Tim Player

Math 142 Midterm Project: Computer Vision Pose Estimation

Problem:

Determine position, velocity of rocket during ascent.

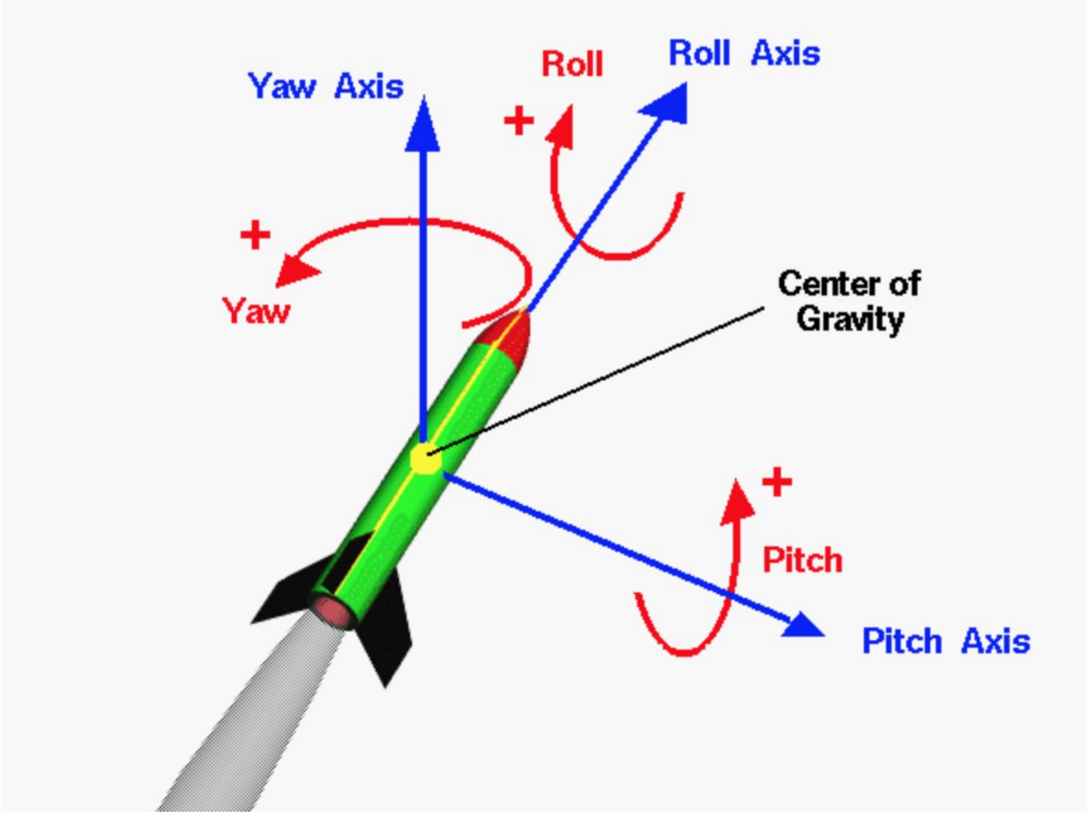
“Inertial” Inputs:

- accelerometer data,
- rate gyroscope data,
- magnetometer data

Special Inputs:

- GPS location at launchpad, apogee
- 30 Hz video

A note on direction conventions:



Horizon Estimation for Global Pitch and Yaw

Horizon Estimation Algorithm:

a. Makes a mask that includes the sky but not the ground.

b. Finds the edges of that mask.

c. Picks the longest edge. Assumes that contains the horizon.

d. Crops the image to only examine the central rectangle (and not the edges of the circular window).

e. Uses Hough line transform to identify a long straight line.

(Gaussian blur, HSV transform, canny edge detection, Hough transform)

Horizon Estimation Algorithm + Demo!

(a)

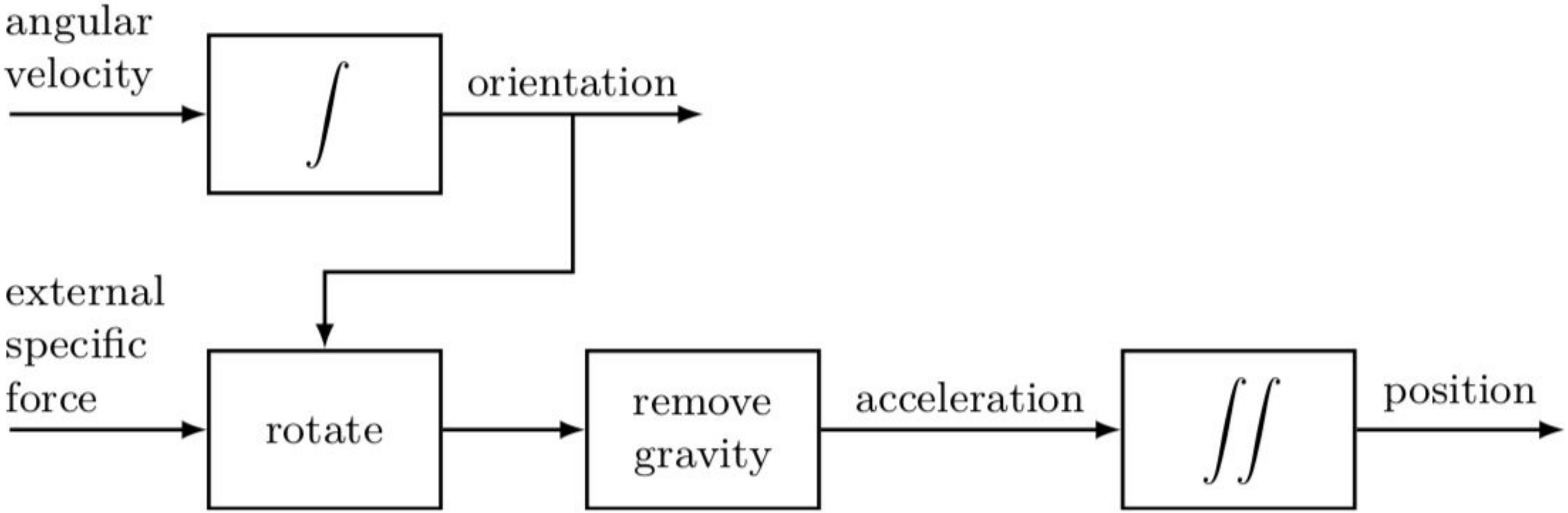
(b)

(c)

(d)

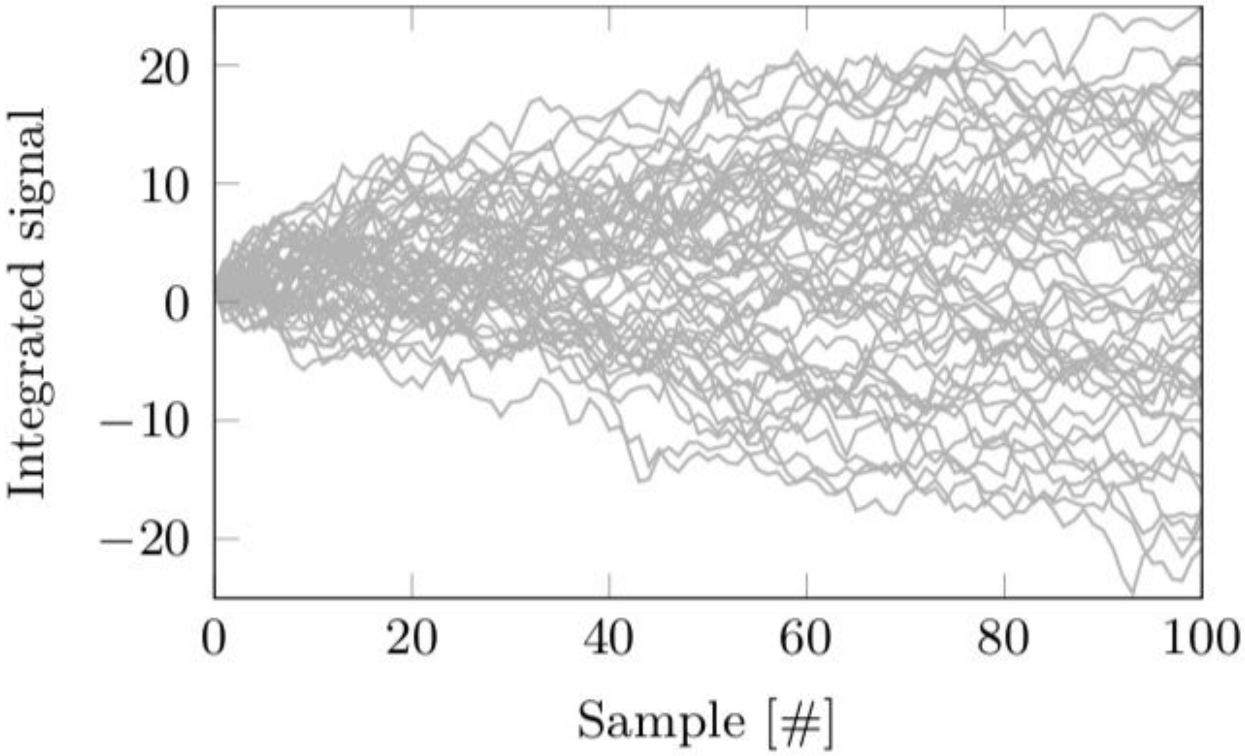
(e)

Sensor Fusion



Kok, Manon, et al. “Using Inertial Sensors for Position and Orientation Estimation.” *Foundations and Trends® in Signal Processing*, vol. 11, no. 1-2, 2017, pp. 1–153., doi:10.1561/2000000094.

Noisy measurements quickly integrate



Solution: Recursive Bayes Filter

Algorithm Bayes_filter$(bel(x_{t-1}), u_t, z_t)$**:**
 for all x_t do
  $\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1})\ bel(x_{t-1})\ dx$
  $bel(x_t) = \eta\ p(z_t \mid x_t)\ \overline{bel}(x_t)$
 endfor
 return $bel(x_t)$

Thrun, Sebastian, et al. *Probabilistic Robotics*. MIT Press, 2006.

Implementation: Kalman Filter

Assumptions:

- Locally linear system dynamics A
- Zero mean gaussian process noise
- Zero mean gaussian measurement noise
- Approximately linear measurement law

Algorithm Kalman_filter($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):

$\bar{\mu}_t = A_t\, \mu_{t-1} + B_t\, u_t$

$\bar{\Sigma}_t = A_t\, \Sigma_{t-1}\, A_t^T + R_t$

$K_t = \bar{\Sigma}_t\, C_t^T (C_t\, \bar{\Sigma}_t\, C_t^T + Q_t)^{-1}$

$\mu_t = \bar{\mu}_t + K_t(z_t - C_t\, \bar{\mu}_t)$

$\Sigma_t = (I - K_t\, C_t)\, \bar{\Sigma}_t$

return μ_t, Σ_t

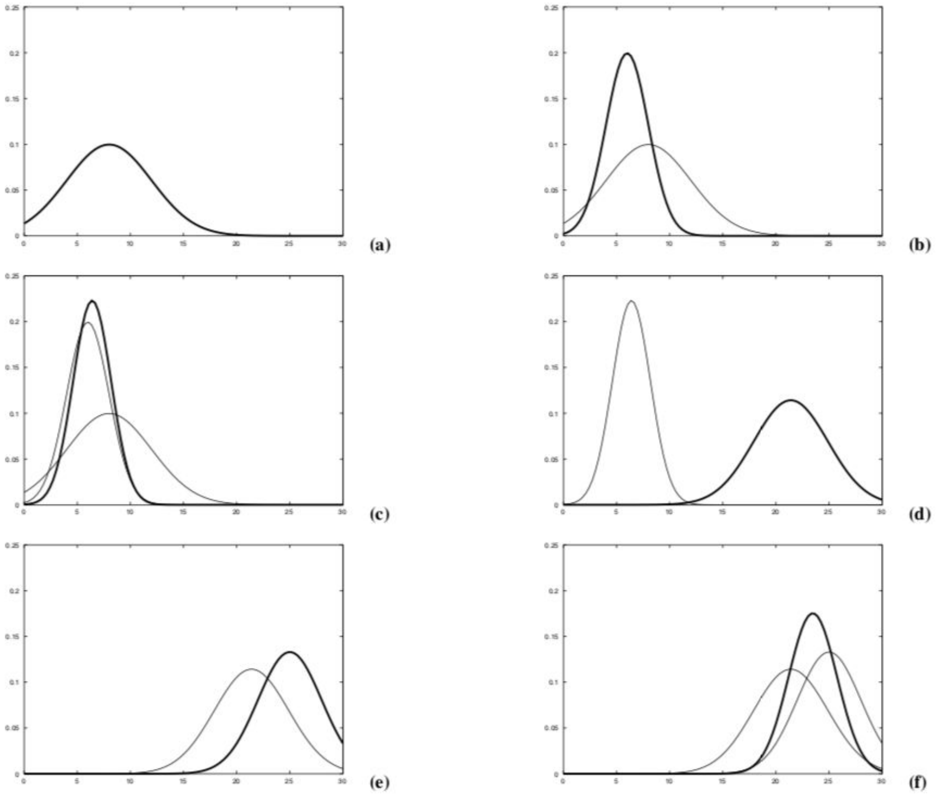


Figure 3.2 Illustration of Kalman filters: (a) initial belief, (b) a measurement (in bold) with the associated uncertainty, (c) belief after integrating the measurement into the belief using the Kalman filter algorithm, (d) belief after motion to the right (which introduces uncertainty), (e) a new measurement with associated uncertainty, and (f) the resulting belief.

Differential Geometry Insight

- Solution space restricted to SE(3) manifold
  - renormalized unit quaternion enforces SO(3) constraint
- Kalman filter derivation can be thought of as projection onto a Hilbert space or gradient descent
- Rocket ascent trajectory is regular curve

Data Pipeline

1. Find initial GPS location on the launch pad.
2. Convert GPS data (not available during ascent) into meters from origin
3. Apply laboratory bias-calibrations to raw accelerometer, gyroscope, and magnetometer data.
4. Flip magnetometer x, y aces to get right-handed system
5. Scale magnetometer dimensions individually to normalize to $(-1, 1)$.
6. Scale magnetometer readings so that at every step t the L_2 norm is 1.
7. Get initial magnetometer orientation from the pre-launch sample mean. Also get sensor variance.
8. Resample magnetometer using smooth interpolation.
9. Convert gyroscope data to radians/sec.
10. Convert accelerometer data to meters/sec using local gravitational constant.
11. Get acceleration baseline from pre-launch sample mean. Also get sensor variance.
12. Resample accelerometer and gyroscope data.
13. Determine the global pitch and yaw corresponding to each image.
14. Smooth the roll using complementary filter.
15. Determine initial orientation using TRIAD method.

Future work:

Expand on existing inertial EKF:

- code function to convert horizon height, slope to global pitch, yaw.
- validate filter convergence with new measurement input
- consider new differential geometry approaches to problem
    - SE(3)-Net?