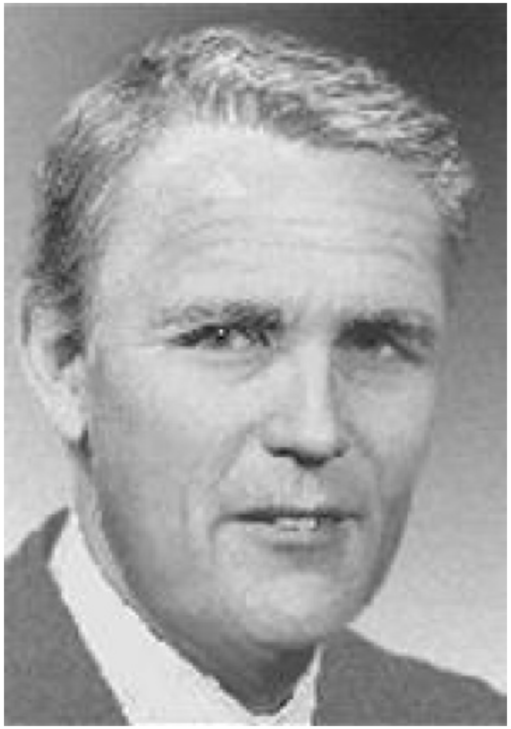
Kalman Filter

Zhang Hongxin
zhx@cad.zju.edu.cn

State Key Lab of CAD&CG, ZJU
2009-03-12

Rudolf Emil Kalman

- Born 1930 in Hungary
- BS and MS from MIT
- PhD 1957 from Columbia
- Filter developed in 1960-61
- Now retired



What is a Kalman Filter?

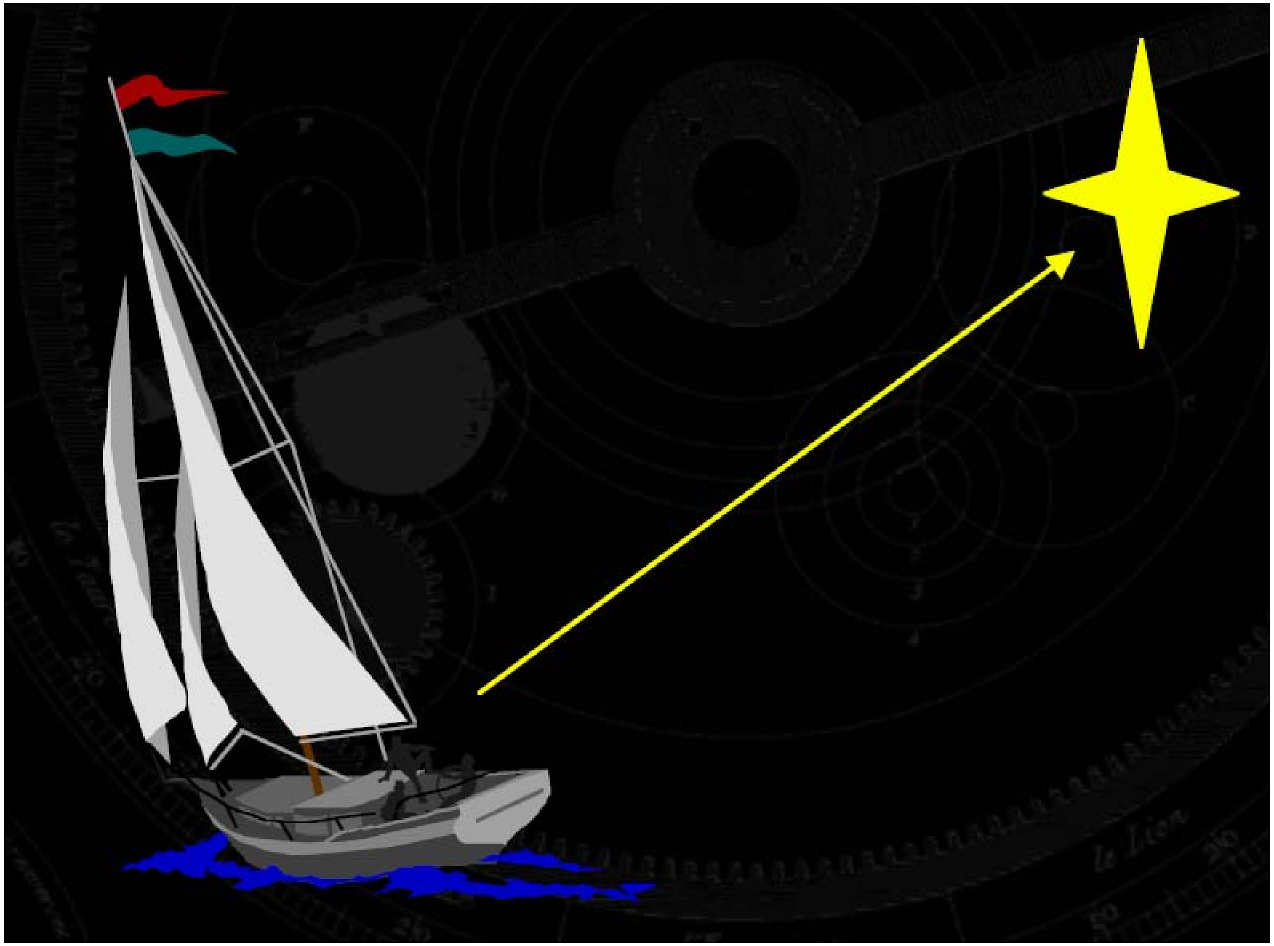
- Just some applied math.
- A linear system: f(a+b) = f(a) + f(b).
- Noisy data in :: hopefully less noisy out.
- But delay is the price for filtering...
- Pure KF does not even adapt to the data.

- An “optimal recursive data processing algorithm”

What is it used for?

- Tracking missiles
- Tracking heads/hands/drumsticks
- Extracting lip motion from video
- Fitting Bezier patches to point data
- Economics
- Navigation

A really simple example



The Process to be Estimated

- Discrete-time controlled process
  - State estimation:

$$\mathbf{x}_k = A\mathbf{x}_{k-1} + B\mathbf{u}_{k-1} + \underline{\mathbf{w}_{k-1}} \qquad \mathbf{x}_k \in \Re^n$$

  - Measurement:

$$\mathbf{z}_k = H\mathbf{x}_{k-1} + \underline{\mathbf{v}_{k-1}} \qquad \mathbf{z}_k \in \Re^m$$

    - Process noise covariance: Q

$$p(\mathbf{w}) \sim N(0, Q)$$

    - Measurement noise covariance: R

$$p(\mathbf{v}) \sim N(0, R)$$

The computational Origins of the Filters

- Priori state estimation error at step k

$$\mathbf{e}_k^- := \mathbf{x}_k - \hat{\mathbf{x}}_k^- \qquad P_k^- = E[\mathbf{e}_k^- \mathbf{e}_k^{-T}]$$

- Posteriori estimation error

$$\mathbf{e}_k := \mathbf{x}_k - \hat{\mathbf{x}}_k \qquad P_k = E[\mathbf{e}_k \mathbf{e}_k^{T}]$$

- Posteriori as a linear combination of a Priori

$$\mathbf{x}_k = A\mathbf{x}_{k-1} + B\mathbf{u}_{k-1} + \mathbf{w}_{k-1}$$
$$\mathbf{z}_k = H\mathbf{x}_{k-1} + \mathbf{v}_{k-1}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K(\mathbf{z}_k - H\hat{\mathbf{x}}_k^-)$$

Measurement *innovation* or *residual*

The computational Origins of the Filters

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \underline{K}(\mathbf{z}_k - H\hat{\mathbf{x}}_k^-)$$

- The *gain* or *blending factor* that minimizes the a posteriori error covariance $P_k = E[\mathbf{e}_k \mathbf{e}_k^T]$

$$K_k = \frac{P_k^- H^T}{H P_k^- H^T + R}$$

$$\lim_{R \to 0} K_k = H^{-1} \qquad \lim_{P_k^- \to 0} K_k = 0$$

The Probabilistic Origins of the Filter

$$E[\mathbf{x}_k] = \hat{\mathbf{x}}_k$$

$$E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T] = P_k$$

- The *a posteriori* state estimate $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K(\mathbf{z}_k - H\hat{\mathbf{x}}_k^-)$ reflects the mean of the state distribution
- The *a posteriori* state estimate error covariance $P_k = E[\mathbf{e}_k \mathbf{e}_k^T]$ reflects the variance of the state distribution

$$p(\mathbf{x}_k \mid \mathbf{z}_k) \sim N(E[\mathbf{x}_k], E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T])$$

$$= N(\mathbf{x}_k, P_k)$$

The Discrete Kalman Filter Algorithm

- *Time update* equations

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

$$P_k^- = AP_k A^T + Q$$

- *Measurement update* equations

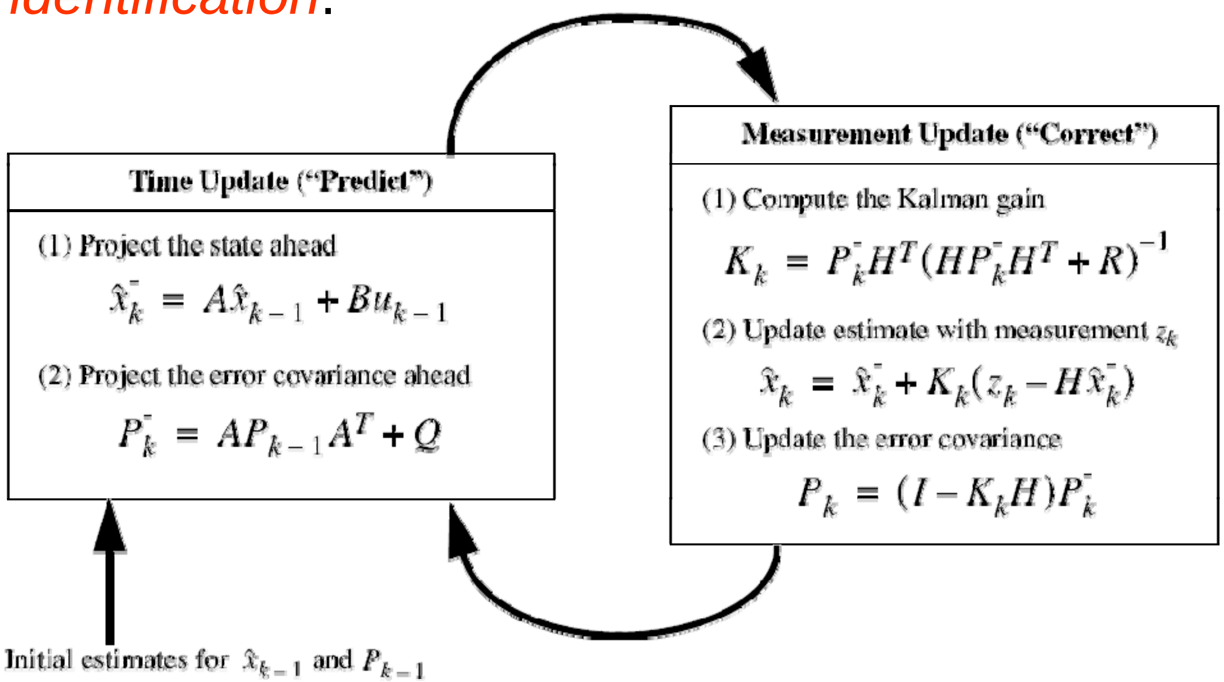
$$K_k = \frac{P_k^- H^T}{HP_k^- H^T + R}$$

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

$$P_k = (I - K_k H)P_k^-$$

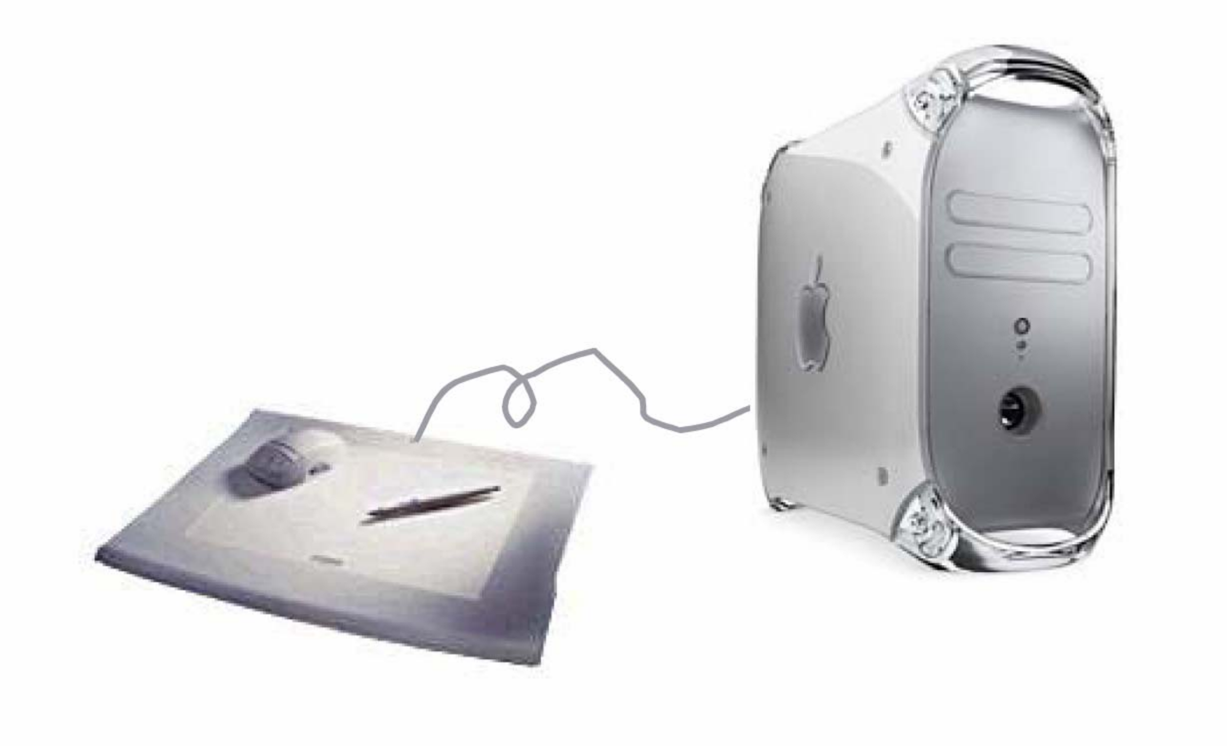
Filter Parameters and Tuning

- The measurement noise covariance R is usually measured prior to operation of the filter.
- Q and R are generally constants during filtering. Superior filter performance can be obtained by tuning them, referred to as *system identification*.



Example: 2D Position-Only

- Apparatus: 2D Tablet



Process Model

$$\mathbf{x}_k \qquad A \qquad \mathbf{x}_{k-1} \qquad \mathbf{w}_{k-1}$$

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \end{bmatrix} + \begin{bmatrix} \sim x_{k-1} \\ \sim y_{k-1} \end{bmatrix}$$

State *k* | State transition | State *k-1* | Noise

$$\mathbf{x}_k = A\mathbf{x}_{k-1} + \mathbf{w}_{k-1}$$

Measurement Model

$$\mathbf{z}_k \qquad H \qquad \mathbf{x}_{k-1} \qquad \mathbf{v}_k$$

$$\begin{bmatrix} u_k \\ v_k \end{bmatrix} = \begin{bmatrix} h_x & 0 \\ 0 & h_y \end{bmatrix} \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \begin{bmatrix} \sim u_k \\ \sim v_k \end{bmatrix}$$

Measurement *k* | Measurement matrix | State *k* | Noise

$$\mathbf{z}_k = H\mathbf{x}_k + \mathbf{v}_k$$

Preparation

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

State Transition

$$Q = E\left\{\mathbf{w} \bullet \mathbf{w}^T\right\} = \begin{bmatrix} Q_{xx} & 0 \\ 0 & Q_{yy} \end{bmatrix}$$

Process

Noise Covariance

$$R = E\left\{\mathbf{v} \bullet \mathbf{v}^T\right\} = \begin{bmatrix} R_{xx} & 0 \\ 0 & R_{yy} \end{bmatrix}$$

Measurement

Noise Covariance

Initialization

$$\mathbf{x}_0 = H\mathbf{z}_0$$

$$P = \begin{bmatrix} \varepsilon & 0 \\ 0 & \varepsilon \end{bmatrix}$$

Predict

$$\mathbf{x}_k^- = A\mathbf{x}_{k-1}$$

$$P_k^- = \underline{A}P_{k-1}A^T + \underline{Q}$$

transition uncertainty

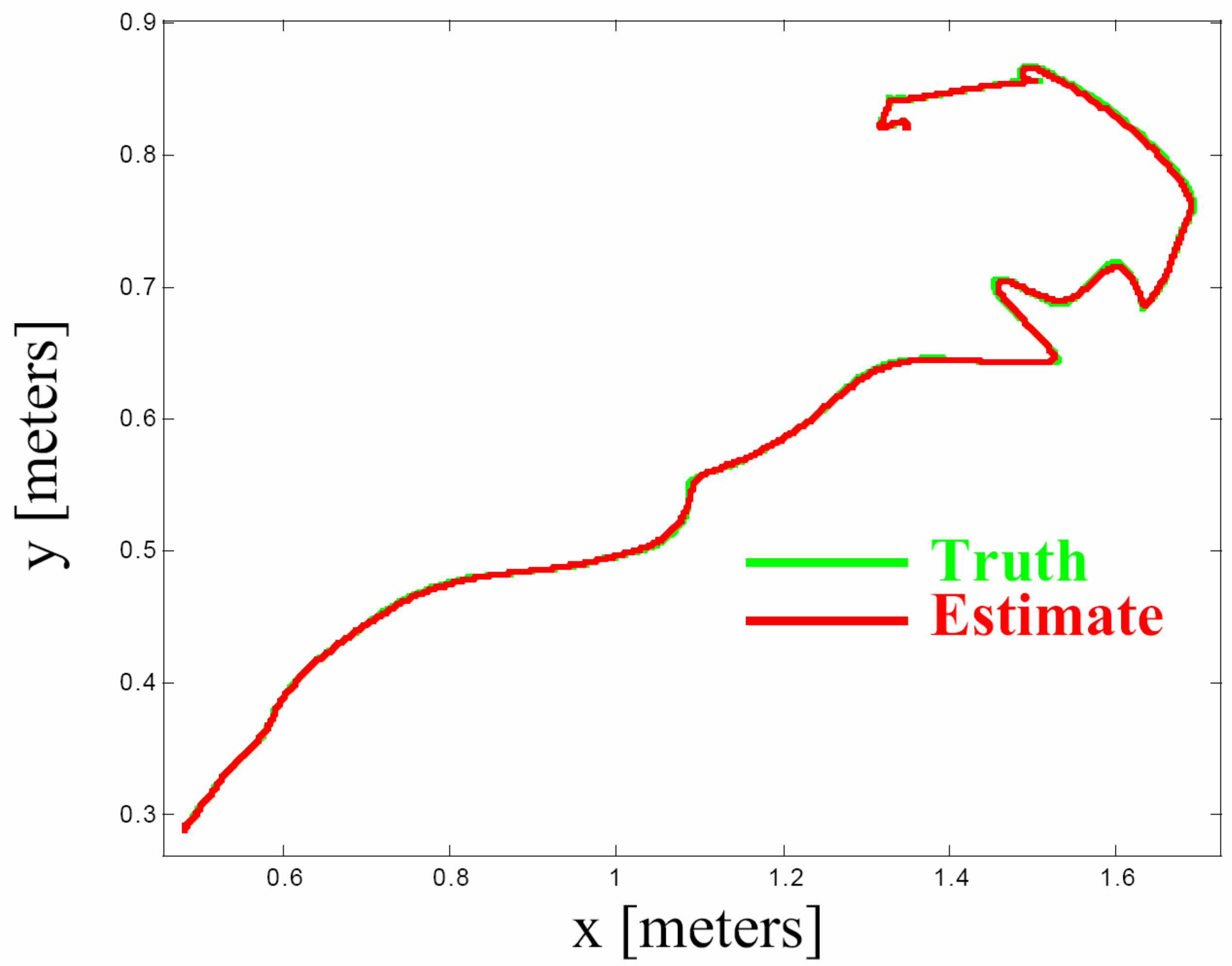
Correct

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

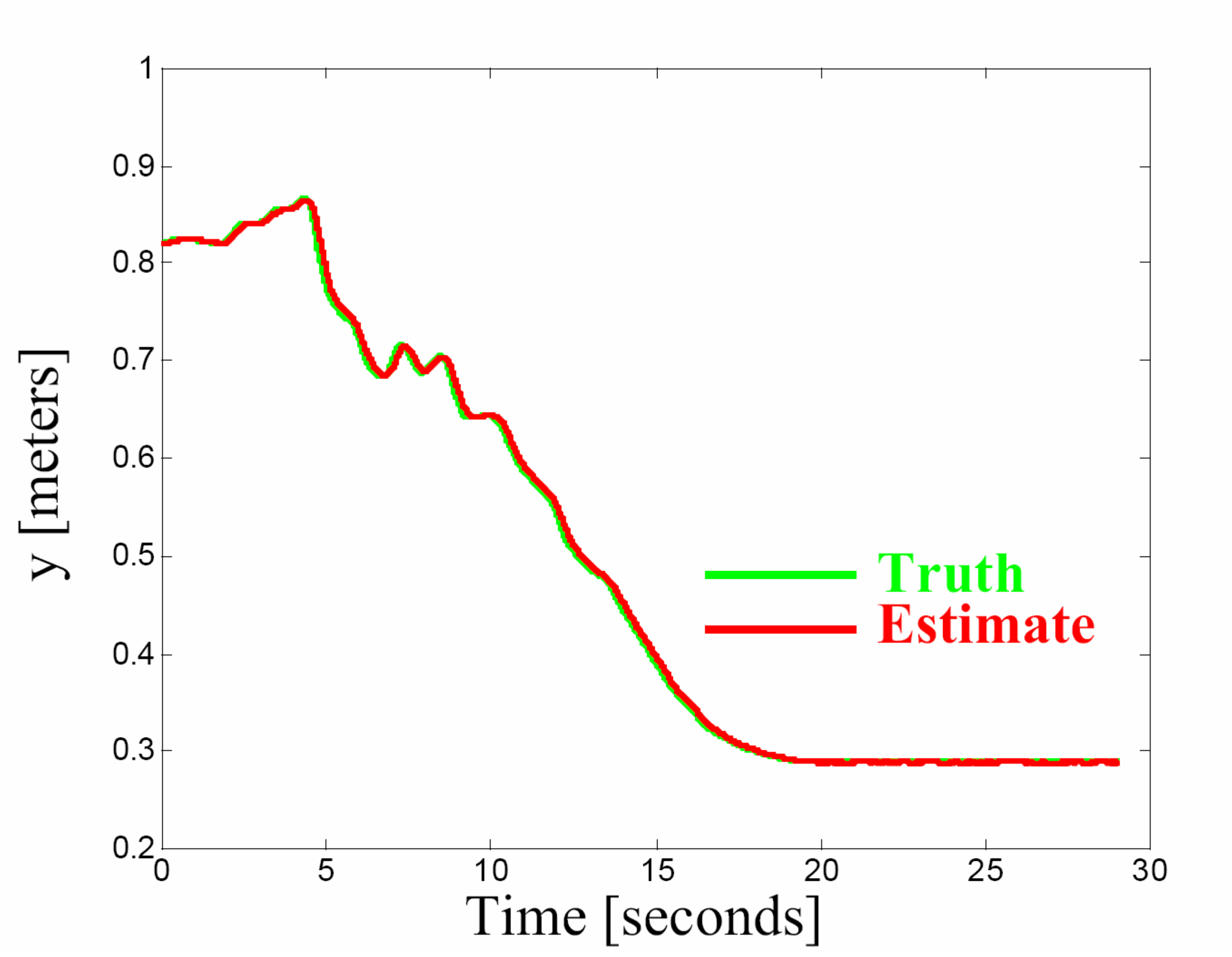
$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k (\mathbf{z}_k - H \hat{\mathbf{x}}_k^-)$$

$$P_k = (I - K_k H) P_k^-$$

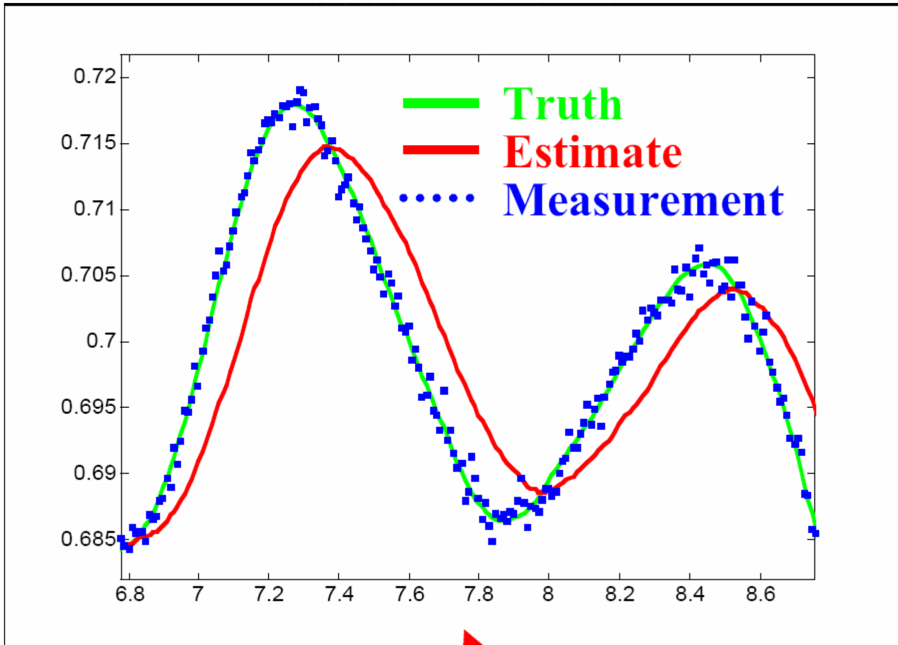
Results: XY Track

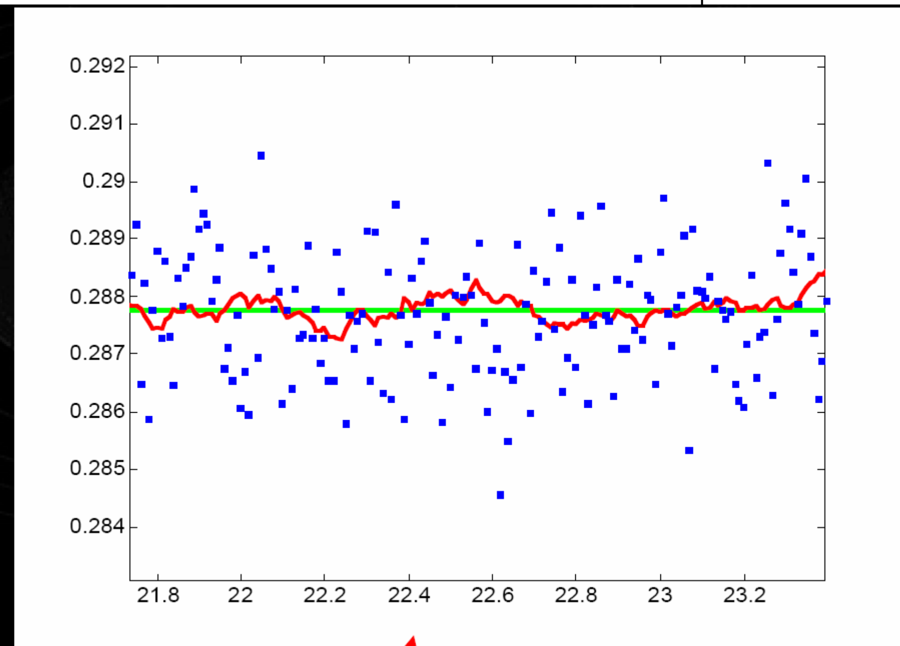


Y Track: Moving then Still

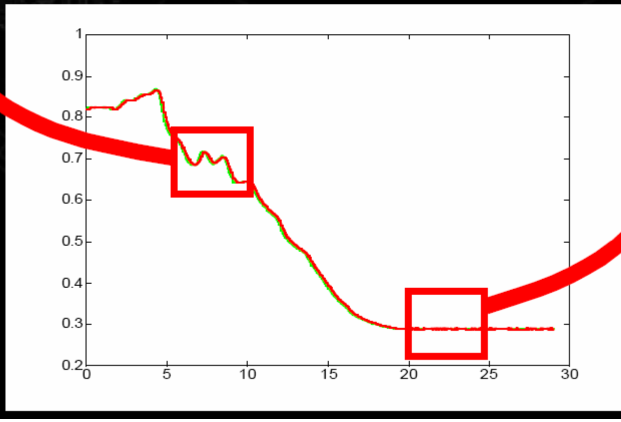


Motion-Dependent Performance





significant *latency* when moving…



…relatively *smooth* when not

The Extended Kalman Filter

- Nonlinear Process (Model)
  - Process dynamics: ***A*** becomes ***a(x)***
  - Measurement: ***H*** becomes ***h(x)***

$$\boldsymbol{x}_k = a(\boldsymbol{x}_{k-1})\boldsymbol{x}_{k-1} + B\boldsymbol{u}_{k-1} + \boldsymbol{w}_{k-1}$$
$$\boldsymbol{z}_k = h(\boldsymbol{x}_{k-1})\boldsymbol{x}_{k-1} + \boldsymbol{v}_{k-1}$$

- Filter Reformulation
  - Use functions instead of matrices
  - Use Jacobians to project forward, and to relate measurement to state

Jacobian?

- Partial derivative of measurement with respect to state

- If measurement is a vector of length M and state has length N
  - Jacobian of measurement function will be MxN matrix of numbers (not equations)

- Evaluating $h(x)$ and $Jacobian(h(x))$ at the same time mostly only cost a little additional computing time.

New Approaches

- Several extensions are available that work better than the EKF in some circumstances

Summary

- A set of mathematical equations that provides an efficient computational (recursive) means to estimate the state of a process.

- Minimizes the mean of the squared error

- Powerful:
  - supports estimations of past, present, and even future states,
  - can do so even when the precise nature of the modeled system is unknown

The End of Kalman Filter

Before the end of this course

- Many techniques I cannot mention yet:
  - Neural network
  - Graphical model
  - Genetic methods
  - …
- It is just a beginning …