Final project – Fall 2015
Scientific Computing
Daniel Hallman and Maike Scherer

The Discrete Kalman Filter

Overview

- Rudolf Emil Kalman
- Use of the Kalman Filter
- Theory behind the Kalman Filter
- Application
- Works Cited

Rudolf Emil Kalman

- Born in 1930
- Electrical Engineer (undergrad) and Mathematician (graduate, PhD)
- Papers in physics, differential equations, prediction theory (statistics)
  → Kalman filter, 1960
- Received a prize for his work from the AMS

Use of the Kalman Filter

- Signal processing
  - GPS data
- Computer Vision
  - Object Tracking
- Image Processing
  - De-noise i.e. filter images
- ...anything that has numbers that needs to be filtered

Note: there is a lot of MATLAB documentation

Idea and Assumptions

- Discrete
  - measurements taken with equal distances, use as unity to "keep the Math rigorous yet elementary" (Kalman)
- Educated guess (Prediction)
- Actual (noisy) data updates prediction (Correction)
  - Only need to keep track of one prior step (fast)
- Everything is related to a certain degree
  - Minimize Estimated Error Covariance (trial and error, update educated guess)
- Assume Gaussian noise (white noise) i.e. the noise has a normal distribution

Notation and Equations

- $\hat{x}_k^-$ *predicted* estimate at time k
- $\hat{x}_k$ estimate obtained by the Kalman filter
- z_k actual measurement obtained at time k
- x_k true state of the system
- Error of the filter: $e_k = x_k - \hat{x}_k$
- Error of the predicted values: $e_k^- = x_k - \hat{x}_k^-$
- The estimate error covariance: $P_k = E[e_k e_k^T]$
- The prediction error covariance: $P_k^- = E[e_k^- e_k^{-T}]$.

Theory

- True answer: $x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}$
- Prediction (*also*: Time Update)

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$$

- Correction (*also*: Measurement update)

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$$

- *K* – Kalman gain

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}$$

Theory

- Time Update

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$$
$$P_k^- = AP_{k-1}A^T + Q$$

- Measurement update

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}$$
$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$$
$$P_k = (I - K_k H)P_k^-$$

Application I

- Tracking an object under constant acceleration

- Equations of motion are used as the prediction model

$$\begin{aligned} x(t) &= x_0 + \dot{x}t + \frac{1}{2}at^2 \\ \dot{x}(t) &= \dot{x}_0 + at \end{aligned}$$

- Discretize model

$$\begin{aligned} x_k &= x_{k-1} + \dot{x}_{k-1}\mathrm{dt} + \frac{1}{2}a\mathrm{dt}^2 \\ \dot{x}_k &= \dot{x}_{k-1} + a\mathrm{dt} \end{aligned}$$

- Matrix-vector notation

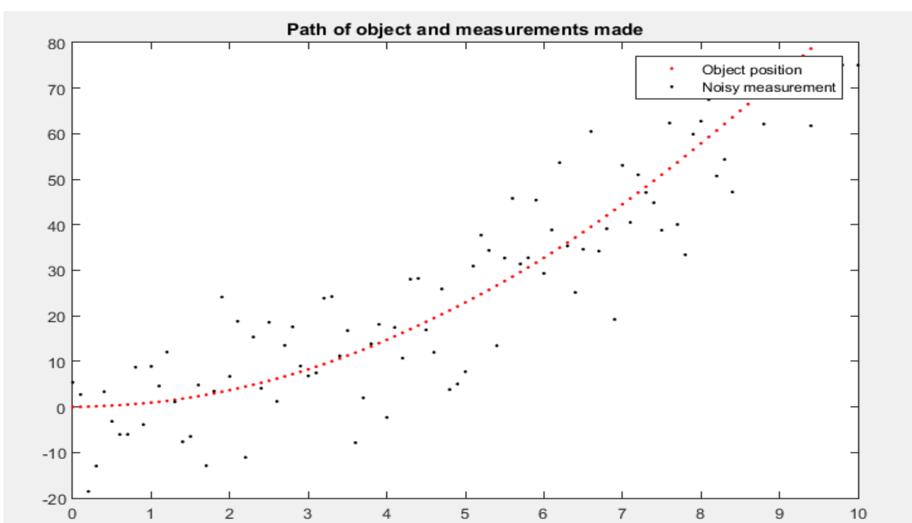
$$\begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix} = \begin{bmatrix} 1 & \mathrm{dt} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ \dot{x}_{k-1} \end{bmatrix} + \begin{bmatrix} \mathrm{dt}^2/2 \\ \mathrm{dt} \end{bmatrix} a$$

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$$

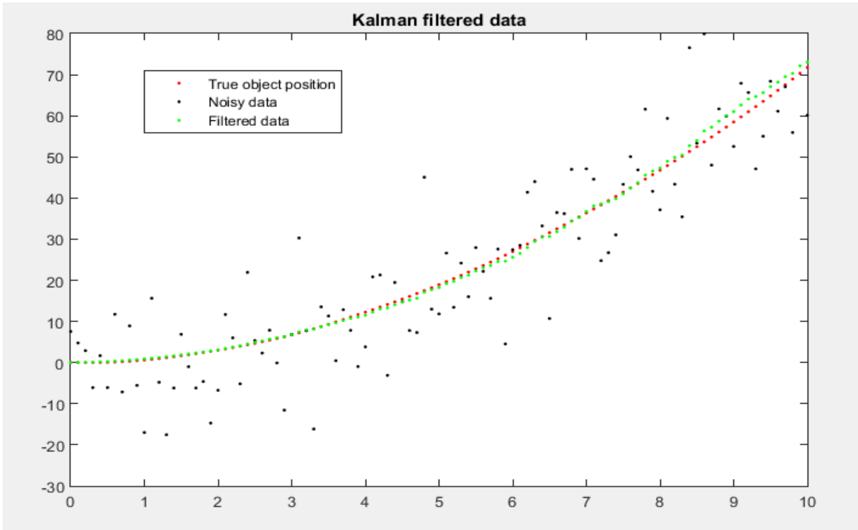
MATLAB excerpt

```
% Kalman filter for tracking an object
for t = 1: length(true_x_full)
    % Time Update
    % Physical model
    x_prediction = A*x_prediction + B*u;
    P = A*P*A' + Q;
    % Q is the 'true object path noise' and is a number we make up

    % Measurement Update
    K = P*H'*inv(H*P*H'+R);
    % R is the noise of our measurement instrument and is a made up number
    x_prediction = x_prediction + K*(measurement(t)-H*x_prediction);
    % H simply plucks the position entry out of the estimate vector
    P = (eye(2)-K*H)*P;

    % record where you think the object is
    x_loc_estimate = [x_loc_estimate; x_prediction(1)];
end
```

Result



Result



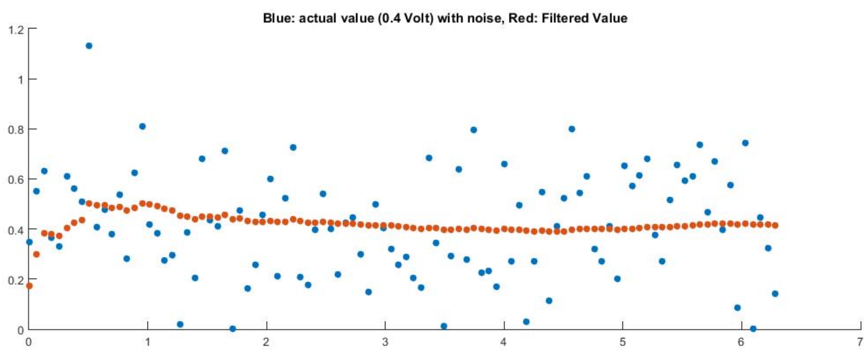
Application II

- De-noising of Voltage Response (constant)
  - Assume σ = 0.1 V
  - 1 – dimensional
  - No control signal
  - A = 1 (next value will be the same as previous)
  - H = 1
  - Initial estimate $\hat{x}_k^-$ = 0
  - $P_0 = 1$ (since there will be some noise)

MATLAB excerpt

```
% Kalman loop
x = zeros(1, n+1);
x(1) = x0;
P = P0;
for i = 1:n
     %time update
   x(i+1) = x(i); % prediction

   % measurement update
   K = P/(P+R); % Kalman gain update
   x(i+1) = x(i+1) + K*(z(i)-x(i+1)); % measurement update
   P = (1-K)*P; % Error covariance update
end
```

Results



Works Cited

- http://www.genealogy.math.ndsu.nodak.edu/id.php?id=13021 *last accessed 12/12/15*
- http://www-history.mcs.st-andrews.ac.uk/Biographies/Kalman.html *last accessed 12/12/15*
- http://www.mathworks.com/discovery/kalman-filter.html *last accessed 12/14/15*
- http://bilgin.esme.org/BitsBytes/KalmanFilterforDummies.aspx *last accessed 12/15/15*
- http://www.cs.unc.edu/~tracker/media/pdf/SIGGRAPH2001_CoursePack_08.pdf *last accessed 12/15/15*
- http://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/ *last accessed 12/15/15*
- R. E. Kalman – A New Approach to Linear Filtering and Prediction Problems *(1960)*
- Greg Welch and Gary Bishop – An Introduction to Kalman Filter *(2006)*
- https://www.youtube.com/watch?v=rUgKnoiRoYo *last accessed 12/12/15*