Advanced IMU Sensor Fusion with Kalman Filtering

Gordon Wetzstein
Stanford University

EE 267 Virtual Reality

Lecture 11

stanford.edu/class/ee267/

May 2, 2016

Overview

- recursive Bayesian filter
- Kalman Filter
- Extended Kalman Filter
- state model, process model, measurement model
- example for IMU

Goal

- fuse measurements from gyro, accelerometer, magnetometer
- each sensor has different problems – get best of all via fusion
- use basic probability to derive optimal sensor fusion algorithm
    - derivation in the lecture, may seem complicated
    - practical implementation turns out reasonably easy
    - TinyEKF (see references)

Goal

- approach: we somehow need to keep track of the desired state (i.e. orientation, bias, etc.) while keeping track of the uncertainty associated with the sensors and our state prediction

- Kalman filtering: can be interpreted as a Bayesian framework to do that

Goal

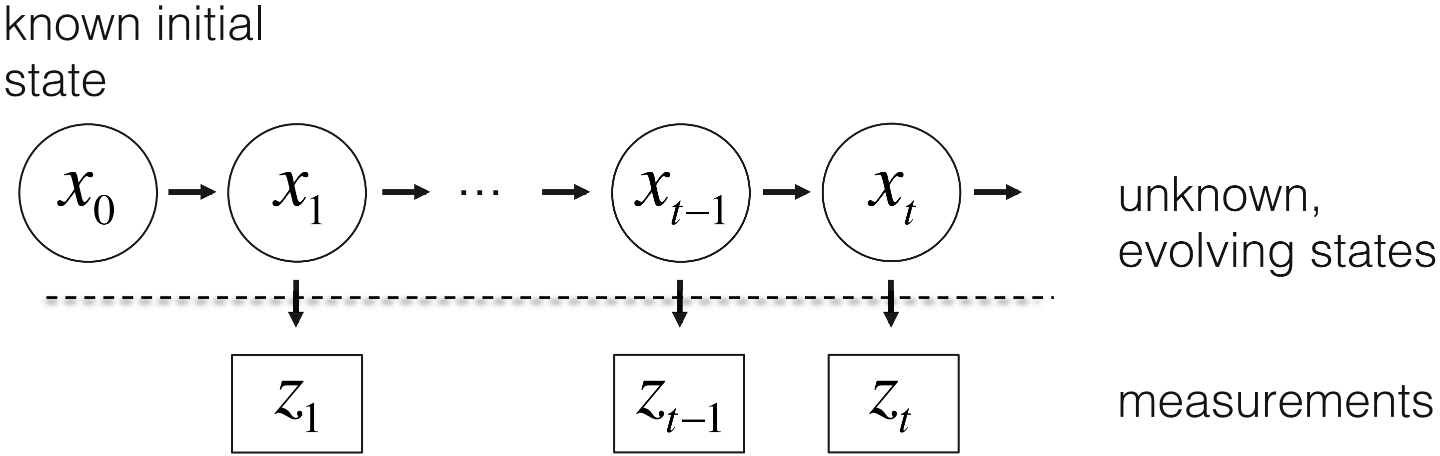
- estimate pdf (i.e. using Bayes' rule) of current state x_t given all previous (and current) measurements $z_{1:t}$

state measurements

probability density function $\longrightarrow$ $p\left(x_t \mid z_{1:t}\right)$

General Model

- Hidden Markov Model (HMM)



General Model

- Hidden Markov Model (HMM)
- assumptions:

$$p\left(x_t \mid x_{1:t-1}, z_{1:t}\right) = p\left(x_t \mid x_{t-1}\right)$$

$$p\left(z_t \mid x_{1:t}, z_{1:t-1}\right) = p\left(z_t \mid x_t\right)$$

- i.e. no systematic measurement noise

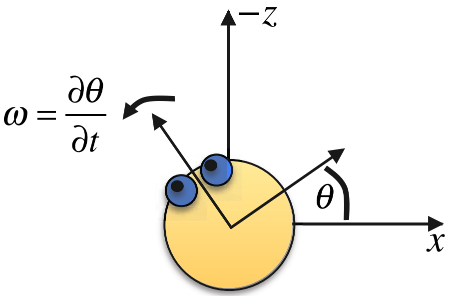
State Model

- what is the state x?
- all model parameters: orientation angles or quaternions, angular velocities, whatever properties you want to track
- simple 1D example:

$$x = \begin{pmatrix} \theta \\ \omega \\ b \end{pmatrix} \begin{array}{l} \leftarrow \text{angle} \\ \leftarrow \text{angular velocity} \\ \leftarrow \text{gyro bias} \end{array}$$

State Model

- what is the state x?
- all model parameters: orientation angles or quaternions, angular velocities, whatever properties you want to track
- simple 1D example:



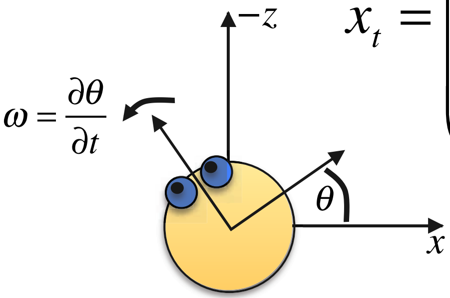
$$x = \begin{array}{cl} \begin{pmatrix} \theta \\ \omega \\ b \end{pmatrix} & \begin{array}{l} \leftarrow \text{angle} \\ \leftarrow \text{angular velocity} \\ \leftarrow \text{gyro bias} \end{array} \end{array}$$

Process Model

- given state at t-1, what's the state at t? $$p\left(x_t \mid x_{t-1}\right)$$

- simple 1D example:

$$x_t = \begin{pmatrix} \theta_t \\ \omega_t \\ b_t \end{pmatrix} = \begin{pmatrix} \theta_{t-1} + \Delta t\left(\omega_{t-1} - b_{t-1}\right) \\ \omega_{t-1} \\ b_{t-1} \end{pmatrix} \begin{matrix} \text{integration} \\ \text{constant} \\ \text{constant} \end{matrix}$$

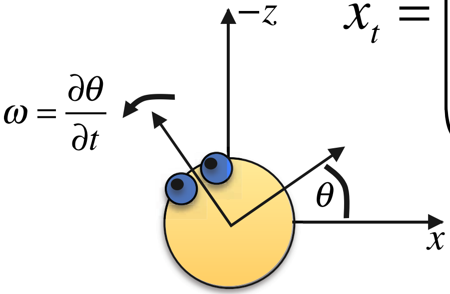


Process Model

- given state at t-1, what's the state at t? $p\left(x_t \mid x_{t-1}\right)$

- simple 1D example:

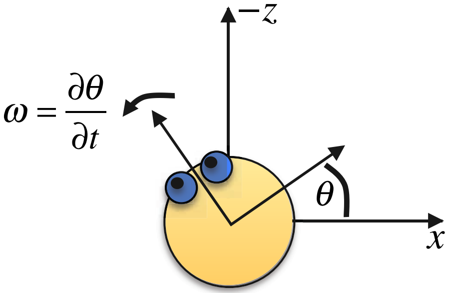
$$x_t = \begin{pmatrix} \theta_t \\ \omega_t \\ b_t \end{pmatrix} = \begin{pmatrix} 1 & \Delta t & -\Delta t \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \theta_{t-1} \\ \omega_{t-1} \\ b_{t-1} \end{pmatrix}$$



linear!

Measurement Model

- models how state parameters map to measurements (linear or nonlinear models) $p(z_t \mid x_t)$
- simple 1D example:
    - measure ω directly with gyro = linear
    - say north is $-z$, then magnetometer measures



$$\theta = \cos^{-1}\left(\frac{m_z}{\sqrt{m_x^2 + m_z^2}}\right) \longleftarrow \text{nonlinear!}$$

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} m_x \\ m_z \end{pmatrix} = \left\| \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\| \left\| \begin{pmatrix} m_x \\ m_z \end{pmatrix} \right\| \cos\theta$$

Recursive Bayes Estimation

- given estimate of previous state x_{t-1} and current measurements z_t, what is x_t? → recursive update!

$$
\begin{aligned}
& p(x_t \mid z_{1:t}) \\
& \quad = \eta\, p(z_t \mid x_t, z_{1:t-1})\, p(x_t \mid z_{1:t-1}) && \text{Bayes' rule} \\
& \quad = \eta\, p(z_t \mid x_t)\, p(x_t \mid z_{1:t-1}) && \text{Markov assumption} \\
& \quad = \eta\, p(z_t \mid x_t) \int p(x_t \mid x_{t-1}, z_{1:t-1})\, p(x_{t-1} \mid z_{1:t-1})\, dx_{t-1} && \text{law of total probability} \\
& \quad = \eta\, \underbrace{p}_{\text{measurement model}}(z_t \mid x_t) \int \underbrace{p}_{\text{process model}}(x_t \mid x_{t-1})\, \underbrace{p}_{\text{previous posterior}}(x_{t-1})\, dx_{t-1} && \text{Markov assumption}
\end{aligned}
$$

Recursive Bayes Estimation

- two steps:
    1. <u>prediction</u>: $$\overline{p}(x_t) = \int p(x_t \mid x_{t-1})\, p(x_{t-1})\, dx_{t-1}$$
    2. <u>update</u>: $$p(x_t) = \eta\, p(z_t \mid x_t)\, \overline{p}(x_t)$$

$$= \eta\, p(z_t \mid x_t) \int p(x_t \mid x_{t-1})\, p(x_{t-1})\, dx_{t-1}$$

measurement model process model previous posterior

Recursive Bayes Estimation

- two steps:
  1. <u>prediction</u>: $\overline{p}(x_t) = \int p(x_t \mid x_{t-1})\, p(x_{t-1})\, dx_{t-1}$
  2. <u>update</u>: $p(x_t) = \eta\, p(z_t \mid x_t)\, \overline{p}(x_t)$

- generic probabilistic model for recursive update
- so far, no assumptions on measurements, processes, etc.
- special cases with assumptions: Kalman Filter (linear, Gaussian), Extended Kalman Filter (linearized, Gaussian)

Kalman Filter

- recursive Bayesian filter for linear, Gaussian models

process model for motion:

$$x_t = A x_{t-1} + w_{t-1}$$

measurement model:

$$z_t = H x_t + v_t$$

linear! Gaussian

state space and noise are random Gaussian variables:

$$x \sim \mathrm{N}(\mu, \Sigma), \; w \sim \mathrm{N}(0, Q), v \sim \mathrm{N}(0, R)$$

Extended Kalman Filter

- recursive Bayesian filter for nonlinear, Gaussian models

process model for motion: $$x_t = A\left(x_{t-1}\right) + w_{t-1}$$

measurement model: $$z_t = H\left(x_t\right) + v_t$$

nonlinear! Gaussian

state space and noise are random Gaussian variables: $$x \sim \mathrm{N}(\mu, \Sigma),\ w \sim \mathrm{N}\left(0, Q\right), v \sim \mathrm{N}\left(0, R\right)$$

Extended Kalman Filter

- linearize nonlinear models using first-order Taylor expansion / Jacobian matrix when necessary

process model for motion: $$x_t = A\left(x_{t-1}\right) + w_{t-1}$$

measurement model: $$z_t = H\left(x_t\right) + v_t$$

nonlinear! Gaussian

state space and noise are random Gaussian variables: $$x \sim \mathrm{N}(\mu, \Sigma),\ w \sim \mathrm{N}(0, Q), v \sim \mathrm{N}(0, R)$$

Extended Kalman Filter

- <u>prediction</u> step for Gaussian state space, given previous state with mean $\mu_{t-1} \equiv x_{t-1|t-1}$ and covariance matrix $\Sigma_{t-1} = P_{t-1|t-1}$

linear process model (Kalman Filter): $x_{t|t-1} = Ax_{t-1|t-1}, \qquad P_{t|t-1} = AP_{t-1|t-1}A^T + Q_t$

nonlinear process model (Extended Kalman Filter): $x_{t|t-1} = A\left(x_{t-1|t-1}\right), \quad P_{t|t-1} = J_A^{\left(x_{t-1|t-1}\right)} P_{t-1|t-1} J_A^{\left(x_{t-1|t-1}\right)^T} + Q_t$

$$x \sim \mathrm{N}(\mu, \Sigma),\ w \sim \mathrm{N}(0, Q), v \sim \mathrm{N}(0, R)$$

Extended Kalman Filter

- <u>update</u> step for Gaussian state space, given current measurements

linear process model (Kalman Filter):

	$y_t = z_t - H_t x_{t\|t-1}$	innovation or measurement residual
	$S_t = H_t P_{t\|t-1} H_t^T + R_t$	innovation covariance
	$K_t = P_{t\|t-1} H_t^T S_t^{-1}$	Kalman gain
	$x_{t\|t} = x_{t\|t-1} + K_t y_t$	update state estimate
	$P_{t\|t} = \left(I - K_t H_t\right) P_{t\|t-1}$	update covariance estimate

Extended Kalman Filter

- <u>update</u> step for Gaussian state space, given current measurements

nonlinear process model (Extended Kalman Filter):

	$y_t = z_t - H\left(x_{t\|t-1}\right)$	innovation or measurement residual
	$S_t = J_H^{\left(x_{t\|t-1}\right)} P_{t\|t-1} J_H^{\left(x_{t\|t-1}\right)^T} + R_t$	innovation covariance
	$K_t = P_{t\|t-1} J_H^{\left(x_{t\|t-1}\right)^T} S_t^{-1}$	Kalman gain
	$x_{t\|t} = x_{t\|t-1} + K_t y_t$	update state estimate
	$P_{t\|t} = \left(I - K_t J_H^{\left(x_{t\|t-1}\right)}\right) P_{t\|t-1}$	update covariance estimate

Example: 6 DOF IMU Sensor Fusion

Euler Angles

Extended Kalman Filter – IMU Euler Angles

- state space (or better: its mean)

$$x = \begin{pmatrix} \theta_x \\ \theta_y \\ \theta_z \\ \omega_x \\ \omega_y \\ \omega_z \\ b_x \\ b_y \\ b_z \end{pmatrix} \begin{array}{l} \left. \begin{array}{l} \\ \\ \\ \end{array} \right| \text{Euler angles} \\ \left. \begin{array}{l} \\ \\ \\ \end{array} \right| \text{angular velocity} \\ \left. \begin{array}{l} \\ \\ \\ \end{array} \right| \text{gyro bias} \end{array}$$

Extended Kalman Filter – IMU Euler Angles

- linear process model (assumes velocity & bias stay constant)

$$x = \begin{pmatrix} \theta_x \\ \theta_y \\ \theta_z \\ \omega_x \\ \omega_y \\ \omega_z \\ b_x \\ b_y \\ b_z \end{pmatrix} \qquad x_{t+1} = Ax_t = \begin{pmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 & -\Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & -\Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & -\Delta t \\ & & & 1 & & & & & \\ & & & & 1 & & & & \\ & & & & & 1 & & & \\ & & & & & & 1 & & \\ & & & & & & & 1 & \\ & & & & & & & & 1 \end{pmatrix} x_t = \begin{pmatrix} \theta_x^{(t)} + \Delta t\left(\omega_x^{(t)} - b_x^{(t)}\right) \\ \theta_y^{(t)} + \Delta t\left(\omega_y^{(t)} - b_y^{(t)}\right) \\ \theta_z^{(t)} + \Delta t\left(\omega_x^{(t)} - b_z^{(t)}\right) \\ \omega_x^{(t)} \\ \omega_y^{(t)} \\ \omega_z^{(t)} \\ b_x^{(t)} \\ b_y^{(t)} \\ b_z^{(t)} \end{pmatrix}$$

Extended Kalman Filter – IMU Euler Angles

- linear process model – covariance matrix

$$x = \begin{pmatrix} \theta_x \\ \theta_y \\ \theta_z \\ \omega_x \\ \omega_y \\ \omega_z \\ b_x \\ b_y \\ b_z \end{pmatrix} \qquad Q_t = \begin{pmatrix} \Delta t^2 & & & & & & & & \\ & \Delta t^2 & & & & & & & \\ & & \Delta t^2 & & & & & & \\ & & & 0.01\Delta t & & & & & \\ & & & & 0.01\Delta t & & & & \\ & & & & & 0.01\Delta t & & & \\ & & & & & & 0.03\Delta t & & \\ & & & & & & & 0.03\Delta t & \\ & & & & & & & & 0.03\Delta t \end{pmatrix}$$

Extended Kalman Filter – IMU Euler Angles

- linear process model – covariance matrix

$$x = \begin{pmatrix} \theta_x \\ \theta_y \\ \theta_z \\ \omega_x \\ \omega_y \\ \omega_z \\ b_x \\ b_y \\ b_z \end{pmatrix} \qquad Q_t = \begin{pmatrix} \Delta t^2 &&&&&&&& \\ & \Delta t^2 &&&&&&& \\ && \Delta t^2 &&&&&& \\ &&& 0.01\Delta t &&&&& \\ &&&& 0.01\Delta t &&&& \\ &&&&& 0.01\Delta t &&& \\ &&&&&& 0.03\Delta t && \\ &&&&&&& 0.03\Delta t & \\ &&&&&&&& 0.03\Delta t \end{pmatrix}$$

integration error from Taylor truncation

estimated error of angular velocity = const. approximation

estimated error of bias = const. approximation

Extended Kalman Filter – IMU Euler Angles

- measurements

$$z = \begin{pmatrix} \tilde{\omega}_x \\ \tilde{\omega}_y \\ \tilde{\omega}_z \\ \bar{a}_x \\ \bar{a}_y \\ \bar{a}_z \end{pmatrix}$$

raw gyro measurements

normalized accelerometer vector

Extended Kalman Filter – IMU Euler Angles

- nonlinear measurement model ☹

$$z = \begin{pmatrix} \tilde{\omega}_x \\ \tilde{\omega}_y \\ \tilde{\omega}_z \\ \bar{a}_x \\ \bar{a}_y \\ \bar{a}_z \end{pmatrix} = H(x) = \begin{pmatrix} \omega_x + b_x \\ \omega_y + b_y \\ \omega_z + b_y \\ -\sin(-\theta_z)\cos(-\theta_x) \\ \cos(-\theta_z)\cos(-\theta_x) \\ \cos(-\theta_x) \end{pmatrix}$$

Extended Kalman Filter – IMU Euler Angles

- nonlinear measurement model

$$J_H^{(x)} = \begin{array}{c} \begin{array}{ccccc} {\partial z}/{\partial x_1} & {\partial z}/{\partial x_2} & {\partial z}/{\partial x_3} & \cdots & {\partial z}/{\partial x_9} \end{array} \\ \left(\begin{array}{ccccc} \quad & \quad & \quad & \quad & \quad \\ \quad & \quad & \quad & \quad & \quad \\ \quad & \quad & \quad & \quad & \quad \end{array} \right) \end{array} \begin{array}{c} \\ {\partial z_1}/{\partial x} \\ \vdots \\ {\partial z_6}/{\partial x} \end{array}$$

Extended Kalman Filter – IMU Euler Angles

- nonlinear measurement model

$$
J_H^{(x)} = \begin{array}{cc}
\begin{matrix} \partial z / \partial x_1 & \partial z / \partial x_2 & \partial z / \partial x_3 & \cdots & \partial z / \partial x_9 \end{matrix} & \\
\begin{pmatrix}
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
\sin(-\theta_z)\sin(-\theta_x) & 0 & -\cos(-\theta_z)\cos(-\theta_x) & 0 & 0 & 0 & 0 & 0 & 0 \\
-\cos(-\theta_z)\sin(-\theta_x) & 0 & -\sin(-\theta_z)\cos(-\theta_x) & 0 & 0 & 0 & 0 & 0 & 0 \\
\cos(-\theta_x) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} & \begin{matrix} \partial z_1 / \partial x \\ \\ \\ \vdots \\ \\ \partial z_6 / \partial x \end{matrix}
\end{array}
$$

Extended Kalman Filter

- <u>prediction</u> step for Gaussian state space, given previous state with mean $\mu_{t-1} \equiv x_{t-1|t-1}$ and covariance matrix $\Sigma_{t-1} = P_{t-1|t-1}$

linear process model (Kalman Filter):

$$x_{t|t-1} = Ax_{t-1|t-1}, \qquad P_{t|t-1} = AP_{t-1|t-1}A^T + Q_t$$

$$x \sim \mathrm{N}(\mu, \Sigma),\ w \sim \mathrm{N}(0, Q), v \sim \mathrm{N}(0, R)$$

Extended Kalman Filter

- <u>update</u> step for Gaussian state space, given current measurements

nonlinear process model (Extended Kalman Filter):

$y_t = z_t - H\left(x_{t\mid t-1}\right)$	innovation or measurement residual
$S_t = J_H^{\left(x_{t\mid t-1}\right)} P_{t\mid t-1} J_H^{\left(x_{t\mid t-1}\right)^T} + R_t$	innovation covariance
$K_t = P_{t\mid t-1} J_H^{\left(x_{t\mid t-1}\right)^T} S_t^{-1}$	Kalman gain
$x_{t\mid t} = x_{t\mid t-1} + K_t y_t$	update state estimate
$P_{t\mid t} = \left(I - K_t J_H^{\left(x_{t\mid t-1}\right)}\right) P_{t\mid t-1}$	update covariance estimate

Extended Kalman Filter – IMU Euler Angles

- linear process model
- could model as uncorrelated variances for all state variables
- gyro and bias error variance is mostly an approximation (hand-tuned values)
- reasonably simple update rules that are derived from recursive Bayesian model
- if you want to make it easy, use a library like TinyEKF (just implement process and measurement models)

Extended Kalman Filter – IMU Euler Angles

- uses significantly more memory and processing than complementary filter!
- couldn't compile on Arduino Metro Mini
- could be a great course project! (e.g. with other microprocessor, e.g. Teensy)

Extended Kalman Filter – Outlook

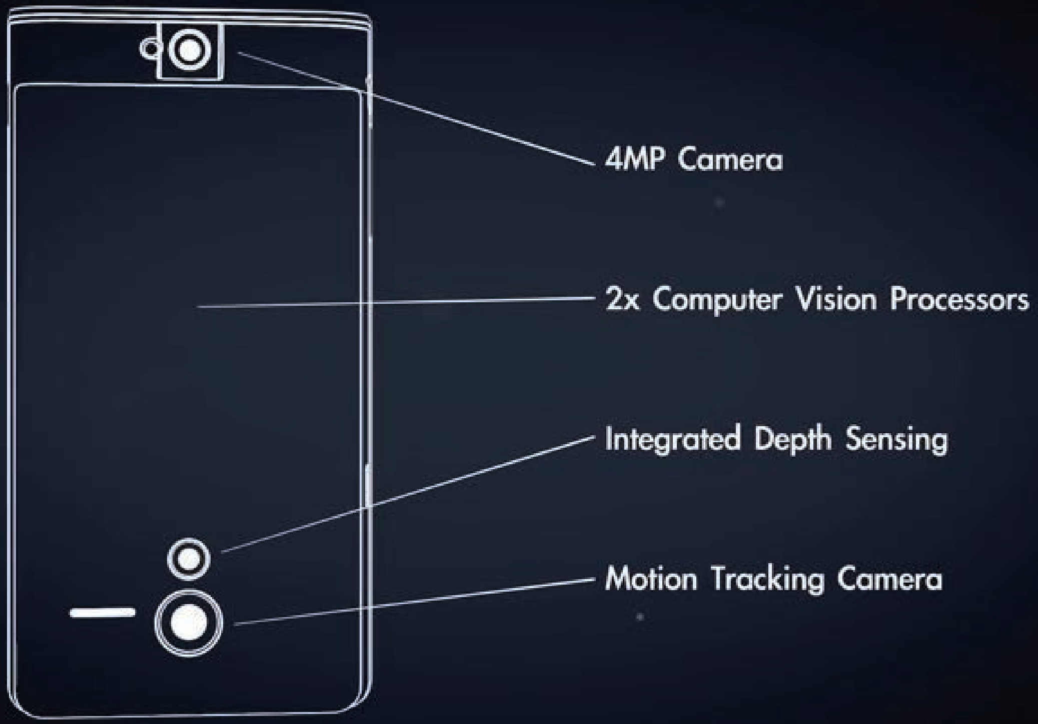
- EKF gets more complicated when working with quaternions
  - process model is also nonlinear
  - not trivial to model the noise quaternion – lots of literature

- other variants of nonlinear Kalman are very useful, e.g. unscented Kalman filter (see Kraft 2003)

Extended Kalman Filter – Outlook

- probabilistic model extends well to other sensors!
- examples of fusing other sensors with IMU:
    - include GPS for rough position estimation (helpful for magnetometer, accelerometer, …)
    - temperature sensor helps with bias estimation for all sensors
    - 2D or RGBD cameras for positional tracking (track landmarks)

Google Project Tango



Lim et al., ICRA 2014

Extended Kalman Filter – Outlook

- Visual SLAM tutorial: http://frc.ri.cmu.edu/~kaess/vslam_cvpr14/
- 2005 DARPA grand challenge – Stanford's Stanley used a host of sensors, including 5 LIDAR sensors, GPS, gyros, accelerometers, video camera, …

Additional Information

TinyEKF: great generic implementation of Extended Kalman Filter: github.com/simondlevy/TinyEKF

All the basics we discussed today with more details

- S. Thrun, W. Burgard, D. Fox "Probabilistic Robotics", MIT Press 2005, chapter 3
- G. Welch, G. Bishop "An Introduction to the Kalman Filter", UNC Tech. Report TR 95-041, 2006
- C. Stachniss "SLAM course", online video lectures on youtube.com, lectures 1-4 (https://www.youtube.com/watch?v=U6vr3iNrwRA&list=PLgnQpQtFTOGQrZ4O5QzbIHgl3b1JHimN_)

More advanced:

- E. Kraft "A Quaternion-based Unscented Kalman Filter for Orientation Tracking", IEEE Proc. Information Fusion, 2003
- A. Mourikis, S. Roumeliotis "A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation", Proc. ICRA 2007