ASEN 5044, Fall 2024
Statistical Estimation for Dynamical Systems

Lecture 33:
The Unscented Kalman Filter (UKF)
[aka the Sigma Point Filter (SPF)]

Prof. Nisar Ahmed (Nisar.Ahmed@Colorado.edu)

Thursday 12/05/2024

Ann and H.J. Smead
Aerospace Engineering Sciences
UNIVERSITY OF COLORADO **BOULDER**

Announcements

- **Final project assignment tasks + Progress report 1 due Fri 12/06**
- **Final Project Progress Report 2 posted due Fri 12/13**
- **Final Project Report Due Tues 12/17, 11:59 pm**
- **Quiz 8 out – due Tues 12/10**
- **Today is the final lecture for semester!**
    - **No live class/lectures/recordings next week**
    - **Thurs 12/12 lecture time = office hours (+ other regular office hours next week)**
- **FCQs now out online**

Some Guidance on Final Project

Hints for Part 1 (focus of Progress Report 1)

- Please title and label axes of your plots properly! (esp. perturbation vs. total states)
- **DO NOT** JUST HAND IN CODE AND PLOTS WITHOUT EXPLANATION! → **ZERO CREDIT** (explain what you did, apart from your code -- code is NOT part of page count)
- Use **physically reasonable** (not too large) perturbation values
  - (i.e. s/c in StatOD should not be skimming bushes; UAV-UGV should not spin off into oblivion…)

→ PAY ATTENTION TO THE NUMBERS AND INTERPRET YOUR RESULTS! (look at posted solution sketches carefully + debug your code in stages)

Hint for Part 2:

- Note on using Q matrices provided for ground truth data generation: interpret Q matrix values given in data logs as CT PSDs for process noise *injected at discrete time steps*
  - i.e. sample noise \tilde{w}($t=t_k$)=w_k ~N(0,Q), and *treat* w_k *as if it were a zero-order hold (ZOH) external input to your CT dynamics for R-K integration* (going from $t=t_k$ to $t=t_{k+1} = t_k + \Delta t$)
  - NL CT dyn fxn given to ode45/R-K routine SHOULD ***NOT*** HAVE RANDOM NUMBER GENERATOR INSIDE IT!!! (o'wise screws up adaptive multi-step integration → nonsense...)

Generally: if using any code assistants and/or collaborating with others:

please note/cite what + whom you collaborated with!!

Overview

Last Time:

- Quick overview of Final Project assignment
- DT Extended Kalman filter (EKF)
- Closer look at linearization and covariance approximations for LKF & EKF
- General considerations for Initializing + Tuning the LKF & EKF

Today:

- Limitations of the EKF and Taylor series-based linearization
- Alternative to linearization: sigma point (unscented) transformation
- Sigma point KF (SPF), aka the Unscented KF (UKF)

Example: Static Scalar Estimator with Nonlinear Data

- Suppose $x \sim \mathcal{N}(\hat{x}^- = \mu_x, P^- = \sigma_x^2)$, for $x_{k+1} = x_k = x$ (static scalar state)

$y = h(x) = x^2$ (scalar noiseless nonlinear measurement)

$\rightarrow$ find $\mu_y = E[y]$ and $\sigma_y^2 = \text{cov}(y)$?

Exact calculation:

$\rightarrow$ let $x = \mu_x + \delta x$, or $\delta x = x - \mu_x$,

then $y = (\mu_x + \delta x)^2 = \mu_x^2 + 2\mu_x \delta x + \delta x^2$,

so $\mu_y = E[y] = \mu_x^2 + 2\mu_x E[\delta x] + E[\delta x^2]$

$= \mu_x^2 + 2\mu_x E[x - \mu_x] + E[(x - \mu_x)^2]$

$\rightarrow \mu_y = \mu_x^2 + \sigma_x^2$ (exact answer)

likewise: $\sigma_y^2 = E[(y - \mu_y)^2] = \ldots$

$\rightarrow \sigma_y^2 = 2\sigma_x^4 + [4\mu_x^2 \sigma_x^2]$ (exact answer)

Linearization results (EKF):

$\rightarrow \mu_{y,EKF} \approx h(\mu_x) = \mu_x^2$

$\sigma_{y,EKF}^2 \approx \tilde{H} \sigma_x^2 \tilde{H}^T$,

P_x

where $\tilde{H}|_{\mu_x} = \frac{\partial h}{\partial x}|_{\mu_x} = 2\mu_x$,

so $\sigma_{y,EKF}^2 = 4\mu_x^2 \sigma_x^2$

$\rightarrow \mu_{y,EKF} = \mu_x^2$

and $\sigma_{y,EKF}^2 = 4\mu_x^2 \sigma_x^2$

EKF mean and variance are clearly incorrect! (mean is biased; variance too small: missing 4th order kurtosis term...)

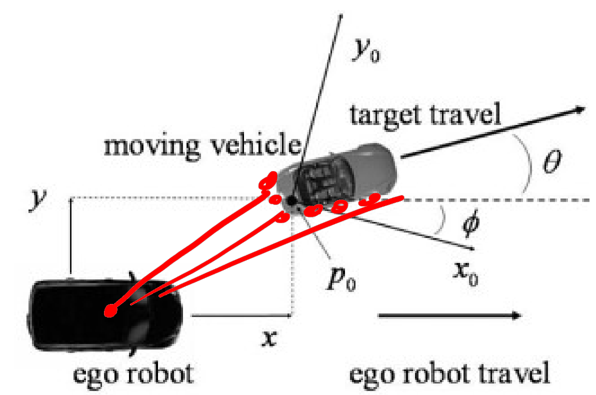
Limitations of the EKF and Taylor Series Linearization

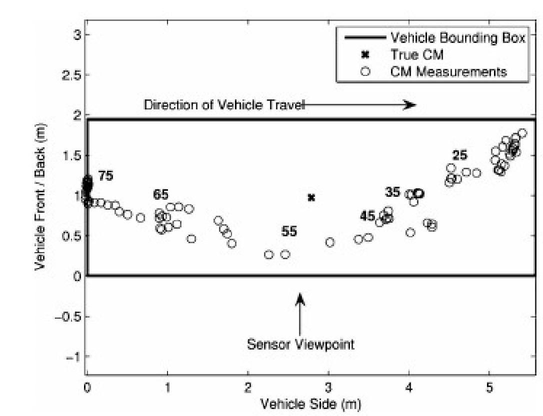
- Truncation of higher order terms in Taylor expansions
  - ○ can incur significant biases that may not easily covered up by process noise tuning

    → in such cases, would need state augmentation (i.e. biases added to state vector) or higher order filtering (e.g. second-order EKF with more Jacobians) or iterated estimates (e.g. IEKF with multiple linearization passes)
  - ○ → incur more computational expense!! And yet, still no guarantees!!!

- Mean and covariance approximations based on Taylor series approximation of dynamics only valid if state estimate close to true state mean
  - ○ no guarantees

    → can under-/over-approximate estimation error covars!!!



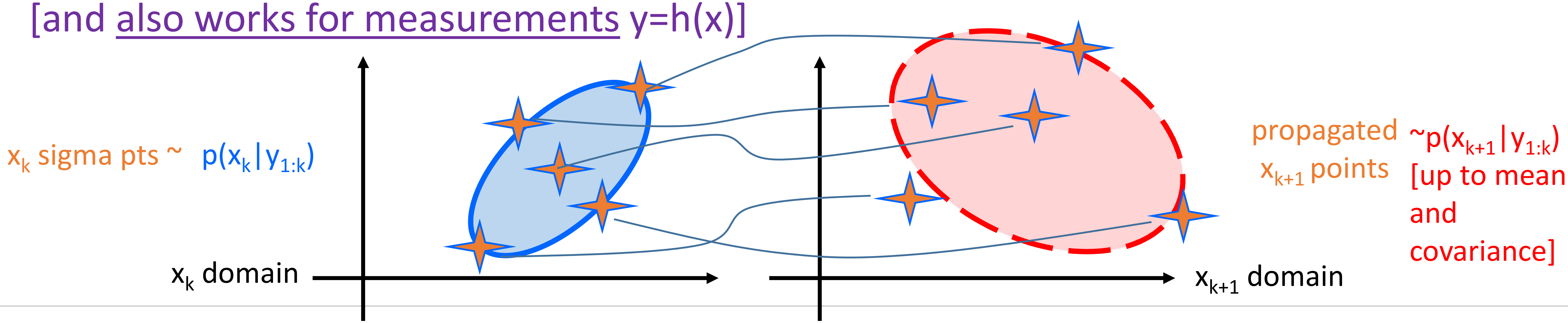


- Need to compute Jacobians
  - ○ Cannot deal with non-smooth dynamics or measurements
    - e.g. lidar data features for tracking moving cars (sudden jumps in min/max bearing angles off car sides):

    I. Miller, M. Campbell, and D. Huttenlocher. "Efficient unbiased tracking of multiple dynamic obstacles under large viewpoint changes." IEEE Transactions on Robotics 27.1 (2011): 29-46.

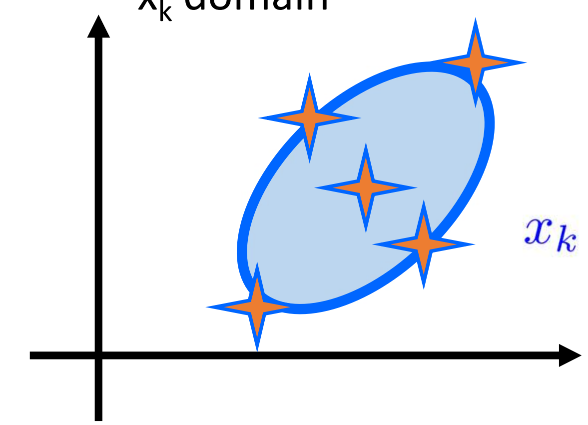
Alternative Approximation: Unscented Transform

- <u>Key idea:</u> easier to directly approximate mean and covariance from one pdf to another, than to analytically approximate the nonlinear state transition function first…
- Easy way to approximate transformed pdf mean/covariance: **<u>simulation-based sampling</u>**
- Use special **"sigma points"** from initial state pdf $p(x_k | y_{1:k})$ for state x_k at time k
  - By definition: weighted sigma pts sample mean/covar = true mean/covar of $p(x_k | y_{1:k})$
- Propagate sigma pts thru, e.g., non-linear dynamics function f(.) to <u>simulate</u> x_{k+1} values
- Use *weighted sample mean/covar of* x_{k+1} *pts* to *approx* mean/covar of $p(x_{k+1} | y_{1:k})$
- <u>No Jacobians:</u> samples retain additional higher order f(.) terms vs. Taylor linearization, [and <u>also works for measurements</u> y=h(x)]



The Unscented (Sigma Point) Transform

- How to actually "sample" sigma points?
- Deterministic selection of x_k from pdf with $E[x_k] = \mu_x$ and $\text{cov}(x_k) = P_{xx}$
  - Minimal set of points in x_k that "sketch out" first two moments of pdf



$\rightarrow$ To generate these points for a given μ_x and P_{xx},

use $P_{xx} = S^T S$, where $S = \text{chol}(P_{xx})$

$\rightarrow$ $2n+1$ sigma points: $\chi^0 = \mu_x$ (at the mean)

$$\chi^i = \begin{cases} \mu_x + (\sqrt{n+\lambda}) \cdot S^{j,T}, \text{ for } i = 1, ..., n, \text{ and } j = 1, ..., n \\ \mu_x - (\sqrt{n+\lambda}) \cdot S^{j,T}, \text{ for } i = n+1, ..., 2n, \text{ and } j = 1, ..., n \end{cases}$$

where: S^j: jth row of S

n: number of states (dim of x_k)

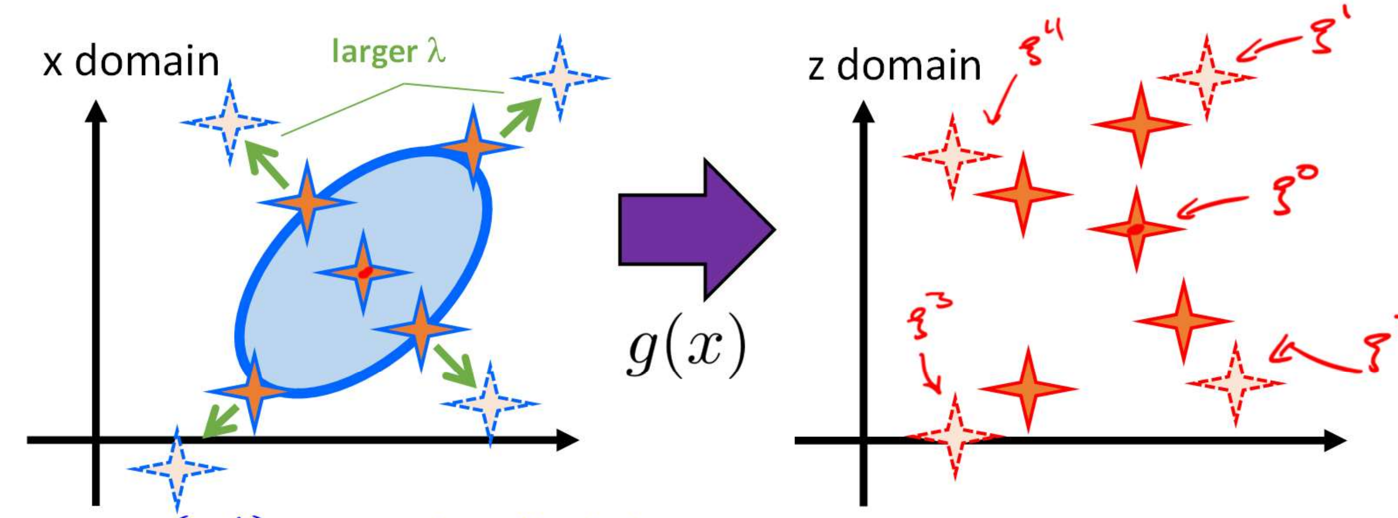
λ: scaling parameter $= \alpha^2 \cdot (n+\kappa) - n$ (typical values: $\kappa = 0$, $\alpha \in [10^{-4}, 1]$)

How to Use Sigma Points (generally)?

- Suppose we want to find $\mu_z = E[z]$ and $P_{zz} = \text{cov}(z)$ for some nonlinear vector function $z = g(x)$, given $x \text{``}\sim\text{''} p(x)$ s.t. $E[x] = \mu_x$, $\text{cov}(x) = P_{xx}$.

$\rightarrow$ Simply propagate sigma pts $\{\chi^i\}_{i=0:2n}$ through $g(x)$, and get 'resultant sample' points $\{\xi^i\}_{i=0:2n}$:

$$\xi^0 = g(\chi^0),$$
$$\xi^i = g(\chi^i),\ i = 1, \cdots, 2n$$



$\{\chi^i\}_{i=0:2n}$ "$\sim$" $p(x)$

$\{\xi^i\}_{i=0:2n}$ "$\sim$" $p(z = g(x))$

ie hopefully up to $E[z]$ & P_{zz}

$\rightarrow$ Recombine resultant points to estimate μ_z and P_{zz}:

$$\mu_z = \sum_{i=0}^{2n} w_m^i \cdot \xi^i,$$

$$P_{zz} = \sum_{i=0}^{2n} w_c^i \cdot (\xi^i - \mu_z)(\xi^i - \mu_z)^T$$

where:

$$\left.\begin{array}{ll} w_m^0 = \frac{\lambda}{n+\lambda}, & w_m^i = \frac{1}{2(n+\lambda)}, \\ w_c^0 = \frac{\lambda}{n+\lambda} + 1 - \alpha^2 + \beta, & w_c^i = w_m^i, \end{array}\right\} \text{for } i = 1, \cdots, 2n$$

$$\lambda = \alpha^2 \cdot (n + \kappa) - n$$

Typically:

*set $\kappa = 0$, $\beta = 2$, and play with α
(large α: sigma pts more spread out)
(small α: approaches EKF)

*but no matter what: will always get right answers if $g(x)$ linear!

Example: Scalar Estimator with Nonlinear Data (Revisited)

- Suppose $x \sim \mathcal{N}(\hat{x}^- = \mu_x, P^- = \sigma_x^2)$, for $x_{k+1} = x_k = x$ (static scalar state)

$$y = h(x) = x^2 \text{ (scalar noiseless nonlinear measurement)}$$

$$\rightarrow \text{find } \mu_y = E[y] \text{ and } \sigma_y^2 = \text{cov}(y)?$$

Unscented transform results (using sigma pts for n=1):

$$\chi^0 = \mu_x, \qquad \chi^i = \mu_x \pm (\sqrt{1+\lambda}) \cdot \sigma_x,$$

$$\rightarrow \xi^0 = h(\chi^0) = \mu_x^2, \qquad \xi^i = h(\chi^i) = (\mu_x \pm \sqrt{1+\lambda} \cdot \sigma_x)^2$$

$$\rightarrow \mu_y = \sum_{i=0}^{2n} w_m^i \cdot \xi^i = \frac{\lambda}{1+\lambda}\mu_x^2 + \frac{1}{2+2\lambda}(\xi^1) + \frac{1}{2+2\lambda}(\xi^2) = \boxed{\mu_x^2 + \sigma_x^2}$$

$$\sigma_y^2 = \sum_{i=0}^{2n} w_c^i \cdot (\xi^i - \mu_y)^2 = \frac{\lambda}{1+\lambda}(\xi^0 - \mu_y)^2 + \frac{1}{2+2\lambda}(\xi^1 - \mu_y)^2 + \frac{1}{2+2\lambda}(\xi^2 - \mu_y)^2$$

$$= [\alpha^2 \cdot \kappa + \beta] \cdot \sigma_x^4 + 4\mu_x^2\sigma_x^2 = \boxed{2\sigma_x^4 + 4\mu_x^2\sigma_x^2 \text{ (for typical values)}}$$

Sigma point / unscented transformation recovers the desired exact values!
(mean is unbiased, variance properly accounts for higher order moment terms)

The Unscented Kalman Filter (aka Sigma Point Filter)

- The nonlinear filtering problem consists of 2 nonlinear transformations, whose statistics will each be approximated by sigma pts for KF-like prediction and measurement updates:

$$x_{k+1} = f(x_k, u_k) + w_k, \;\; w_k \sim \mathcal{N}(0, Q_k) \qquad\qquad y_{k+1} = h(x_{k+1}) + v_{k+1}, \;\; v_{k+1} \sim \mathcal{N}(0, R_{k+1})$$

1. Dynamics Prediction Step from time step k → k+1:

(a) Given $\hat{x}_k^+, P_k^+$ from time step k, set $S_k = \text{chol}(P_k^+)$, and generate $2n+1$ sigma pts:

$$\chi_k^0 = \hat{x}_k^+$$

$$\chi_k^i = \begin{cases} \hat{x}_k^+ + (\sqrt{n+\lambda}) \cdot S_k^{j,T}, \text{ for } i = 1, ..., n, \text{ and } j = 1, ..., n \\ \hat{x}_k^+ - (\sqrt{n+\lambda}) \cdot S_k^{j,T}, \text{ for } i = n+1, ..., 2n, \text{ and } j = 1, ..., n \end{cases}$$

S_k^j: jth row of S_k

n: number of states (dim of x_k)

$\lambda = \alpha^2 \cdot (n + \kappa) - n$

(b) Propagate each χ_k^i through nonlinear dynamics $f(\cdot)$ to get resultant pts $\bar{\chi}_{k+1}^0$ and $\bar{\chi}_{k+1}^i$

(i.e. using full R-K integrator on *each* χ_k^i for $i = 0, 1..., 2n$)

(c) Recombine resultant pts to get predicted mean and covariance: (using w_m^i and w_c^i as defined on prev slides)

$$\boxed{\hat{x}_{k+1}^- \approx \sum_{i=0}^{2n} w_m^i \cdot \bar{\chi}_{k+1}^i, \qquad P_{k+1}^- \approx \sum_{i=0}^{2n} w_c^i \cdot (\bar{\chi}_{k+1}^i - \hat{x}_{k+1}^-)(\bar{\chi}_{k+1}^i - \hat{x}_{k+1}^-)^T + Q_k}$$

The Unscented Kalman Filter (aka Sigma Point Filter)

- The nonlinear filtering problem consists of 2 nonlinear transformations, whose statistics will each be approximated by sigma pts for KF-like prediction and measurement updates:

$$x_{k+1} = f(x_k, u_k) + w_k, \;\; w_k \sim \mathcal{N}(0, Q_k) \qquad\qquad y_{k+1} = h(x_{k+1}) + v_{k+1}, \;\; v_{k+1} \sim \mathcal{N}(0, R_{k+1})$$

2. Measurement Update Step at time k+1 given observation y(k+1):

(a) Given $\hat{x}^-_{k+1}, P^-_{k+1}$ from Prediction Step, set $\bar{S}_{k+1} = \text{chol}(P^-_{k+1})$, & generate $2n+1$ (new) sigma pts:

$$\chi^0_{k+1} = \hat{x}^-_{k+1}$$

$$\chi^i_{k+1} = \begin{cases} \hat{x}^-_{k+1} + (\sqrt{n+\lambda}) \cdot \bar{S}^{j,T}_{k+1}, & \text{for } i = 1, ..., n, \text{ and } j = 1, ..., n \\ \hat{x}^-_{k+1} - (\sqrt{n+\lambda}) \cdot \bar{S}^{j,T}_{k+1}, & \text{for } i = n+1, ..., 2n, \text{ and } j = 1, ..., n \end{cases}$$

$\bar{S}^j_{k+1}$: jth row of $\bar{S}_{k+1}$

n: number of states (dim of x_k)

$\lambda = \alpha^2 \cdot (n + \kappa) - n$

(b) Propagate each χ^i_{k+1} through nonlinear measurement fxn $h(\cdot)$ to get resultant pts γ^0_{k+1} and γ^i_{k+1}

(i.e. such that $\gamma^i_{k+1} = h(\chi^i_{k+1})$ for $i = 0, 1..., 2n$)

(c) Get predicted measurement mean and measurement covariance: (w^i_m and w^i_c defined on slide 12)

$$\hat{y}^-_{k+1} \approx \sum_{i=0}^{2n} w^i_m \cdot \gamma^i_{k+1}, \qquad\qquad P_{yy,k+1} \approx \sum_{i=0}^{2n} w^i_c \cdot (\gamma^i_{k+1} - \hat{y}^-_{k+1})(\gamma^i_{k+1} - \hat{y}^-_{k+1})^T + R_{k+1}$$

The Unscented Kalman Filter (aka Sigma Point Filter)

2. Measurement Update Step at time k+1 given observation y(k+1):

(d) Get state-measurement cross-covariance matrix $(n \times p)$: (w_m^i & w_c^i defined on slide 12; χ_{k+1}^i generated at this step)

$$P_{xy,k+1} \approx \sum_{i=0}^{2n} w_c^i \cdot (\chi_{k+1}^i - \hat{x}_{k+1}^-)(\gamma_{k+1}^i - \hat{y}_{k+1}^-)^T$$

(e) Estimate Kalman gain matrix $(n \times p)$:

$$K_{k+1} \approx P_{xy,k+1} \cdot [P_{yy,k+1}]^{-1}$$

(f) Perform Kalman state and covariance update with observation y_{k+1}:

$$\boxed{\begin{aligned} \hat{x}_{k+1}^+ &= \hat{x}_{k+1}^- + K_{k+1}(y_{k+1} - \hat{y}_{k+1}^-) \\ P_{k+1}^+ &= P_{k+1}^- - K_{k+1} P_{yy,k+1} K_{k+1}^T \\ &= P_{k+1}^- - P_{xy,k+1}[P_{yy,k+1}]^{-1} P_{xy,k+1}^T \end{aligned}}$$

Where do these come from?
→ See Advanced Topic Lecture #28!

Generalizations and Caveats for the UKF (SPF)

- Preceding filtering algorithm is for additive process and measurement noise only
- Can **generalize to non-additive (e.g. multiplicative) noises** by augmenting sigma points with **process and measurement noise sigma pts** – see Simon p. 450-451 & refs therein

$$x_{k+1} = f(x_k, u_k, w_k), \quad y_{k+1} = h(x_{k+1}, v_{k+1})$$

- Computational issues with the UKF
  - can implement square root version of UKF to guarantee posdef covariances
  - dealing with matrix inverse – can also implement information filter version
  - more computation to propagate/sample sigma pts – can sometimes cheat a little…
  - tuning spread of sigma pts via λ/α in addition to Q_{UKF} – use truth model testing…

- UKF only captures first two moments of transformed pdfs for recursive filtering – breaks down when higher order pdf moments needed!
  - e.g. highly asymmetric / multi-modal pdfs