Announcements

- **MP4** is out. Due on Nov 6 @ 11:59pm.

**Final project upcoming deadlines:**

- Nov 13, a short video of your progress.
- Dec 16, final project presentation.

**Grades are out for:**

- Midterm 1
- Project abstract and picture-title
Tracking Systems in VR

What do we want to track? Think about rigid bodies:

1. Head wearing HMD
2. Eyes
3. Palms of hands
4. Fingers
5. Entire body
6. Movable objects - controller, coffee cup, desk
7. Other people in the space
Tracking Systems in VR

What do we want to track:

For each rigid body, estimate

Rotation:

Position:

OR

Homogeneous transformation matrix:

Task:
Tracking Systems in VR: Estimating 2D Orientation

Initial conditions

For constant velocity

For variable velocity

Discrete-time approximation
Tracking Systems in VR: Estimating 2D Orientation

Discrete-time approximation

$$\theta_k \approx \theta_0 + \sum_{i=1}^{k} \Delta \theta_i =$$

$$= \theta_0 + \sum_{i=1}^{k} \omega_i \Delta t$$

$$\Delta t = 1 \text{ msec}$$

$$\omega_i = \omega \left( (i-1) \Delta t \right)$$

Estimation given sensor readings $$\omega_i$$ every $$\Delta t$$

Q1: does it matter where we put the gyroscope on the disk?
Q2: How many lines of code to implement this estimation?
Axis-angle: 3-axis gyroscope measures:

\[ \omega = \]

Examples of VR systems with IMU only tracking:
Tracking Systems in VR: Estimating 3D Orientation

Axis-angle: \((\vec{v}, \Theta)\)

3-axis gyroscope measures:
\[
\vec{\omega} = (\omega_x, \omega_y, \omega_z)
\]

How large is \(\Delta Q_i\)?
Tracking Systems in VR: Estimating 3D Orientation

Integrate sensor readings to estimate orientation:

Recall 2D:

\[ \hat{\Theta}_k = \Theta_0 + \sum_{i=1}^{k} \Delta \hat{\Theta}_i = \]

\[ = \Theta_0 + \Delta \hat{\Theta}_1 + \Delta \hat{\Theta}_2 + \ldots + \Delta \hat{\Theta}_k \]

3D:

\[ \hat{Q}_k = \]

Recursively/Incrementally:

2D:

\[ \hat{\Theta}_{\text{current}} = \]

3D:

\[ \hat{Q}_{\text{current}} = \]

Problem:
Estimating 3D Orientation: Drift Correction

Expression for drift error:

$2D: \quad d_k =$

$3D: \quad d_k =$

An unusual quantity to have for an error!
Correcting for drift errors challenge:

1) Use another sensor

2) Gradually apply corrections
   - Fast enough to
   - Slow enough to
Estimating 3D Orientation: Drift Correction

Separate the rotational drift error into two components:

1) To correct:

2) To correct: