Special Case : Kalman Filter

Linear Dynamical Systems

- Next state is a linear function of current state + zero-mean Gaussian noise
- Each observation is a linear function of current state + zero-mean Gaussian noise
- Initial prior distribution on first state is Gaussian

=> All distributions remain Gaussian!

=> Means and covariances can be estimated over time by a Kalman filter

Kalman Filter Derivation (in 1D)

Assume:

$\Rightarrow P(x_{k-1} \mid y_1 \cdots y_{k-1}) \sim N(\mu_{k-1}, \sigma^2_{k-1})$

Linear motion and measurement models:

motion model

$$x_k = \underbrace{a\,x_{k-1} + b}_{\substack{\text{linear (affine actually)} \\ \text{state transition}}} + \underbrace{c}_{\substack{\text{zero-mean} \\ \text{process noise}}} \qquad c \sim N(0, r^2)$$

measurement model

$$y_k = \underbrace{d\,x_k}_{\text{linear measurement}} + \underbrace{e}_{\substack{\text{zero-mean} \\ \text{observation noise}}} \qquad e \sim N(0, s^2)$$

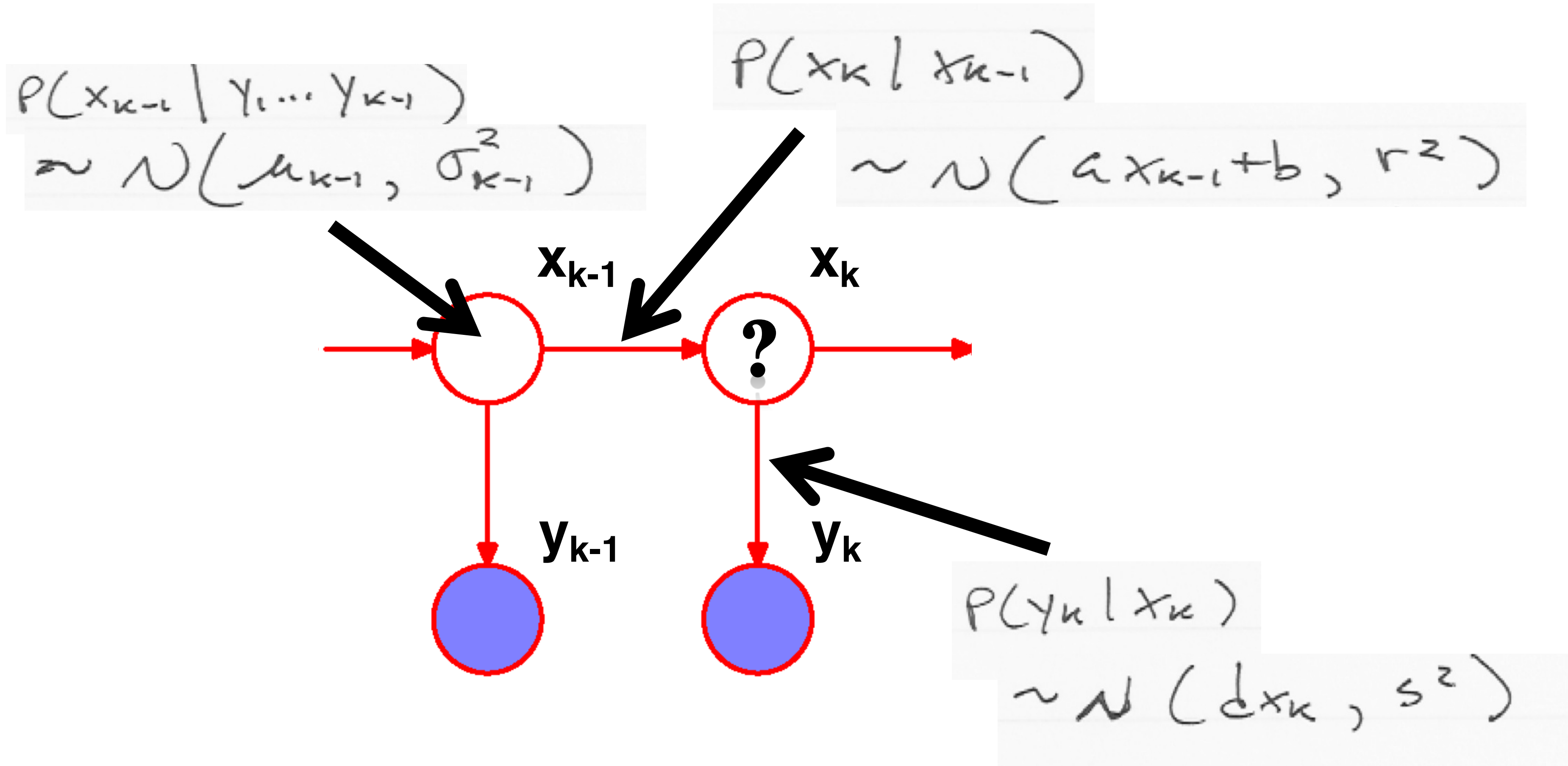
Implied by motion and measurement models:

$\Rightarrow P(x_k \mid x_{k-1}) \sim N(a x_{k-1} + b, r^2)$

$\Rightarrow P(y_k \mid x_k) \sim N(d x_k, s^2)$

Robert Collins
CSE586

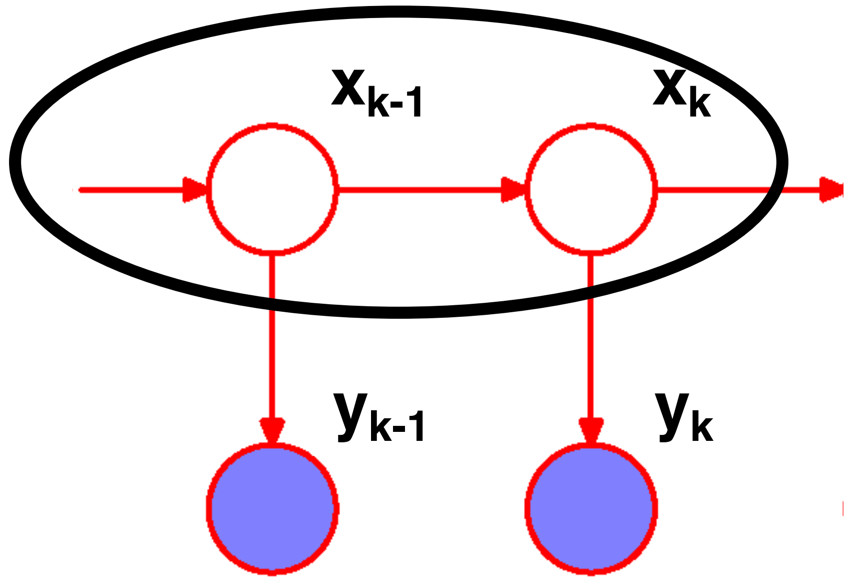
Kalman Filter Derivation



What is $P(x_k \mid y_1,\ldots,y_k)$?

Derivation Strategy

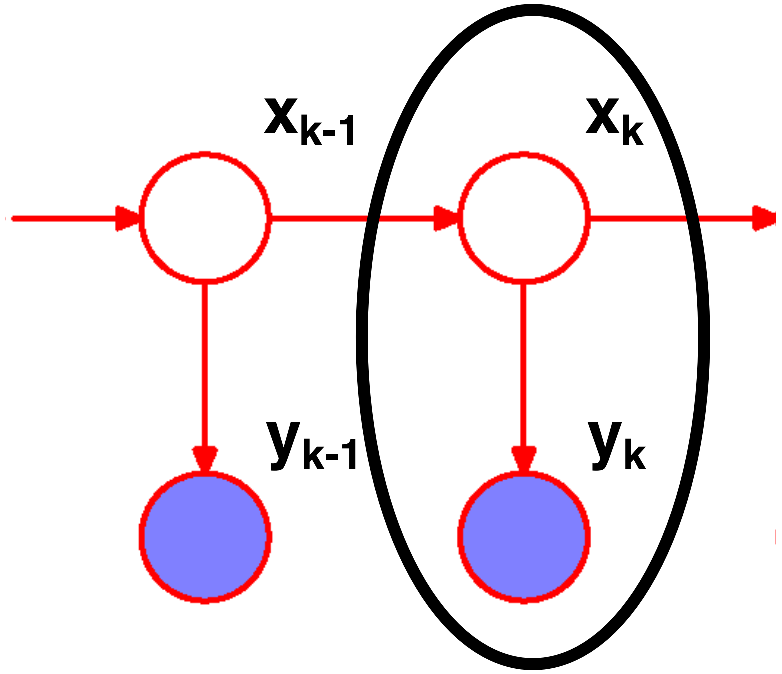
Combine $P(x_{k-1} \mid y_1,\ldots,y_{k-1})$ and $P(x_k|x_{k-1})$ to compute $P(x_k \mid y_1,\ldots,y_{k-1})$



Step 1: motion prediction

Derivation Strategy

Combine $P(x_k | y_1,\ldots,y_{k-1})$ and $P(y_k|x_k)$ to compute $P(x_k | y_1,\ldots,y_k)$



Step 2: data correction

Robert Collins
CSE586

Step 1: Motion Prediction

Combine $P(x_{k-1} \mid y_1,\ldots,y_{k-1})$ and $P(x_k \mid x_{k-1})$ to compute $P(x_k \mid y_1,\ldots,y_{k-1})$

$$P(x_k \mid y_1 \ldots y_{k-1}) = \int_{x_{k-1}} P(x_k \mid x_{k-1}) P(x_{k-1} \mid y_1 \ldots y_{k-1})\, dx_{k-1}$$

$$= \int_{x_{k-1}} N(x_k ; a x_{k-1} + b, r^2)\, N(x_{k-1} \mid \mu_{k-1}, \sigma_{k-1}^2)\, dx_{k-1}$$

$$= \text{constant} \int \exp\left\{ -\frac{1}{2} \underbrace{\left[\frac{(a x_{k-1} + b - x_k)^2}{r^2} + \frac{(x_{k-1} - \mu_{k-1})^2}{\sigma_{k-1}^2} \right]}_{\text{Quadratic form in } x_{k-1} \text{ and } x_k} \right\}$$

Quadratic form in x_{k-1} and x_k. Therefore this is a joint Gaussian distribution over x_{k-1} and x_k.

See Bishop PRML Chap 2

Integrated over x_{k-1} yields marginal distribution $P(x_k \mid y_1,\ldots,y_{k-1})$

$$P(x_k \mid y_1 \ldots y_{k-1}) \sim N(a\mu_{k-1} + b,\ a^2\sigma_{k-1}^2 + r^2)$$

Side note

We can derive $P(x_k | \cdot)$ in a simpler way.

$x_{k-1} \sim N(\mu_{k-1}, \sigma_{k-1}^2)$

Note that $x_k = a\, x_{k-1} + b + c$ <motion model $\cdot\, c \sim N(0, r^2)$

Let $z = a\, x_{k-1} + b$. An ~~linear~~ affine Transform of a Gaussian is also Gaussian

with $P(z) \sim N(a \mu_{k-1} + b,\ a^2 \sigma_{k-1}^2)$

Now, $x_k = z + c$

$z \sim N(a\mu_{k-1} + b,\ a^2 \sigma_{k-1}^2)$

$c \sim N(0, r^2)$

so X_K is the sum of two Gaussian random variables.

Appeal to some facts about sum of r.v.s, and convolutions of Gaussians...

Robert Collins
CSE586

Side note

specifically:

- distribution of the sum of two r.v.s is the convolution of their two individual distributions
- convolution of two Gaussians is another Gaussian, where

$$\text{mean}_{new} = \text{mean}_1 + \text{mean}_2$$

$$\text{variance}_{new} = \text{variance}_1 + \text{variance}_2$$

so

$$P(x_k \mid y_1 \ldots y_{k-1}) \sim N(a\mu_{k-1} + b,\ a^2\sigma_{k-1}^2 + r^2)$$

Robert Collins
CSE586

Step 2: Data Correction

Combine $P(x_k | y_1,\ldots,y_{k-1})$ and $P(y_k|x_k)$ to compute $P(x_k | y_1,\ldots,y_k)$

$$P(x_k | y_1 \ldots y_k) = \frac{P(y_k | x_k) P(x_k | y_1 \ldots y_{k-1})}{\int [\text{numerator}]}$$

This is just Bayes rule, applied to

$$P(x_k) \sim N(x_k | a\mu_{k-1} + b, \; a^2\sigma^2_{k-1} + r^2) \equiv N(\overbrace{\mu_k^-, \sigma_k^{2-}}^{\text{Define these symbols to make notation simpler}})$$

$$L(y_k|x_k) \sim N(y_k | d\, x_k, \; s^2)$$

Through some algebra (completing the square), we find that:

$$P(x_k | y_1 \ldots y_k) \sim N(\mu_k^+, \sigma_k^{2+})$$

where

$$\mu_k^+ = \frac{\sigma_k^{2-} \frac{y_k}{d} + \frac{s^2}{d^2} \mu_k^-}{\sigma_k^{2-} + \frac{s^2}{d^2}} = \frac{d\,\sigma_k^{2-} y_k + s^2 \mu_k^-}{d^2 \sigma_k^{2-} + s^2}$$

$$\sigma_k^{2+} = \frac{\sigma_k^{2-} \frac{s^2}{d^2}}{\sigma_k^{2-} + \frac{s^2}{d^2}} = \frac{\sigma_k^{2-} s^2}{d^2 \sigma_k^{2-} + s^2}$$

Robert Collins
CSE586

Step 2: Data Correction

Combine $P(x_k | y_1,\ldots,y_{k-1})$ and $P(y_k|x_k)$ to compute $P(x_k | y_1,\ldots,y_k)$

$$P(x_k | y_1 \ldots y_k) = \frac{P(y_k | x_k)\, P(x_k | y_1 \ldots y_{k-1})}{\int [\text{numerator}]}$$

This is just Bayes rule, applied to

$$P(x_k) \sim N(x_k | a\mu_{k-1} + b \,,\; a^2\sigma_{k-1}^2 + r^2)$$

$$L(y_k | x_k) \sim N(y_k | d\, x_k \,,\; s^2)$$

Through some algebra (completing the square), we find that:

$$P(x_k | y_1 \ldots y_k) \sim N(\mu_k^+ , \sigma_k^{2+})$$

where

$\mu_k^+ =$

$\sigma_k^{2+} =$

This becomes the Gaussian prior for the next iteration of motion prediction and data correction. Thus we continue propagating forward into subsequent time steps.

N-Dimensional Kalman Filter

Kalman filter on n-dimensional state vectors is more complicated looking because of matrix and vector notation, but otherwise it is derived in the same way as what we have just done.

Robert Collins
CSE586

Generalizing to N Dimensions

Let’s first rewrite our equations in a different form:

$$\mu_k^+ = \underbrace{\mu_k^-}_{\text{predicted state}} + \underbrace{\left[\frac{d\,\sigma_k^{2-}}{d^2\sigma_k^{2-} + s^2}\right]}_{\substack{\text{multiplicative factor} \\ \text{("gain" term)}}} \underbrace{(y_k - \underbrace{d\,\underbrace{\mu_k^-}_{\text{predicted state}}}_{\text{predicted measurement}})}_{\substack{\text{residual error between actual} \\ \text{and predicted measurements} \\ \text{"innovation"}}}$$

This makes it clearer ~~that~~ how the updated state is related to the predicted state and the difference between that predicted state and the observed measurement.

Robert Collins
CSE586

Generalizing to N Dimensions

Let's first rewrite our equations in a different form:

similarly

$$\sigma_k^{2+} = \left(1 - \left[\frac{d\sigma_k^{2-}}{d^2\sigma_k^{2-} + s^2}\right] d\right) \sigma_k^{2-}$$

shows how new variance is a multiplicative factor times the old variance.

Note the "gain" term in common to both equations, and write it using its own symbol

$$K_k = \frac{d\sigma_k^{2-}}{d^2\sigma_k^{2-} + s^2} \quad \Big\} \quad \underline{\text{Kalman gain}}$$

Robert Collins
CSE586

N-Dimensional Formulation

Linear Dynamical System:

$$\overset{n\times 1}{x_k} = \overset{n\times n}{A}\, \overset{n\times 1}{x_{k-1}} + \overset{n\times 1}{b} + \overset{n\times 1}{c} \qquad c \sim N(0, \overset{n\times n}{R})$$

$$\overset{m\times 1}{y_k} = \overset{m\times n}{D}\, \overset{n\times 1}{x_k} + \overset{m\times 1}{e} \qquad e \sim N(0, \overset{m\times m}{S})$$

The Kalman filter maintains an estimate of the posterior state distribution for x_k, $x_k \sim N(\underset{n\times 1}{\mu_k}, \underset{n\times n}{\Sigma_k})$

Robert Collins
CSE586

N-Dimensional Kalman Filter

KF state prediction (motion update)

$$\mu_k^- = A\mu_{k-1} + b$$

$$\Sigma_k^- = A\,\Sigma_{k-1}A^T + R$$

KF correction (measurement update)

$$K_k = \Sigma_k^- D^T (D\,\Sigma_k^- D^T + S)^{-1}$$

$$\mu_k = \mu_k^- + K_k(y_k - D\mu_k^-)$$

$$\Sigma_k = (I - K_k D)\,\Sigma_k^-$$

Example: Constant Velocity

2D "constant velocity" model

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ u \\ v \end{bmatrix}$$

where
(x,y) = current location
(u,v) = current velocity

Constant velocity motion model would be:

$$x_k = x_{k-1} + u_{k-1}$$
$$y_k = y_{k-1} + v_{k-1}$$
$$\left.\begin{aligned} u_k &= u_{k-1} \\ v_k &= v_{k-1} \end{aligned}\right\} \text{velocity stays constant}$$

Robert Collins
CSE586

Example: Constant Velocity

2D "constant velocity" model

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ u \\ v \end{bmatrix}$$

$$\underset{n\times 1}{x_k} = \underset{n\times n}{A}\,\underset{n\times 1}{x_{k-1}} + \underset{n\times 1}{b} + \underset{n\times 1}{c} \qquad c \sim N(0, \underset{n\times n}{R})$$

$$\underset{m\times 1}{y_k} = \underset{m\times n}{D}\,\underset{n\times 1}{x_k} + \underset{m\times 1}{e} \qquad e \sim N(0, \underset{m\times m}{S})$$

Constant velocity motion model would be:

$$x_k = x_{k-1} + u_{k-1}$$
$$y_k = y_{k-1} + v_{k-1}$$
$$u_k = u_{k-1}$$
$$v_k = v_{k-1}$$

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

Robert Collins
CSE586

Example: Constant Velocity

2D "constant velocity" model

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ u \\ v \end{bmatrix}$$

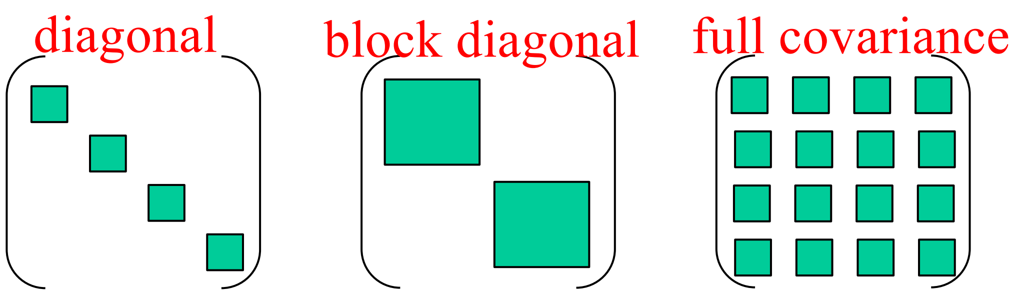
$$\overset{n\times 1}{x_k} = \overset{n\times n}{A}\,\overset{n\times 1}{x_{k-1}} + \overset{n\times 1}{b} + \overset{n\times 1}{c} \qquad c \sim N(0, \overset{n\times n}{R})$$

$$\overset{m\times 1}{y_k} = \overset{m\times n}{D}\,\overset{n\times 1}{x_k} + \overset{m\times 1}{e} \qquad e \sim N(0, \overset{m\times m}{S})$$

Constant velocity motion model would be:

$$\begin{aligned} x_k &= x_{k-1} + u_{k-1} \\ y_k &= y_{k-1} + v_{k-1} \\ u_k &= u_{k-1} \\ v_k &= v_{k-1} \end{aligned} \qquad A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad b = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

several choices of noise model
specified by covariance matrix R



Robert Collins
CSE586

Example: Constant Velocity

2D "constant velocity" model

$$\mathbf{X} = \begin{bmatrix} x \\ y \\ u \\ v \end{bmatrix}$$

$$\underset{n\times 1}{x_k} = \underset{n\times n}{A}\,\underset{n\times 1}{x_{k-1}} + \underset{n\times 1}{b} + \underset{n\times 1}{c} \qquad c \sim N(0, \underset{n\times n}{R})$$

$$\underset{m\times 1}{y_k} = \underset{m\times n}{D}\,\underset{n\times 1}{x_k} + \underset{m\times 1}{e} \qquad e \sim N(0, \underset{m\times m}{S})$$

Observation model

x_k, y_k: we observe a noisy version of this

$\cancel{u_k}$, $\cancel{v_k}$: we cannot observe this directly

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

Observation noise specified by 2x2 covariance matrix S

Robert Collins
CSE586

Using KF for Prediction

2D "constant velocity" model

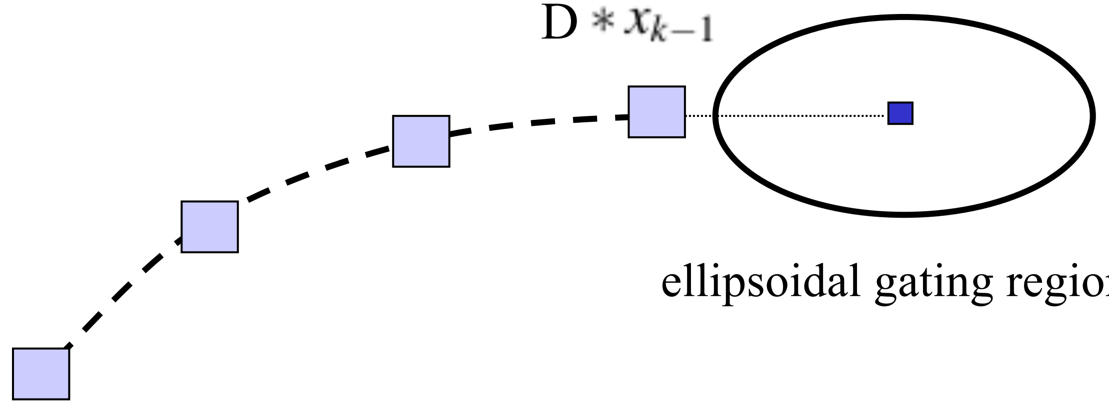
$$x = \begin{bmatrix} x \\ y \\ u \\ v \end{bmatrix} \quad A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\underset{n\times 1}{x_k} = \underset{n\times n}{A}\,\underset{n\times 1}{x_{k-1}} + \underset{n\times 1}{b} + \underset{n\times 1}{c} \qquad c \sim N(0, \underset{n\times n}{R})$$

$$\underset{m\times 1}{y_k} = \underset{m\times n}{D}\,\underset{n\times 1}{x_k} + \underset{m\times 1}{e} \qquad e \sim N(0, \underset{m\times m}{S})$$

$D * A * x_{k-1}$

$D * x_{k-1}$



$\tilde{y}' S^{-1} \tilde{y} \leq$ threshold

ellipsoidal gating region

$$\hat{\mathbf{x}}_{k|k-1} = A_k \hat{\mathbf{x}}_{k-1|k-1} \quad \text{(predicted state)}$$

$$\mathbf{P}_{k|k-1} = A_k \mathbf{P}_{k-1|k-1} A_k^T + R_k \quad \text{(predicted estimate covariance)}$$

$$\tilde{y} = y_k - D_k \hat{\mathbf{x}}_{k|k-1} \quad \text{(innovation or measurement residual)}$$

$$\mathbf{S}_k = D_k \mathbf{P}_{k|k-1} D_k^T + S_k \quad \text{(innovation (or residual) covariance)}$$