ASEN 5044, Fall 2024
Statistical Estimation for Dynamical Systems

Lecture 30:
KF Tuning;
Intro to Nonlinear Filtering

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

Thursday 11/14/2024

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

Announcements

- Midterm 2 due today at 11:59 pm (Gradescope)

- **Final project partners signup: see instructions + project descriptions + signup link on Canvas – please complete by Mon 11/18**
  - **Part 1 of final project assignment to be released this Mon 11/18!!**

- **HW 8: out today, due Fri 11/22 (last HW!)**

- **Quiz 8 out ~~11/28~~ 12/05 (after return from Fall Break - last Quiz!)**

Overview

Last Time

- How to tell if your (linear) KF is actually working correctly???
- **KF dynamic consistency analysis** and **“Truth Model Testing” (TMT)**
- **Chi-square hypothesis tests (NEES/NIS)** – check if KF’s state errors/measurement residuals make sense for given system + measurement + noise models
  - **Do actual state errors/meas. residuals agree with KF’s estimated error covariances?**
  - **Formal statistical tests to examine this question**

Today (+ Next Time):

- Tie up Linear KF loose ends: **NIS test (derivations/details) + KF tuning procedure**
- Intro to **nonlinear dynamical state estimation** (discrete time)
  - **Optimal non-linear state estimation problem definition and setup**

- **Two popular sub-optimal “analytic” approximations**
  - **Linearized KF**
  - **Extended KF (EKF)**

READ SIMON BOOK, CHAPTERS 13.1-13.2 (nonlinear KFs)

2^{nd} Statistical Test for KF Performance: NIS

- Similar test exists for looking at multiple runs of simulated or real data with NIS – useful if truth model is unavailable (i.e. only real data at hand)

NIS: $\epsilon_{y,k} = e_{y,k}^T (S_k)^{-1} e_{y,k}$ [scalar], where $e_{y,k} = y_k - \hat{y}_k = y_k - H\hat{x}_k^- \in \mathbb{R}^{p\times 1}$ [innovation vector], $S_k = H_k P_k^- H_k^T + R \in \mathbb{R}^{p\times p}$ [is $cov(e_{y,k})$] [innovation error covariance]

Recall: $\epsilon_{y,k} \sim \chi^2_p$ $\forall$ time steps k [if p constant] if KF works properly

→ can perform N independent actual/Monte Carlo sim experimental runs [i.e. get actual/simulated sensor data from "full" dyn. system on N field/Monte Carlo runs]

& then calculate: $\bar{\epsilon}_y(k) = \frac{1}{N}\sum_{i=1}^{N} \epsilon_y^i(k)$ ← i indexes expt. run [sample average of NIS @ time k across N experiments]

→ Analysis & χ^2 test (similar to NEES): $N\cdot\bar{\epsilon}_y(k) \sim \chi^2_{N\cdot p}$, so again we have that $\bar{\epsilon}_y(k)$ should be in interval $r_1 \le \bar{\epsilon}_y(k) \le r_2$ $(1-\alpha)\cdot 100\%$ of the time

[except: here we are testing zero meanness & size of $e_{y,k}$, not $e_{x,k}$!]

→ In Matlab: $r_1 = \text{chi2inv}(\frac{\alpha}{2}, N\cdot p)./N$

$r_2 = \text{chi2inv}(1-\frac{\alpha}{2}, N\cdot p)./N$

Recall: α = type I / "False Alarm" rate (usually 0.05 or 0.01)

N = # Monte Carlo runs or Expt. runs

p = size of y_k

Kalman Filter Tuning Procedure

- What to do if your KF fails NEES/NIS chi-square consistency tests?
- Assume you've eliminated possibility of coding errors → then inconsistency must be explained by some error in formulation of stochastic state space model.

Strategy: Assume F, G, H, R, u (input vector) all correct (i.e. double-check these first).

Then we focus on tuning elements of Q_{KF} to pass consistency test.

(since we often know the least about Q in DT LTI model)

- If $\bar{\epsilon}_{y,k}$ 'too small' $\rightarrow$ Q_{KF} probably 'too big':

recall: $\text{NIS} = e_{y,k}^T (S_k)^{-1} e_{y,k}$, where $S_k = HP_k^- H^T + R$

$\rightarrow$ if $e_{y,k}^T (S_k)^{-1} e_{y,k}$ too small $\rightarrow$ S_k too large $\rightarrow$ P_k^- **too large**

$\rightarrow$ estimate fits data 'too well': KF not trusting dynamics model enough

$\rightarrow$ KF adjusting $\hat{x}_k^-$ too much in response to meas. innovation $e_{y,k}$

also recall: $P_k^- = FP_{k-1}^+ F^T + Q_{KF}$ $\rightarrow$ if Q_{KF} too large, P_k^- will be too large

Tuning the KF: How to Select Q_{KF} (KF guess of Q)

- So, if $\bar{\epsilon}_{y,k}$ too small (i.e. KF too conservative/pessimistic), try decreasing Q_{KF}

  KF gain too large/saturated

  e.g. $Q_{KF}^{new} = (1/10)\, Q_{KF}$ [order of mag. adjust & see what happens]
  → need to play w/ to see improvement...

- What if $\bar{\epsilon}_{y,k}$ too big?

  $\rightarrow$ by similar logic, **KF is too optimistic/overconfident**

  $\rightarrow$ Q_{KF} **too small** (not enough prediction/model uncertainty)

  $\rightarrow$ KF does not update $\hat{x}_k^-$ enough in response to $e_{y,k}$

  $\rightarrow$ KF constantly 'too surprised' by new data y_k, but KF gain K_k too small

  $\Rightarrow$ In this case, try increasing Q_{KF} → e.g. $Q_{KF}^{new} = 10 \cdot Q_{KF}$ [order of mag adjust & see what happens...]

- **In either case:** after adjusting Q_{KF} by overall factor, go in and tweak individual elements of Q_{KF} based on knowledge of system (e.g. which states you think are noisiest, most poorly modeled, etc.)

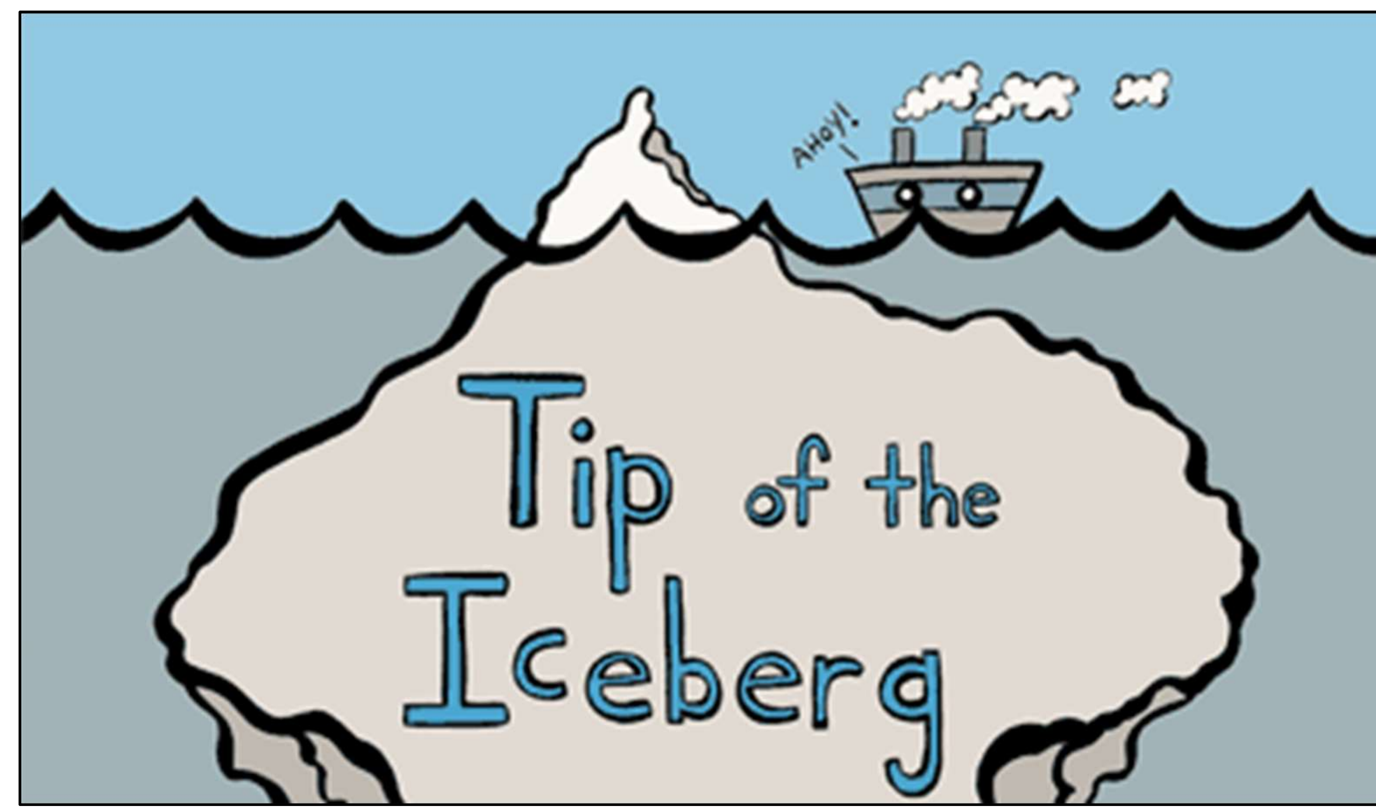
Introduction to Nonlinear Estimation

Roadmap for remaining few lectures:

- To what extent **do linear estimation methods apply to non-linear problems**?
- Basic but widely used methods based on **linearization**:
  - Linearized KF
  - Extended Kalman Filter (EKF)
  - Unscented Kalman Filter (UKF)
- Survey more powerful/general methods:
  - Nonlinear least squares (NLS)
  - Maximum likelihood
  - Bayesian filters and estimators

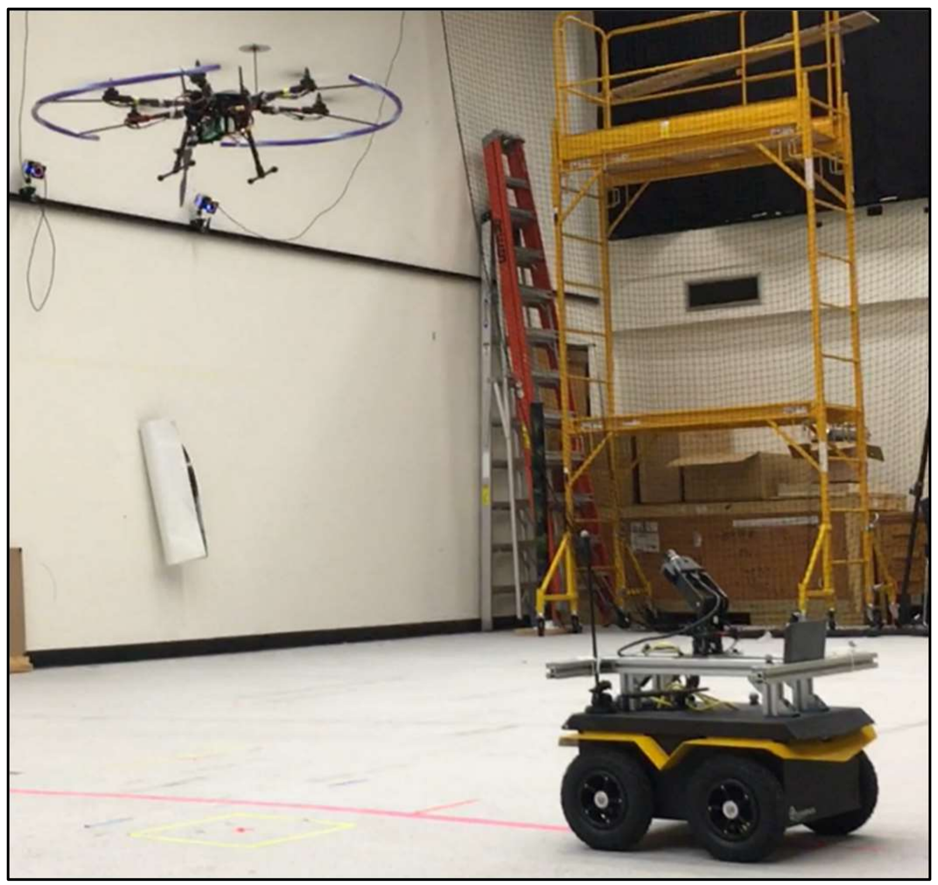
  (particle filters, Gaussian mixture KFs, …)

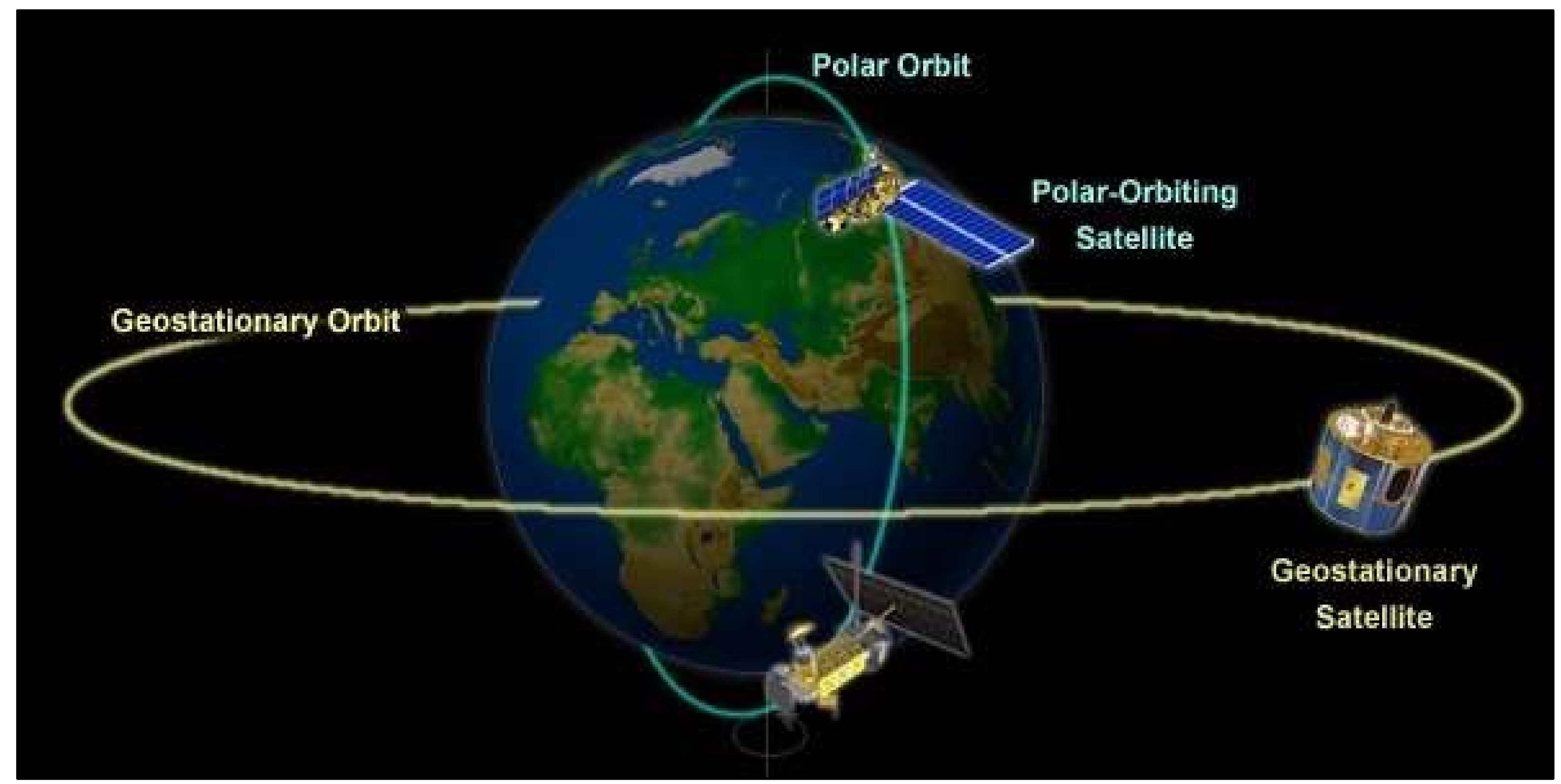
  [advanced classes]



Example Applications: Final Project Systems

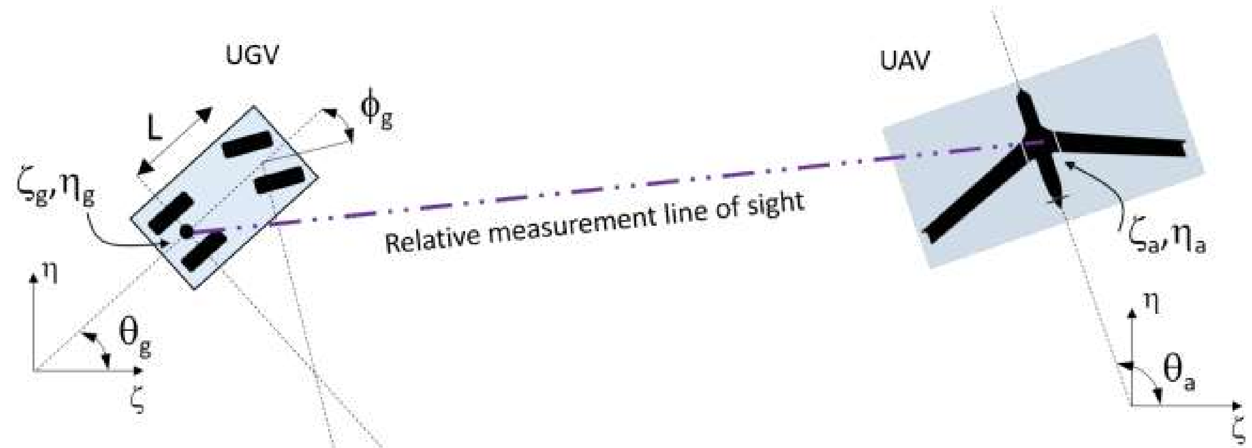
- You will be quite familiar with a non-linear filtering problem by end of semester…

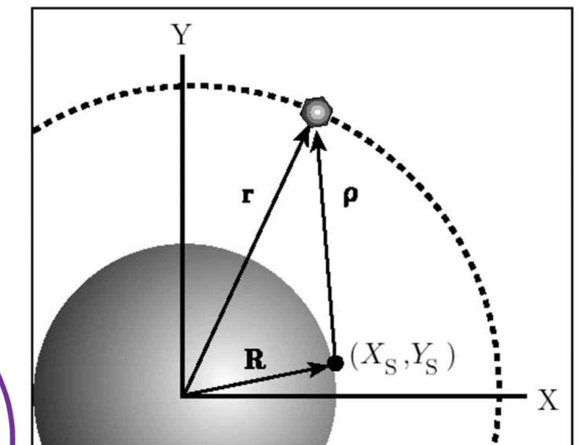




Example Applications: Final Project Systems

- You will be quite familiar with a non-linear filtering problem by end of semester…





$$\dot{\zeta}_g = v_g \cos\theta_g + \tilde{w}_{x,g}$$
$$\dot{\eta}_g = v_g \sin\theta_g + \tilde{w}_{y,g}$$
$$\dot{\theta}_g = \frac{v_g}{L}\tan\phi_g + \tilde{w}_{\omega,g},$$

$$\dot{\zeta}_a = v_a \cos\theta_a + \tilde{w}_{x,a}$$
$$\dot{\eta}_a = v_a \sin\theta_a + \tilde{w}_{y,a}$$
$$\dot{\theta}_a = \omega_a + \tilde{w}_{\omega,a}$$

$\mathcal{F}$ (full nonlinear CT dynamics fxn vector of ODEs)

$$\ddot{X} = -\frac{\mu X}{r^3} + u_1 + \tilde{w}_1$$
$$\ddot{Y} = -\frac{\mu Y}{r^3} + u_2 + \tilde{w}_2$$

$$\mathbf{y}^i(t) = \begin{bmatrix} \rho^i(t) \\ \dot{\rho}^i(t) \\ \phi^i(t) \end{bmatrix} + \tilde{\mathbf{v}}^i(t)$$

$$\mathbf{y}(t) = \begin{bmatrix} \tan^{-1}\left(\frac{\eta_a - \eta_g}{\xi_a - \xi_g}\right) - \theta_g \\ \sqrt{(\xi_g - \xi_a)^2 + (\eta_g - \eta_a)^2} \\ \tan^{-1}\left(\frac{\eta_g - \eta_a}{\xi_g - \xi_a}\right) - \theta_a \\ \xi_a \\ \eta_a \end{bmatrix} + \tilde{\mathbf{v}}(t),$$

$\mathcal{H}$ (full nonlinear CT algebraic meas. vector fxn)

$$\rho^i(t) = \sqrt{(X(t) - X_s^i(t))^2 + (Y(t) - Y_s^i(t))^2}$$
$$\dot{\rho}^i(t) = \frac{[X(t) - X_s^i(t)]\cdot[\dot{X}(t) - \dot{X}_s^i(t)] + [Y(t) - Y_s^i(t)]\cdot[\dot{Y}(t) - \dot{Y}_s^i(t)]}{\rho^i(t)}$$
$$\phi^i(t) = \tan^{-1}\left(\frac{Y(t) - Y_s^i(t)}{X(t) - X_s^i(t)}\right).$$

Non-linear vs. Linear Estimation Problems

- What makes these estimation problems **“non-linear”?**
- Real problems often have non-linearities in dynamics and/or measurements
- Dynamics: solutions to EOM/ODEs do not obey superposition
  - cannot just look at deterministic and random inputs separately
  - no simple closed-form solutions or behaviors (tricky to analyze)
- Measurements: complex relationships to states
  - difficult to “invert” to get state info from sensor data (observability analysis hard)
- Process/sensor noises not necessarily additive (or Gaussian)

- Final project systems all have “smooth” nicely differentiable non-linearities
- Can get “non-smooth” types: e.g. saturation, hysteresis, angle wrap, discrete switches,…
- Linear estimation methods often adapted to “smooth” cases via linearization

→ approx to optimal LS filters: many caveats and no guarantees!!! (but generally still work fine)

(Semi-)formal problem statement…

- How to define an "optimal" state estimator for a nonlinear dynamical system?

$x(t) \in \mathbb{R}^n$

$y(t) \in \mathbb{R}^p$

Given:

$$\dot{x}(t) = \mathcal{F}[x(t), u(t), \tilde{w}(t)]$$

$\mathcal{F}$: nonlinear CT dyn fxn (ODE vector fxn)

$$y(t) = \mathcal{H}[x(t), \tilde{v}(t)]$$

$\mathcal{H}$: " " meas. fxn (algebraic)

state trans. fxn [e.g. integrates $\mathcal{F}$ from time k→k+1, i.e. via ODE45/Runge-Kutta]

$$x(k+1) = f[\underline{x(k)}, \underline{u(k)}, \underline{w(k)}], \qquad w(k) = \mathcal{N}(0, Q) \text{ (AWGN)}$$

$$y(k+1) = h[x(k+1), v(k+1)], \qquad v(k) = \mathcal{N}(0, R) \text{ (AWGN)}$$

nonlinear DT fxns

- Follow same logic as before with linear systems to set up a cost fxn J(K)in DT:

$$\text{let} \;\; e_k^+ = x_k - \hat{x}_k^+,$$

$$J(k) = E[e_k^{+T} e_k^+] = (E[\text{trace}(e_k^+ e_k^{+T})] = \text{trace}(E[e_k^+ e_k^{+T}]) = \text{trace}(P_k^+)$$

FACT: it is possibly to show that, generally, $J(k)$ is minimized by:

$$\hat{x}_k^+ = E[x_k | y_{1:k}]_{p(x_k|y_{1:k})} \text{ (conditional mean of } x_k \text{ given all data } y_1, ..., y_k)$$