Q1. **[5 Points]** Hardware Devices and Components for AR

a. [1 Point] In your own words explain what is the Vergence Accommodation Conflict (VAC)?

b. [1 Point] Explain in your own words how the Meta Quest 3s HMD tracks the hand controllers (both when they are visible and when they are intermittently occluded)

a. It is an unmatch of the focuse of human eyes and the focus in the head set

b. It use the IMU of the handle to track where the hands are, and use the vision algorithm to see the feature of the hand to track that.

c. [1 Point] You are developing your own AR HMD. You decided to use Pancake lenses due to their advantages over Fresnel lens. What other component(s) in the HMD will be required/preferred due to your choice of using Pancake lenses?

d. [1 Point] You are developing your own AR HMD. If "cost" and "manufacturability" of the lenses were **not** the determining factors, why might you still choose Fresnel lenses over Pancake lenses?

e. [1 Point] You are developing the latest and greatest untethered AR HMD (with onboard processing and power) and cost is not a consideration. Why might you still pick a computationally less powerful processor over the most powerful processor out there, assuming they both fit inside the HMD and are easily available?

c. Due to the more reflect times of pancake set has. we maybe need a lighter screen, to have higher back light and also need heat release set.

d. because the Fresnel do not reduce the light of the screen and save power, also make the headset light

e. the battery life is an important part of AR set, it requires a long stand one not a heat and consume CPU. and Also a better CPU means needs a better equiped system to be heavier, high heat make the fps lower.

Q2. **[7 Points]** Kalman Filter for Pose Estimation

a. [0.5 Points] Write out the symbolic Kalman Filter equations
b. [1 Point] Explain in your own words the matrices A, B, P, Q, R *and* K
c. [0.5 Points] What is the difference between Kalman Filter and Extended Kalman Filter?
d. [1 Point] How does the Kalman filter (or EKF) help with pose estimation in AR?

Predict:
$$\begin{cases} \hat{X}_k^- = A_k \hat{X}_{k-1} + B_k U_k \\ P_k^- = A_k P_{k-1} A_k^T + Q_k \end{cases}$$

Update:
$$Y = z - H \cdot X_k$$
$$S = H \cdot P_k \cdot H^T + R$$
$$K = P_k \cdot H^T \cdot S^{-1}$$
$$X_{k+1} = X_k + K \cdot Y$$
$$P = (I - K \cdot H) \cdot P_k$$

b). A: transfer the status of last timestamp to the next one

B: Put the input (accelerat ..) to the status

P: the evaluation of unknown of current

Q: the ability of model to conqur the interup from outside

R: the uncertainty of the sensor.

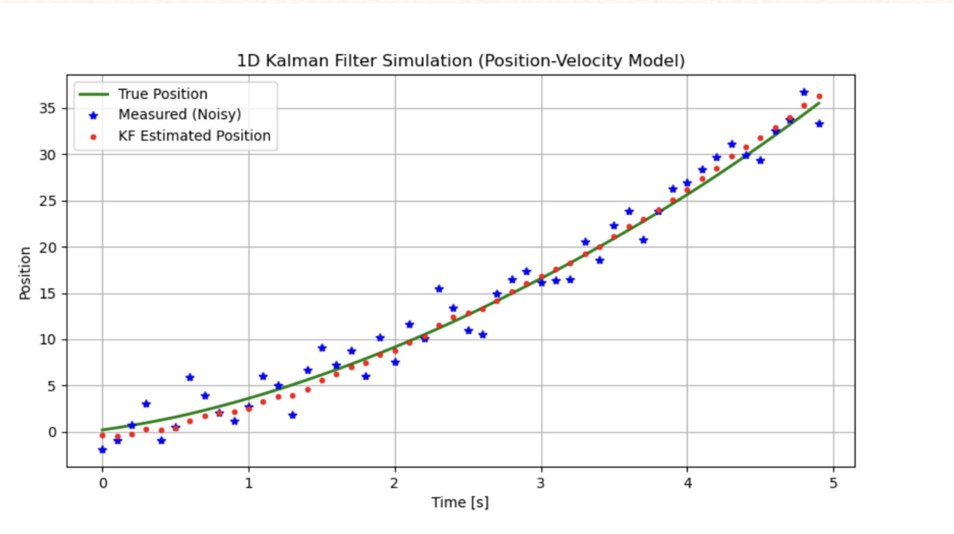
K: the balance between trust model and trust the sensor

c) KF: use ABH to directly compute
it is linear
EKF: it is unlinear and use Taylor's
to be linear. use Jacobian to replace A, H.
make the mistake less.

d): intergrate IMU and camera to output
a smooth low-latency 6DoF pose

predict when the camera lose frame
also predict velocity and align the
Timestamp- and denoise

e)



f)