Robotics Group Project - 5CCS2RGP

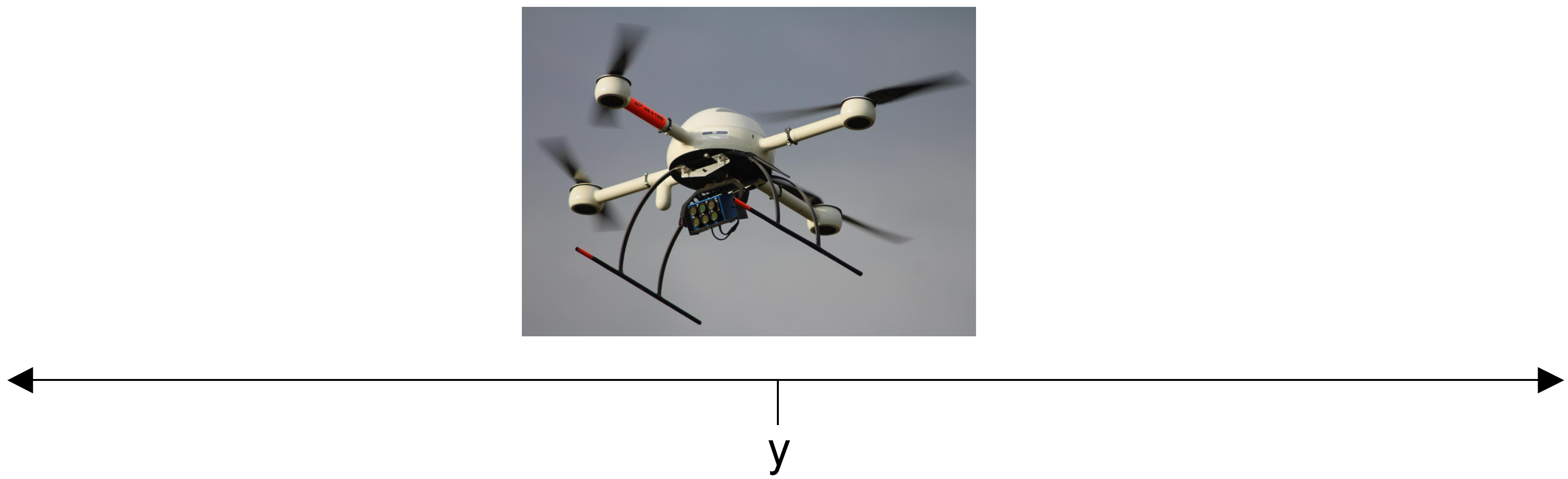
Lecture 7: Kalman Filter

Hongbin Liu, King's College London, 2013

What is a Kalman Filter?

- Recursive data processing algorithm
- Generates optimal estimate of desired quantities given the set of measurements
- Optimal?
  - For linear system and white Gaussian errors, Kalman filter is “best” estimate based on all previous measurements
  - For non-linear system optimality is ‘qualified’
- Recursive?
  - Doesn’t need to store all previous measurements and reprocess all data each time step

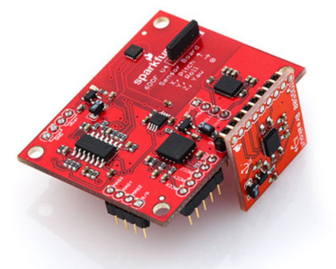
Conceptual Overview



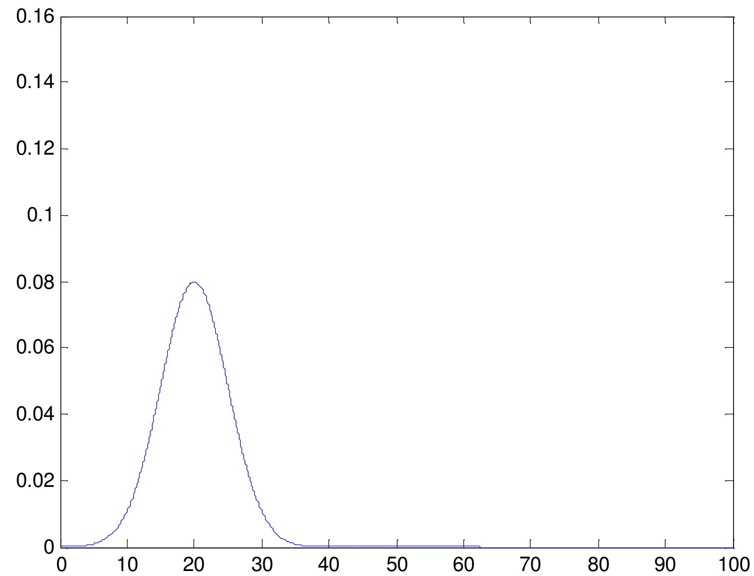
- Lost on the 1-dimensional line
- Position – $y(t)$
- Assume Gaussian distributed measurements

Conceptual Overview

State space – position
Measurement - position



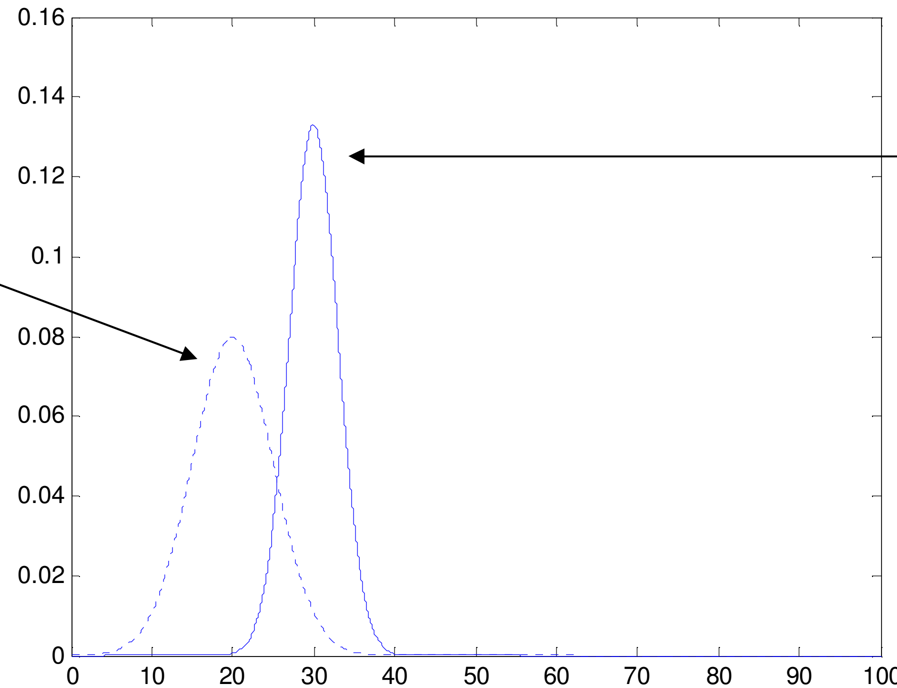
Inertial sensor (IMU)



- Measurement at t_1: Mean = z_1 and Variance = σ_{z1}
- Optimal estimate of position is: $\hat{y}(t_1) = z_1$
- Variance of error in estimate: $\sigma^2_x(t_1) = \sigma^2_{z1}$
- **Robot in same position** at time t_2 - <u>Predicted</u> position is z_1

Conceptual Overview

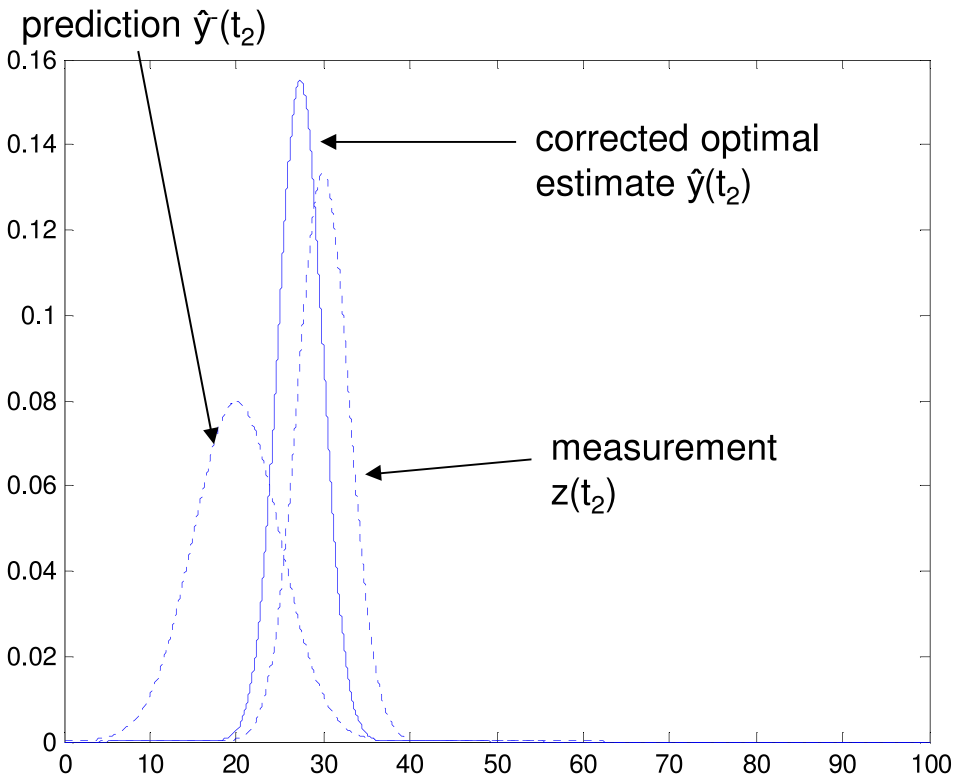
prediction $\hat{y}^{-}(t_2)$
State (by looking at the stars at t2)



Measurement usign GPS $z(t_2)$

- So we have the prediction $\hat{y}^{-}(t_2)$
- GPS Measurement at t_2: Mean = z_2 and Variance = σ_{z2}
- Need to <u>correct</u> the prediction by IMU due to measurement to get $\hat{y}(t_2)$
- Closer to more trusted measurement – should we do linear interpolation?

Conceptual Overview



Kalman filter helps you fuse measurement and prediction on the basis of how much you trust each

(I would trust the GPS more than the IMU)

- Corrected mean is the new optimal estimate of position (basically you've 'updated' the predicted position by IMU using GPS
- New variance is smaller than either of the previous two variances

Conceptual Overview (The Kalman Equations)

- Lessons so far:

Make prediction based on previous data - $\hat{y}^-$, σ^-

↓

Take measurement – z_k, σ_z

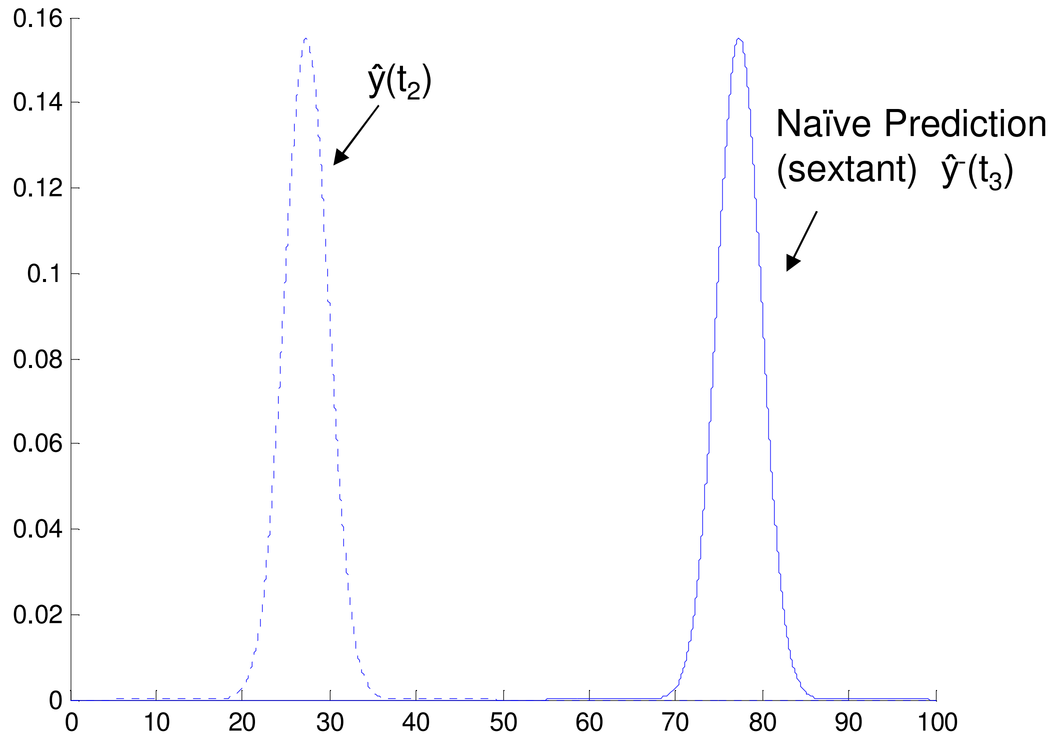
↓

Optimal estimate ($\hat{y}$) = Prediction + (Kalman Gain) * (Measurement - Prediction)

Variance of estimate = Variance of prediction * (1 – Kalman Gain)

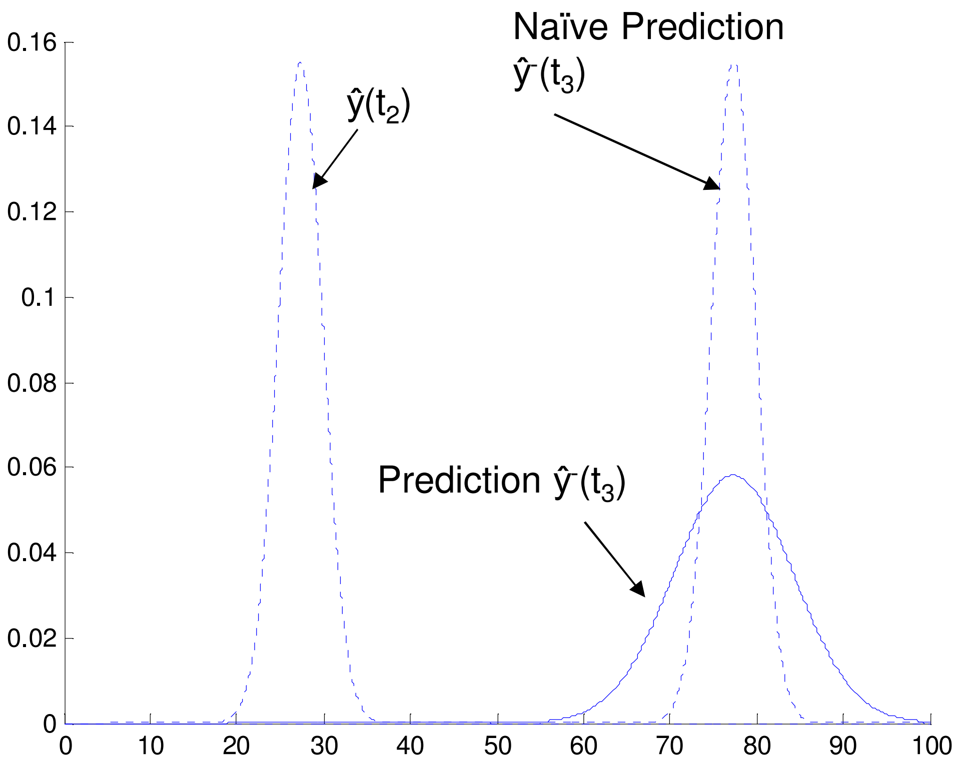
Conceptual Overview

What if the boat was now moving?



- At time t_3, robot moves with velocity dy/dt=u
- Naïve approach: Shift probability to the right to predict
- This would work if we knew the velocity exactly (perfect model)

Conceptual Overview

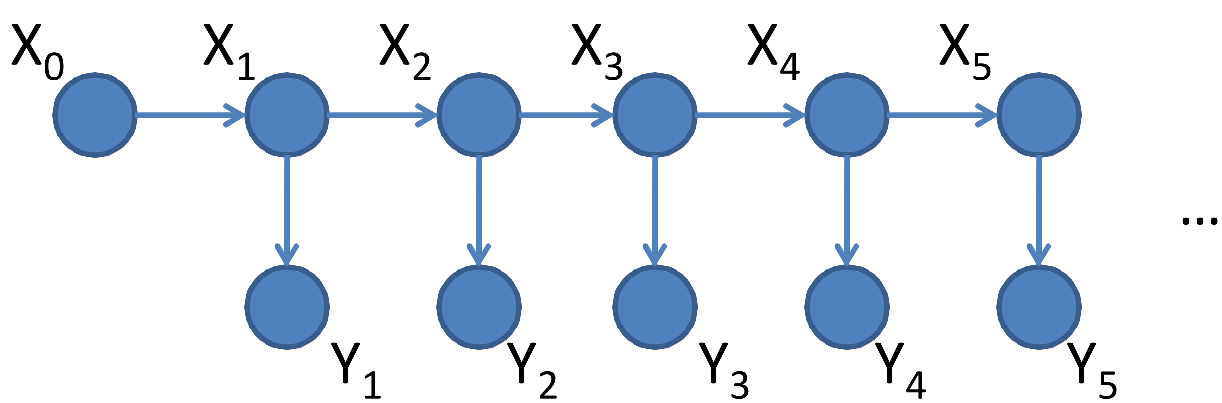


But you may not be so sure about the exact velocity

- Better to assume imperfect model by adding Gaussian noise
- dy/dt = u + w
- Distribution for prediction moves and spreads out

Recursive update of state

- Kalman filtering algorithm: repeat…
  - Time update:
    from $X_{t|t}$, compute **a priori** distrubution $X_{t+1|t}$
  - Measurement update:
    from $X_{t+1|t}$ (and given $\mathbf{y}_{t+1}$), compute **a posteriori** distribution $X_{t+1|t+1}$



Conceptual Overview

- Initial conditions ($\hat{y}_{k-1}$ and σ_{k-1})
- Prediction ($\hat{y}^{-}_{k}$, σ^{-}_{k})
  - Use initial conditions and model (eg. constant velocity) to make prediction
- Measurement (z_k)
  - Take measurement
- Correction ($\hat{y}_k$, σ_k)
  - Use measurement to correct prediction by ‘blending’ prediction and residual – always a case of merging only two Gaussians
  - Optimal estimate with smaller variance

Discrete Kalman Filter

Estimates the state x of a discrete-time controlled process that is governed by the linear stochastic difference equation

$$\boxed{y_t = A_t y_{t-1} + B_t u_t + \varepsilon_t}$$

with a measurement

$$\boxed{z_t = H_t x_t + \delta_t}$$

The set of Kalman Filtering Equations

Prediction (Time Update)

(1) Project the state ahead

$$\hat{y}^-_k = Ay_{k-1} + Bu_k$$

(2) Project the error covariance ahead

$$P^-_k = AP_{k-1}A^T + Q$$

Correction (Measurement Update)

(1) Compute the Kalman Gain

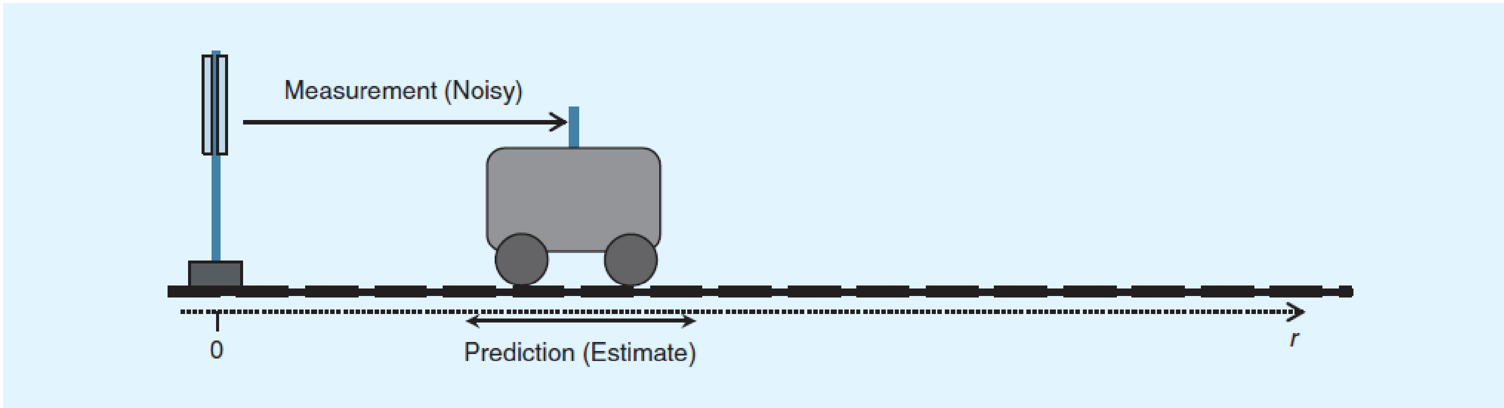
$$K = P^-_k H^T (HP^-_k H^T + R)^{-1}$$

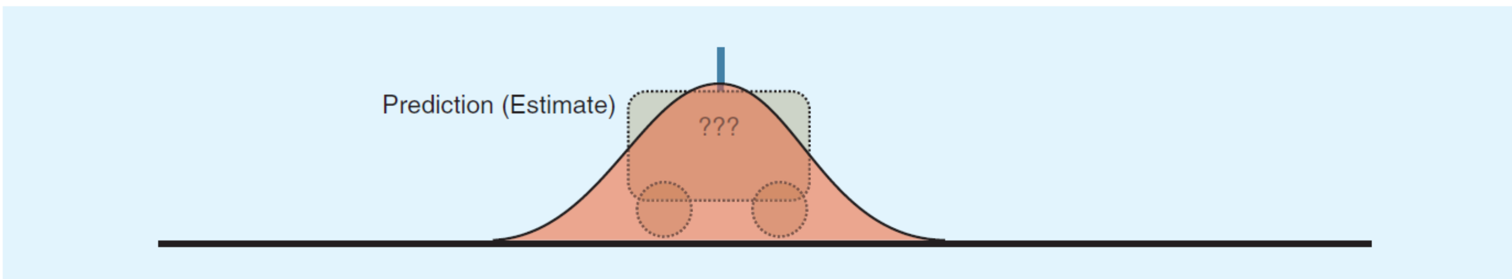
(2) Update estimate with measurement z_k

$$\hat{y}_k = \hat{y}^-_k + K(z_k - H\,\hat{y}^-_k)$$

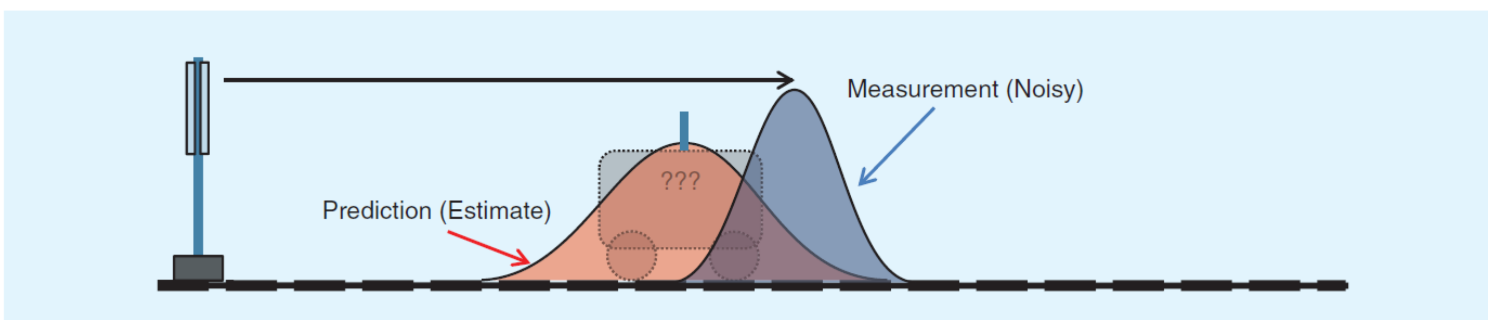
(3) Update Error Covariance

$$P_k = (I - KH)P^-_k$$

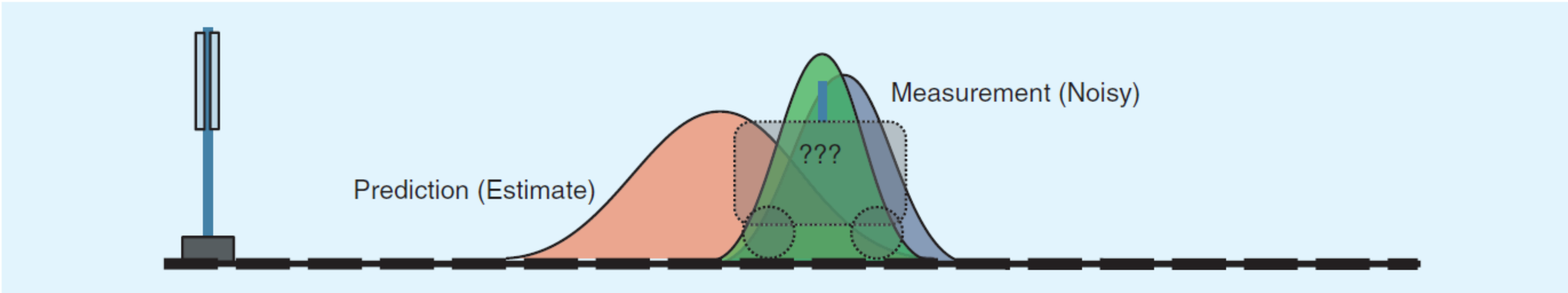




[FIG3] **Here, the prediction of the location of the train at time $t = 1$ and the level of uncertainty in that prediction is shown. The confidence in the knowledge of the position of the train has decreased, as we are not certain if the train has undergone any accelerations or decelerations in the intervening period from $t = 0$ to $t = 1$.**



[FIG4] **Shows the measurement of the location of the train at time $t = 1$ and the level of uncertainty in that noisy measurement, represented by the blue Gaussian pdf. The combined knowledge of this system is provided by multiplying these two pdfs together.**



[FIG5] **Shows the new pdf (green) generated by multiplying the pdfs associated with the prediction and measurement of the train's location at time $t = 1$. This new pdf provides the best estimate of the location of the train, by fusing the data from the prediction and the measurement.**

$$p(x) \sim N(\mu, \sigma^2):$$

$$p(u_1) = \frac{1}{\sqrt{2\pi\sigma_1}} e^{-\frac{1}{2}\frac{(x-\mu_1)^2}{\sigma_1^2}}$$

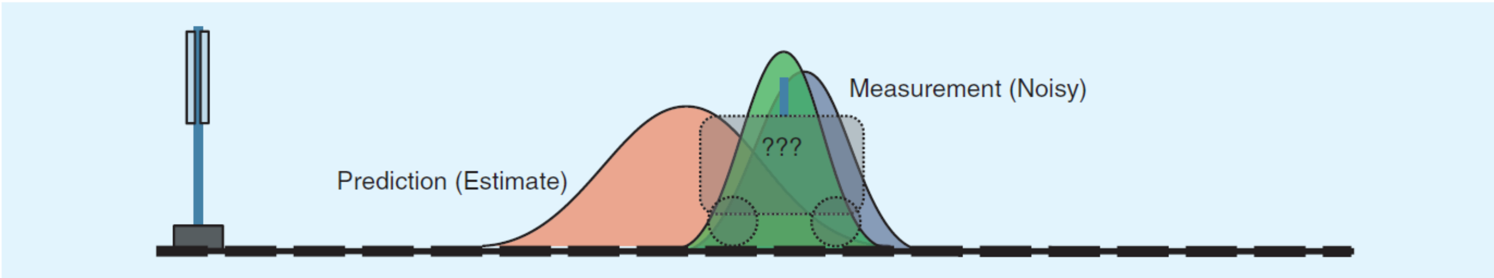
$$p(u_2) = \frac{1}{\sqrt{2\pi\sigma_2}} e^{-\frac{1}{2}\frac{(x-\mu_2)^2}{\sigma_2^2}}$$

$$p(u_1)p(u_2) = \frac{1}{2\pi\sigma_1\sigma_2} e^{-\frac{1}{2}\left[\frac{(x-\mu_1)^2}{\sigma_1^2}+\frac{(x-\mu_2)^2}{\sigma_2^2}\right]}$$

$$p(x_1)p(x_2) = \beta\frac{1}{2\pi\sigma_{fuse}} e^{-\frac{1}{2}\left[\frac{x-\mu_{fuse}}{\sigma_{fuse}}\right]^2}$$

$$\mu_{fuse} = \frac{\sigma_2{}^2}{\sigma_1{}^2+\sigma_2{}^2}\mu_1 + \frac{\sigma_1{}^2}{\sigma_1{}^2+\sigma_2{}^2}\mu_2 = \mu_1 + \sigma_1{}^2\frac{\mu_2-\mu_1}{\sigma_1{}^2+\sigma_2{}^2}$$

$$\sigma_{fuse}{}^2 = \frac{\sigma_1{}^2\sigma_2{}^2}{\sigma_1{}^2+\sigma_2{}^2}$$



[FIG5] Shows the new pdf (green) generated by multiplying the pdfs associated with the prediction and measurement of the train's location at time $t = 1$. This new pdf provides the best estimate of the location of the train, by fusing the data from the prediction and the measurement.

$$z = hx + v$$

$$z \subset N(\mu_z, \sigma_z), \sigma_z = v$$

$$\mu_2 = \frac{\mu_z}{h}$$

$$\sigma_2^{\ 2} = \frac{\sigma_z^{\ 2}}{h^2}$$

$$\mu_{fuse} = \mu_1 + \sigma_1^{\ 2} \frac{\frac{\mu_z}{h} - \mu_1}{\sigma_1^{\ 2} + \frac{\sigma_z^{\ 2}}{h^2}}$$

$$\mu_{fuse} = \mu_1 + h\sigma_1^{\ 2} \frac{\mu_z - h\mu_1}{\left(h^2\sigma_1^{\ 2} + \sigma_z^{\ 2}\right)}$$

$$\mu_{fuse} = \mu_1 + \frac{h\sigma_1^{\ 2}}{\left(h^2\sigma_1^{\ 2} + \sigma_z^{\ 2}\right)}(\mu_z - h\mu_1)$$

$$K = \frac{h\sigma_1^{\ 2}}{\left(h^2\sigma_1^{\ 2} + \sigma_z^{\ 2}\right)}$$

$$\mu_{fuse} = \mu_1 + K\left(\mu_z - h\mu_1\right)$$

$$\sigma_{fuse}^{\ \ 2} = \sigma_1^{\ 2} - Kh\sigma_1^{\ 2}$$

- $\mu_{\text{fused}} \rightarrow \hat{\mathbf{x}}_{t|t}$: the state vector following data fusion
- $\mu_1 \rightarrow \hat{\mathbf{x}}_{t|t\text{-}1}$: the state vector before data fusion, i.e., the prediction
- $\sigma^2_{\text{fused}} \rightarrow \mathbf{P}_{t|t}$: the covariance matrix (confidence) following data fusion
- $\sigma_1^2 \rightarrow \mathbf{P}_{t|t\text{-}1}$: the covariance matrix (confidence) before data fusion
- $\mu_2 \rightarrow \mathbf{z}_t$: the measurement vector
- $\sigma_2^2 \rightarrow \mathbf{R}_t$: the uncertainty matrix associated with a noisy set of measurements
- $H \rightarrow \mathbf{H}_t$: the transformation matrix used to map state vector parameters into the measurement domain

Correction (Measurement Update)

(1) Compute the Kalman Gain

$$K = P^-_k H^T (H P^-_k H^T + R)^{-1}$$

(2) Update estimate with measurement z_k

$$\hat{y}_k = \hat{y}^-_k + K(z_k - H\,\hat{y}^-_k)$$

(3) Update Error Covariance

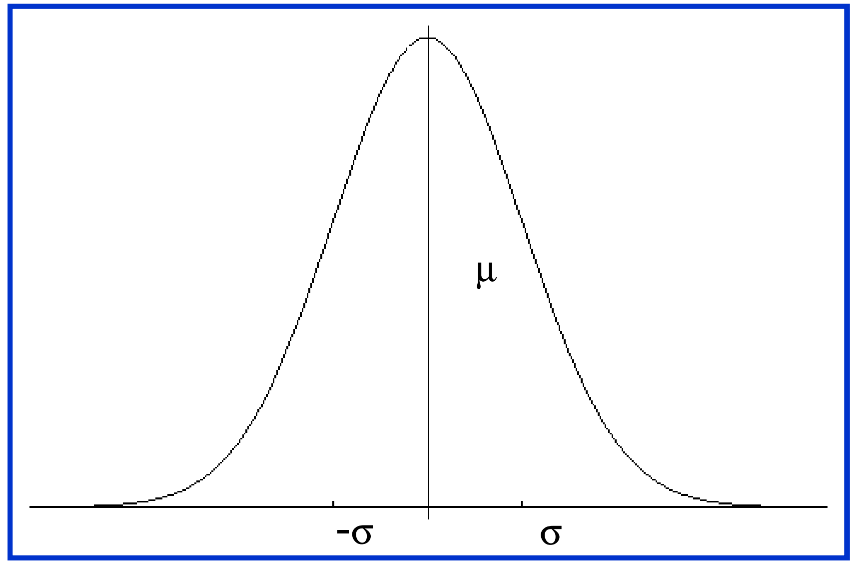
$$P_k = (I - KH)P^-_k$$

Gaussians

$$p(x) \sim N(\mu, \sigma^2):$$

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$$

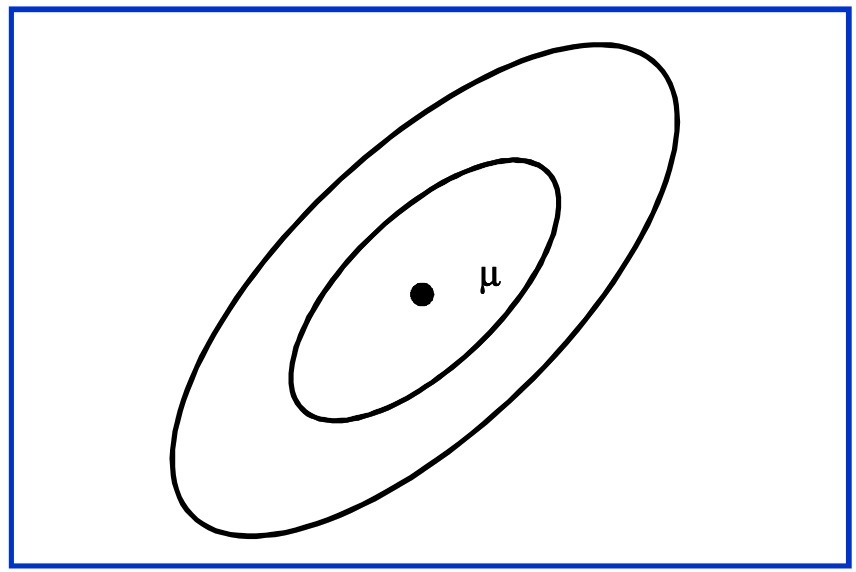
Univariate



$$p(\mathbf{x}) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}):$$

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

Multivariate



Properties of Gaussians

$$\left.\begin{array}{l} X \sim N(\mu, \sigma^2) \\ Y = aX + b \end{array}\right\} \Rightarrow \quad Y \sim N(a\mu + b, a^2\sigma^2)$$

$$\left.\begin{array}{l} X_1 \sim N(\mu_1, \sigma_1^{\,2}) \\ X_2 \sim N(\mu_2, \sigma_2^{\,2}) \end{array}\right\} \Rightarrow p(X_1) \cdot p(X_2) \sim N\left(\frac{\sigma_2^{\,2}}{\sigma_1^{\,2} + \sigma_2^{\,2}} \mu_1 + \frac{\sigma_1^{\,2}}{\sigma_1^{\,2} + \sigma_2^{\,2}} \mu_2, \quad \frac{1}{\sigma_1^{\,-2} + \sigma_2^{\,-2}} \right)$$

Multivariate Gaussians

$$\left.\begin{array}{l} X \sim N(\mu, \Sigma) \\ Y = AX + B \end{array}\right\} \Rightarrow \quad Y \sim N(A\mu + B, A\Sigma A^T)$$

$$\left.\begin{array}{l} X_1 \sim N(\mu_1, \Sigma_1) \\ X_2 \sim N(\mu_2, \Sigma_2) \end{array}\right\} \Rightarrow p(X_1) \cdot p(X_2) \sim N\left(\frac{\Sigma_2}{\Sigma_1 + \Sigma_2}\mu_1 + \frac{\Sigma_1}{\Sigma_1 + \Sigma_2}\mu_2, \quad \frac{1}{\Sigma_1^{-1} + \Sigma_2^{-1}}\right)$$

- We stay in the "Gaussian world" as long as we start with Gaussians and perform only linear transformations.

Up To Higher Dimensions

- Our previous Kalman Filter discussion was of a simple one-dimensional model.
- Now we go up to higher dimensions:
  - State vector: $\mathbf{x} \in \Re^n$
  - Sense vector: $\mathbf{z} \in \Re^m$
  - Motor vector: $\mathbf{u} \in \Re^l$
- First, a little statistics.

Expectations

- Let x be a random variable.
- The expected value $E[x]$ is the mean:

$$E[x] = \int x p(x) dx \approx \bar{x} = \frac{1}{N} \sum_{1}^{N} x_i$$

  – The probability-weighted mean of all possible values. The sample mean approaches it.

- Expected value of a vector **x** is by component.

$$E[\mathbf{x}] = \bar{\mathbf{x}} = [\bar{x}_1, \cdots \bar{x}_n]^T$$

Variance and Covariance

- The variance is $E[\ (x\text{-}E[x])^2\ ]$

$$\sigma^2 = E[(x-\bar{x})^2] = \frac{1}{N}\sum_{1}^{N}(x_i - \bar{x})^2$$

- Covariance matrix is $E[\ (\mathbf{x}\text{-}E[\mathbf{x}])(\mathbf{x}\text{-}E[\mathbf{x}])^T\ ]$

$$C_{ij} = \frac{1}{N}\sum_{k=1}^{N}(x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j)$$

  – Divide by N–1 to make the sample variance an *unbiased estimator* for the population variance.

Biased and Unbiased Estimators

- Strictly speaking, the sample variance

$$\sigma^2 = E[(x-\overline{x})^2] = \frac{1}{N}\sum_1^N (x_i - \overline{x})^2$$

is a biased estimate of the population variance. An unbiased estimator is:

$$s^2 = \frac{1}{N-1}\sum_1^N (x_i - \overline{x})^2$$

- **But**: *"If the difference between N and N–1 ever matters to you, then you are probably up to no good anyway …"* [Press, et al]

Covariance Matrix

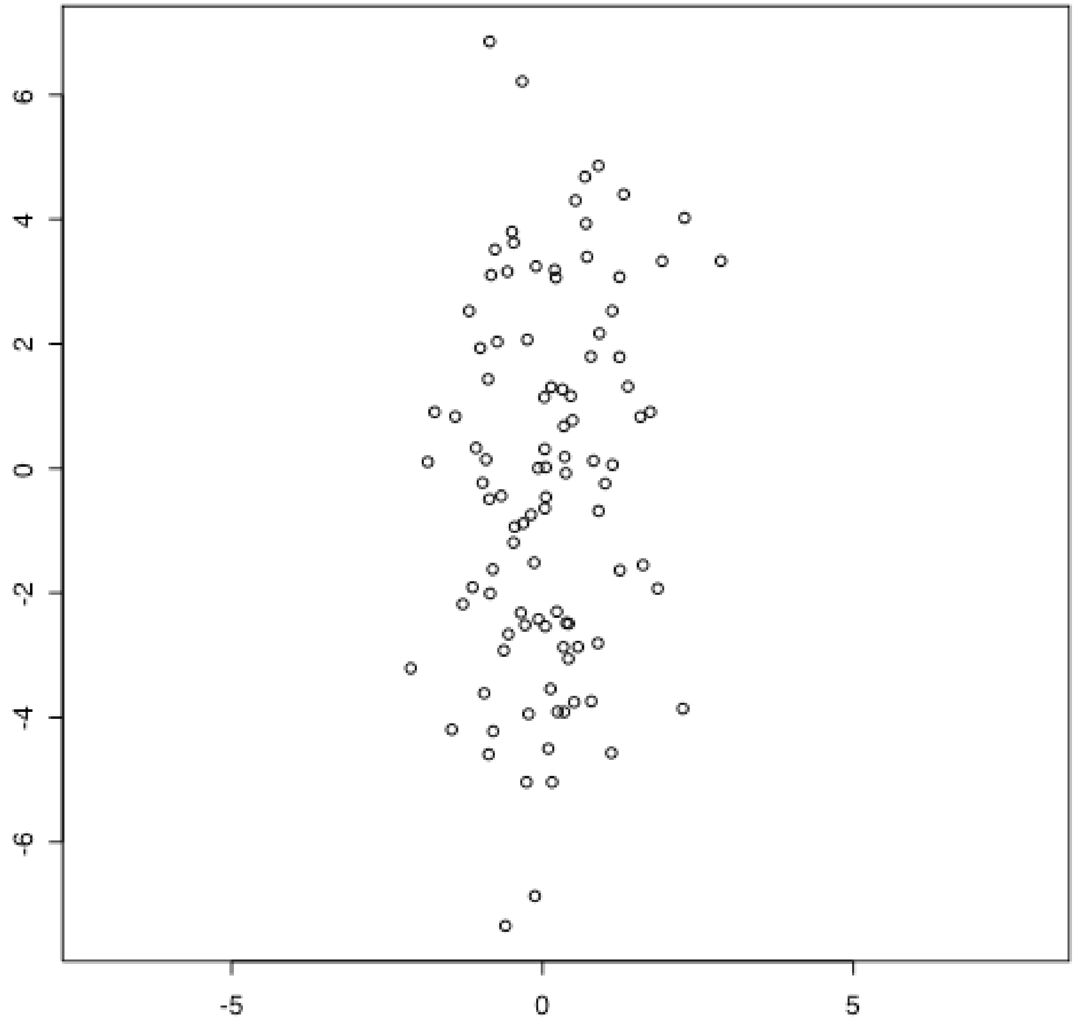
- Along the diagonal, C_{ii} are variances.
- Off-diagonal C_{ij} are essentially correlations.

$$\begin{bmatrix} C_{1,1} = \sigma_1^2 & C_{1,2} & & C_{1,N} \\ C_{2,1} & C_{2,2} = \sigma_2^2 & & \\ & & \ddots & \vdots \\ C_{N,1} & & \cdots & C_{N,N} = \sigma_N^2 \end{bmatrix}$$

Independent Variation

- x and y are Gaussian random variables (N=100)
- Generated with $\sigma_x=1 \quad \sigma_y=3$
- Covariance matrix:

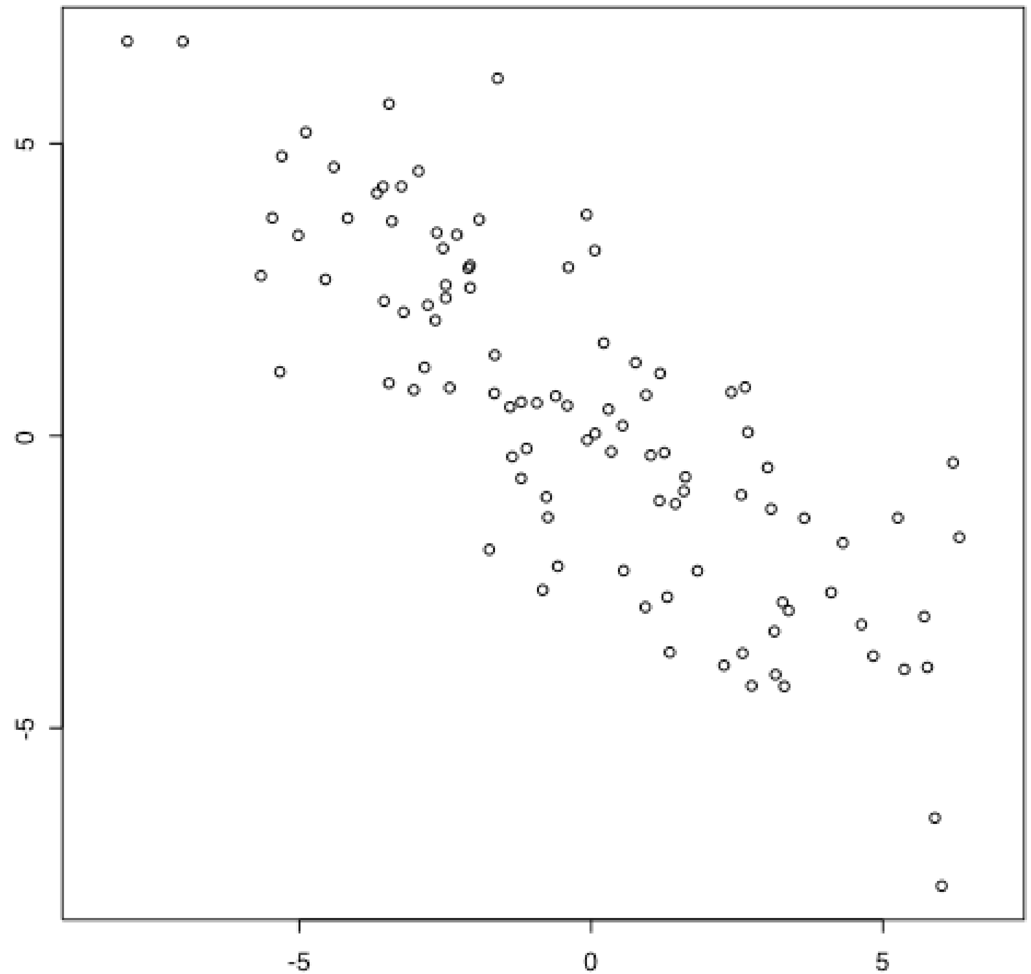
$$C_{xy} = \begin{bmatrix} 0.90 & 0.44 \\ 0.44 & 8.82 \end{bmatrix}$$



Dependent Variation

- c and d are random variables.
- Generated with $c=x+y \quad d=x-y$
- Covariance matrix:

$$C_{cd} = \begin{bmatrix} 10.62 & -7.93 \\ -7.93 & 8.84 \end{bmatrix}$$



Discrete Kalman Filter

- Estimate the state $\mathbf{x} \in \Re^n$ of a linear stochastic difference equation

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

  – process noise **w** is drawn from $N(0,\mathbf{Q})$, with covariance matrix **Q**.

- with a measurement $\mathbf{z} \in \Re^m$

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

  – measurement noise **v** is drawn from $N(0,\mathbf{R})$, with covariance matrix **R**.

- **A, Q** are $n \times n$ **B** is $n \times l$ **R** is $m \times m$ **H** is $m \times n$.

Estimates and Errors

- $\hat{\mathbf{x}}_k \in \Re^n$ is the estimated state at time-step k.
- $\hat{\mathbf{x}}_k^- \in \Re^n$ after prediction, before observation.
- Errors:

$$\mathbf{e}_k^- = \mathbf{x}_k - \hat{\mathbf{x}}_k^-$$

$$\mathbf{e}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k$$

- Error covariance matrices:

$$\mathbf{P}_k^- = E[\mathbf{e}_k^- \mathbf{e}_k^{-T}]$$

$$\mathbf{P}_k = E[\mathbf{e}_k \mathbf{e}_k^T]$$

- Kalman Filter's task is to update $\hat{\mathbf{x}}_k \; \mathbf{P}_k$

Time Update (Predictor)

- Update expected value of **x**

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

- Update error covariance matrix **P**

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

Measurement Update (Corrector)

- Update expected value

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

  – *innovation* is $\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-$

- Update error covariance matrix

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^-$$

The Kalman Gain

- The optimal Kalman gain $\mathbf{K}_k$ is

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1}$$

$$= \frac{\mathbf{P}_k^- \mathbf{H}^T}{\mathbf{H}\mathbf{P}_k^- \mathbf{H}^T + \mathbf{R}}$$

Kalman Filter Algorithm

1. Algorithm **Kalman_filter**(μ_{t-1}, P_{t-1}, u_t, z_t):

2. Prediction:
3. $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$
4. $\bar{P}_t = A_t P_{t-1} A_t^T + Q_t$

5. Correction:
6. $K_t = \bar{P}_t H_t^T (H_t \bar{P}_t H_t^T + R_t)^{-1}$
7. $\mu_t = \bar{\mu}_t + K_t (z_t - H_t \bar{\mu}_t)$
8. $P_t = (I - K_t H_t) \bar{P}_t$

9. Return μ_t, P_t

Extended Kalman Filter

- Suppose the state-evolution and measurement equations are non-linear:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_{k-1}$$

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k$$

  – process noise **w** is drawn from $N(0,\mathbf{Q})$, with covariance matrix **Q**.
  – measurement noise **v** is drawn from $N(0,\mathbf{R})$, with covariance matrix **R**.

The Jacobian Matrix

- For a scalar function $y=f(x)$,

$$\Delta y = f'(x) \qquad \Delta x$$

- For a vector function $\mathbf{y}=f(\mathbf{x})$,

$$\Delta \mathbf{y} = \mathbf{J} \qquad \Delta \mathbf{x} = \begin{bmatrix} \Delta y_1 \\ \vdots \\ \Delta y_n \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \cdots & \frac{\partial f_1}{\partial x_n}(\mathbf{x}) \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1}(\mathbf{x}) & \cdots & \frac{\partial f_n}{\partial x_n}(\mathbf{x}) \end{bmatrix} \cdot \begin{bmatrix} \Delta x_1 \\ \vdots \\ \Delta x_n \end{bmatrix}$$

Linearize the Non-Linear

- Let **A** be the Jacobian of *f* with respect to **x**.

$$\mathbf{A}_{ij} = \frac{\partial f_i}{\partial x_j}(\mathbf{x}_{k-1}, \mathbf{u}_k)$$

- Let **H** be the Jacobian of *h* with respect to **x**.

$$\mathbf{H}_{ij} = \frac{\partial h_i}{\partial x_j}(\mathbf{x}_k)$$

- Then the Kalman Filter equations are almost the same as before!

EKF Update Equations

- Predictor step: $$\hat{\mathbf{x}}_k^- = f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k)$$ $$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q}$$
- Kalman gain: $$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1}$$
- Corrector step: $$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - h(\hat{\mathbf{x}}_k^-))$$ $$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H})\mathbf{P}_k^-$$