotation) (2 marks), and

(iv) calculate the quaternion product of quaternion a and b (2 marks). Provide a couple of sentences in the report for the function interfaces that you created (what is the input/output) (2 marks).

Resources
Chapter 2 in [2] (provided in DUO coursework folder)
Section III in [1] (provided in DUO coursework folder)
Chapter 3.3 in LaValle's book (the module's coursebook)
Wikipedia link

PROBLEM 2 - 10 MARKS:

One of your colleagues believes that estimating orientation from the gyroscope (rotational rate) is a very simple problem and suggests to implement the most simple dead reckoning filter: calculate current position by using a previously determined position (starting at the identity quaternion) and advancing that position based upon an estimated speed over the elapsed time). Implement the filter to estimate position by only integrating the gyroscope readings. Consider the initial orientation q[0] to be the identity quaternion [1,0,0,0].

Resources
Section III in [1] (provided in DUO coursework folder)
Chapter 9.2 in LaValle's book (the module's coursebook)

PROBLEM 3 - 20 MARKS:

Years ago, as a student, you attended a VR module where you learned that simple dead reckoning suffers from gyroscopic drift error. You decide to use another source of data available from the IMU, the accelerometer. Implement gravity-based tilt correction using the accelerometer. Instead of just integrating the gyroscope, include accelerometer information too. At all times suppose that there is no linear acceleration. Transform acceleration measurements into the global frame (2 marks). Calculate the tilt axis (2 marks) and find the angle ϕ between the up vector and the vector obtained from the accelerometer (2 marks). Use the complementary filter to fuse the two signals (7 marks). Try a few different alpha values (e.g., 0.01, 0.1, ...), investigate and comment on their effect on drift compensation in your report (7 marks).

Resources
Section V in [1] (provided in DUO coursework folder)
Chapter 9.2 in LaValle's book (the module's coursebook)

PROBLEM 4 - 20 MARKS:

Your next step will be to mitigate yaw drift. Luckily, the cheap IMU magnetometer has been very well-calibrated for the North-East region so you will not do anything for calibration. At every measurement use the magnetometer to estimate if there has been yaw drift by comparing a reference magnetometer reading taken at the beginning of the IMU rotations and each current magnetometer reading (5 marks). Construct and apply the complementary filter as in section VI in [1] (10 marks). Try a few different alpha values (e.g., 0.01, 0.1, ...), investigate and comment on their effect on drift compensation in your report (5 marks).

Resources

Section VI in [1] (provided in DUO coursework folder)
Chapter 9.2 in LaValle's book (the module's coursebook)

PROBLEM 5 - 20 MARKS:

The not-a-computer-scientist CEO of the company is a bit uncertain that your job position is crucial to the success of the company and why it is a justified expense during these hard times. Create visualisations to demostrate the effectiveness of your methods. In particular, first create three 2D plots that show for the input dataset the (i) tri-axial angular rate in deg/s as a function of time (you can use three different colors in the same plot for XYZ), (ii) tri-axial acceleration in g (m/s^2) as a function of time and (iii) tri-axial magnetic flux in Gauss (G) as a function of time (6 marks). Then, create three side-by-side 2D plots that show tri-axial Euler angles (in degrees) for the three implemented methods (gyroscope integration, gyroscope + accelerometer, gyroscope + accelerometer + magnetometer) (again, you can use three different colors in the same plot for XYZ) (6 marks). Finally, create three side-by-side 3D plots that show 3 perpendicular vectors (XYZ) in space corresponding to the 3 main axis of your IMU that rotate in time as the IMU is being rotated (animated plots). Playback the data in the three plots in actual speed, and half the speed (6 marks). Include screenshots and comment on the stability of each method in your report (2 marks). You can use matplotlib for this. The CEO can now understand the importance of your role to the success of the company. This will help you not only keep your job but also get a pay raise!

PROBLEM 6 - FOR L4 ONLY! 30 MARKS:

Your team leader has the exciting idea to try positional tracking using only the orientation sensor. You hypothesize that this can not work well without an external frame of reference, due to the quadratic growth of drift error. However, it may worth the try. Implement crude positional tracking using only the IMU and a kinematic head model. Double integrate the accelerometer and see if the input dataset can indicate an actual change in position (10 marks). Only implement double integration **without** solving the system (equations 15 , 16) in [1]. Create a 3D plot that shows 3 perpendicular vectors (XYZ) in space corresponding to the 3 main axis of your IMU that rotate and translate in time as the IMU is been rotated (animated plot) (10 marks). Note that the IMU was not translating much in space in the provided dataset. Comment on what you see in the 3D plot in your report (10 marks).

Resources
Section VIII in [1] (provided in DUO coursework folder)
Chapter 9.2 in LaValle's book (the module's coursebook)

References

[1] LAVALLE, S. M., YERSHOVA, A., KATSEV, M., AND ANTONOV, M. Head tracking for the oculus rift. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (2014), IEEE, pp. 187–194.

[2] MADGWICK, S. An efficient orientation filter for inertial and inertial/magnetic sensor arrays. *Report x-io and University of Bristol (UK) 25* (2010), 113–118.