Kalman Filters

Colin Shaw
04.27.2017

1. **Introduction**
    a. Welcome to Computer Science Club
    b. Thank you to RevUnit for sponsoring

2. **What are Kalman filters used for?**
    a. General time series analysis
        i. Econometrics
    b. Motion planning
        i. Guidance
        ii. Navigation
        iii. Apollo lander and nuclear missiles
    c. Sensor fusion
        i. Lidar
        ii. Radar

3. **How Kalman filters fit in the larger world**
    a. Measurement
        i. Measuring the world around you
        ii. Coordinates relative to you
        iii. Examples
            1. Kalman filters
                a. Plain (linear) Kalman filters
                b. Extended Kalman filters
                c. Unscented Kalman filters
            2. Double Exponential Smoothing
            3. Recursive Total Least Squares
    b. Localization
        i. Measuring your position in the world
        ii. Your position in larger map coordinates
        iii. Examples
            1. Particle filters
            2. Histogram filters
    c. Both measurement *and* localization
        i. Simultaneous Localization and Measurement (SLAM)

4. **Torrid history of the introduction of the Kalman filter**
    a. Rudolph Emil Kalman's 1960 paper
        i. Hidden model was looked at with great skepticism
        ii. Not met with open arms in the community
        iii. EE journals would not publish it (ended up in ME)
    b. Unscented Kalman filters
        i. Met with skepticism
        ii. Again hard to publish

5. **What type of process is a Kalman filter?**
    a. Linear quadratic estimator
        i. Linear system
        ii. RMSE can be minimized by minimizing quadratic
    b. Bayesian estimator
        i. Prior
        ii. Distribution
        iii. Posterior
        iv. Conjugate priors
    c. Markov process
        i. Future probabilities determined by most recent state

6. **How are we presenting Kalman filters here?**
    a. Developing intuition about a basic motion model
    b. Look at what we need to use the motion model
    c. Discuss noisy process and measurements
    d. Jump to talking about probability distributions
    e. Show the full set of Kalman filter equations
    f. Walk through a Kalman filter code example
    g. Discuss more complex motion models and sensor fusion
    h. Discuss deviation from a linear model

7. **Intuition regarding the motion model**
    a. Draw measurements at time **t** = 0, 1, 2
    b. Infer measurement at time **t** = 3
    c. How do we know where the next point is?
        i. Mental model is more than the measurements
        ii. It has implied velocity
    d. How do we correct for errors
        i. Error is the actual minus predicted (residual)
        ii. Apply scaled residual to update model

8. **More formality to the motion model**
    a. Vector of position x and velocity v (or $\dot{x}$)

    $$\mathbf{x}_k = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$$

    b. State updates occur with a state transition function

    $$\mathbf{F} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$

    c. We can only observe measurables, so there is a measurement function **H**

    $$\mathbf{H} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

    d. The errors can be assumed to be *acceleration* noise
        i. There is a force causing the change in velocity
        ii. There is an acceleration associated with the force
        iii. This noise is distributed N(0, σ_a^2)

9. **Adding the notion of noise**
    a. What if we had noisy measurements only
        i. For each **t** = 0, 1, 2, … we assume position error is distributed N(0, σ_x^2)
        ii. What does this give us?
            1. The standard deviation at these positions
            2. In effect, nothing
        iii. We need more information to be able to refine estimates of uncertainty
            1. Think about the Bayesian estimation notion of the problem
            2. We can use a prior and a new distribution to compute a posterior

10. **Brief probability distribution review**
    a. Gaussian.ipynb
    b. What is a normal distributed random variable
        i. Mean
        ii. Standard deviation
    c. Adding normally distributed random variables
        i. Analytical result
            1. $\mu = \mu_1 + \mu_2$
            2. $\sigma^2 = \sigma_1^2 + \sigma_2^2$
        ii. Pointwise addition
        iii. Convolution (ConvolutionAddition.ipynb)
    d. Multiplying Gaussians
        i. Analytical result
            1. $\mu = \mu_1 \sigma_2^2 + \mu_2 \sigma_1^2 / \sigma_1^2 + \sigma_2^2$ (normalized weighted average)
            2. $\sigma^2 = \sigma_1^2 \sigma_2^2 / \sigma_1^2 + \sigma_2^2$ (parallel sum)
        ii. PDF demonstration

11. **Kalman filter model**
    a. Prediction
        i. Predicted State Estimate

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k$$

        ii. Predicted Estimate Covariance

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^{\mathrm{T}} + \mathbf{Q}_k$$

    b. Measurement
        i. Innovation Residual

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}$$

        ii. Innovation Covariance

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k$$

        iii. Optimal Kalman Gain

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^{\mathrm{T}} \mathbf{S}_k^{-1}$$

        iv. Updated State Estimate

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k$$

        v. Updated Covariance Estimate

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}$$

12. **Intuition about what the Kalman filter is doing**
    a. PositionVelocity.ipynb
    b. Steps
        i. Initial position estimate (poor velocity knowledge)
        ii. Project to arbitrary velocity distribution
        iii. New measurement (decent position, poor velocity knowledge)
        iv. Posterior distribution now has much better position and velocity
        v. Repeat
    c. Basic idea
        i. Additive noise (both prediction and measurement)
        ii. Multiplicative refinement in the Kalman gain calculation

13. **Measuring error**
    a. Root Mean Squared Error (RMSE)
        i. How to compute

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

        ii. Is not normalized
        iii. Depends on knowing ground truth
    b. Normalized Innovation Squared (NIS)
        i. ChiSquared.ipynb
        ii. How to compute

$$NIS_k = \tilde{y}_k^T S_k{}^{-1} \tilde{y}_k$$

        iii. Is normalized
        iv. Does not depend on ground truth

14. **Kalman filter code example**
    a. KalmanFilter.ipynb
    b. Numpy for matrix arithmetic
    c. Steps
        i. Initialize state and covariance
        ii. Compute state prediction
        iii. Update with measurement
        iv. Repeat
    d. Error estimation

15. **More complex motion models**
    a. Sensor fusion
        i. Merge two (or more) types of measurements
        ii. Get better results than any provide alone
    b. Example
        i. Radar
            1. Radius (inaccurate)
            2. Radial velocity (accurate)
            3. Angle (inaccurate)
        ii. Lidar
            1. Radius (accurate)
            2. Angle (accurate)
    c. Measurement and state transfer function in general could be nonlinear

16. **Deviation from the linear Kalman filter model**
    a. GaussianTransformation.ipynb
    b. Extended Kalman filters
        i. Linearize the nonlinear mappings
        ii. Jacobian evaluation instead of matrix-vector multiplication
        iii. Can be expensive to compute
        iv. Not so good results in difficult cases
            1. Very nonlinear functions
            2. Non-normal distributions
    c. Unscented Kalman filters
        i. Sample sigma points
        ii. Map sigma points
        iii. Estimator of distribution without knowing specifics
        iv. Relatively inexpensive to compute
        v. More stable results in most cases