Selection and Integration of Sensors

Alex Spitzer aes368@cornell.edu
11/23/14
Sensors

- Perception of the outside world
- Cameras, DVL, Sonar, Pressure
- Accelerometers, Gyroscopes, Magnetometers
- Position vs Orientation
Why?

- Knowing vehicle state is important!
  - Heading and velocity in particular
- Position, velocity, orientation, and angular velocity
- Theory: Accel and Gyro enough
- Intuition: 6 numbers, 6 DOF
- Absolute measurements are essential!
  - e.g. ADCP (Acoustic Doppler Current Profiler) popularly known as a DVL yields velocity
  - Magnetometers yield heading (one angle)
- Other cues useful (e.g. visual)
Accelerometers

- Proper acceleration
- Usually a set of 3 orthogonal units that measure acceleration in 3 axes
- Can be used to derive orientation
- Which way is gravity pointing?
- Assumption: no additional accelerations
Gyroscopes

- Angular Velocity, RPMs, rad/s, etc.
- Can be used to derive orientation much like accelerometers can be used to derive velocity
- Integrate!
- Combine with accelerometers
- Gyro gain
Magnetometers

- Measure the magnetic field
- Provide an absolute measurement of heading
- Susceptible to magnetic interference and distortions of magnetic field
- Calibration in vehicle important!
What is a 9 axis IMU/AHRS?

- IMU - Inertial Measurement Unit
- AHRS - Attitude and Heading Reference System
- Typically contain accelerometers, gyroscopes, and magnetometers (3 each)
- 6 axis units usually contain accelerometers and one of the other two
- Raw output available
- Microcontroller? Fused/processed output available
- Simple trig to fancy sensor fusion
Integrating is not enough

- Why can’t I just integrate gyroscope data to get my orientation?
- Need reference orientation
- Integrating random noise quickly leads to massive error
- Intuition: random walk
- Absolute measurements are essential!
- Ideal: GPS, does not work underwater
- DVL: velocity is nice
- Magnetism: orientation in surface plane
- Gravity: orientation in up and down plane
Where to process

- Firmware vs Software
- Fancy sensors => fancy firmware
- Paying for the math, algorithm
- On unit: Firmware can greatly simplify your job
- In addition to firmware, higher end units may have better hardware (i.e. more accurate raw data)

- Off unit: More control in software
- Output raw sensor bits, fuse
Selection

- High End: DVL + AHRS
- Position tracked by integrating DVL velocity (need orientation)
- Careful with firmware filtering, can introduce lag
- Play around with settings

- Alternative: More orientation processing off sensor

- System compatibility important

- No DVL? Visual cues important
Integration

- Placement
- Away from magnets, CPU, etc.
- Current generates magnetic field
- Electric motors contain strong magnets
- Sensor boom?

- Communication: usually RS-232, i.e. serial
- USB hub
- Microcontroller accepts commands
- Implementing serial protocols not really fun
- C, Python (pyserial)
- SDK is a plus
Integration

- Output
- 100 Hz probably fast enough
- No need to clock faster than consumer
- Polling vs Continuous mode

- Format?
- Euler angles are simplest to use, but stay away
- Matrices or quaternions better
- See webinar on Controller - 11/29
Calibration (Gyros and Accel)

- **Gyro gain**
  Keep still to capture gain
  Steady drift in heading, pitch, and roll values? Capture gyro gain

- **Accel gain, same deal**
  Capture local gravity vector (keep sensor flat)

- See data sheets on exact procedures
Calibration (Magnets)

● Hard iron vs Soft iron
● Hard iron: permanent magnets
● Moving permanent magnets impossible to calibrate for (but can compensate with knowledge)

● Soft iron: Changes in magnetic permeability
● Major offenders: iron, steel
● Aluminum okay

● Hard iron adds offset to magnetic readings
● Soft iron skews magnetic readings
Calibration

- Most high end IMUs come with software suites
- Exclusively run on Windows
- “Manual” calibration may be tedious

- Do it once well and forget
- Stay away from hard and soft iron.
- Add components? Recalibrate.
- Change location? Recalibrate.

- Complete soft iron calibration may be infeasible
Linearization of Heading

- Poor soft iron compensation can lead to “non linear” headings
- Simple way to correct: Look up table with interpolation
Sensor Fusion

- Basic filtering, e.g. Low pass filter
- Idea: Low jerk!
- Kalman filter
- Simple example: IMU outputs heading (computed from magnetometer) and heading rate (from gyroscopes)

Heading is noisy, gyro is subject to drift
Simple option? Just filter the heading data (doesn’t use gyro)
How about we average result of both? We can do better...
Kalman filter! state is (heading, heading rate)

Predict: next_heading = heading + heading_rate*dt
Correct: heading = next_heading + K(measured_heading - next_heading)
heading = hdg + hdg_rate*dt + K(measured_hdg - hdg - hdg_rate*dt)
heading = (1 - K) * (hdg + hdg_rate*dt) + K * measured_hdg
heading = (1 - K) * (gyro_heading) + K * magnetometer_heading

Result is smoother than heading data and does not suffer from drift. Best of both worlds!
Advanced Sensor Fusion

- Fusing variable and its derivative is easy.
- We want to estimate orientation (fuse all 9 axes)
- Kalman filter is limited (assumes linear model)
- EKF (Extended Kalman filter) linearizes about the state using Jacobian - “de facto standard”
- UKF (Unscented Kalman filter) captures non-linearities more accurately using particularly chosen sample points (to the 3rd order)
- State? Quaternion + angular velocity vector
Orientation

- Attitude
- 3 DOFs
- Most compact? Euler angles, Rotation vector
- Euler angles suffer from gimbal lock
- Matrices are nice but redundant
- Quaternions are hip and cool, industry standard
Orientation

• Euler’s Rotation Theorem: Any orientation can be represented by a single vector (or a unit axis and an angle around that axis)

  q0 = \cos(\theta/2)
  q1 = v_x \times \sin(\theta/2)
  q2 = v_y \times \sin(\theta/2)
  q3 = v_z \times \sin(\theta/2)

• Limit 4 vector to unit length
• Multiplication has similar semantics to matrices
Questions?
aes368@cornell.edu