ASEN 5044, Fall 2024
Statistical Estimation for Dynamical Systems

Lecture 21 : DT White Noise Modeling; Monte Carlo Simulation of DT Systems; Non-Deterministic Optimal State Estimation

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

Tuesday 10/22/2024

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

Announcements

HW 6 out, due **NEXT Fri Nov 1** via Gradescope

Quiz 6 to be released this Fri (postponed from last week)

- **Midterm 1 grades + solutions to be posted by this Thurs**

- **Midterm 2: will be released Thurs Nov 7, due Thurs Nov 14**

Last Time

- CT and DT LTI Systems Driven by White Noise (Process + Measurement Noise)
- Specifying DT LTI process noise Q covar from CT LTI proc noise PSD W

CT:

$$\dot{x} = Ax(t) + Bu(t) + \Gamma\underline{\underline{\tilde{w}(t)}}$$
$$y(t) = Cx(t) + Du(k) + \tilde{v}(t)$$

$$\tilde{w}(t) = \text{process noise (white: } E[\tilde{w}(t)] = 0,\ E[\tilde{w}(t)\tilde{w}^T(\tau)] = \underline{W} \cdot \delta(t-\tau))$$
$$\tilde{v}(t) = \text{measurement noise (white: } E[\tilde{v}(t)] = 0,\ E[\tilde{v}(t)\tilde{v}^T(\tau)] = V \cdot \delta(t-\tau))$$

DT:

$$x(k+1) = Fx(k) + Gu(k) + \underline{\underline{w(k)}}$$
$$y(k) = Hx(k) + Mu(k) + \underline{v(k)}$$

$$w(k) = \text{process noise (white: } E[w(k)] = 0,\ E[w(k)w^T(j)] = \underline{Q} \cdot \delta(k,j))$$
$$v(k) = \text{measurement noise (white: } E[v(k)] = 0,\ E[v(k)v^T(j)] = R \cdot \delta(k,j))$$

$$Q = \int_0^{\Delta t} e^{A(\Delta t-\tau)}\ \Gamma\ \underline{W}\ \Gamma^T\ e^{A^T(\Delta t-\tau)}\ d\tau$$

Today

- Specifying CT LTI meas noise V PSD + convert to DT LTI meas noise covar R
- **Van Loan's method**: how to actually compute DT covar Q from CT PSD W?
  - Example for 1D robot car
- DT Additive white Gaussian noise (AWGN)
- Monte Carlo simulations of DT stochastic systems with AWGN
- Intro to **optimal non-deterministic state estimation**

Finding DT Noise Parameters: R (measurement noise intensity)

- Now look at the measurement process

$$y(t) = Cx(t) + \tilde{v}(t)$$

$$E[\tilde{v}(t)] = 0,\ E[\tilde{v}(t)\tilde{v}^T(\tau)] = \underline{\underline{V}} \cdot \delta(t-\tau)$$

sample at $\underline{\underline{\Delta T}}$

$$y(k) = Hx(k) + v(k)$$

$$E[v(k)] = 0,\ E[v(k)v^T(j)] = \underline{\underline{R}} \cdot \delta(k,j)$$

→ Given CT intensity V covar matrix, how to convert to DT intensity covar matrix R?

...the wrong way: instantaneous sampling only:

$$y(t) \xrightarrow{t=t_k} y(k) = y(t=t_k) = C\,x(t_k) + \tilde{v}(t_k)$$

[finite ΔT sampling time]

<u>Problem</u>: since $\tilde{v}$ is <u>pure white noise</u>

$$E[\tilde{v}(t_k)\tilde{v}^T(\tau_k)] = \underline{V} \cdot \delta(t_k - \tau_k)$$

→ can have arbitrarily large (∞) variance in the sampled signal!

[all freqs. present in $\tilde{v}(t)$ spectrum!]

Can lead to aliasing of hi-freq. components of $\tilde{v}(t)$ w.r.t. finite sample time ΔT!

<u>...the right way:</u> Need to avoid aliasing, so use an anti-aliasing (AAF) <u>before</u> sample $y(k)$:

$$y(t) \longrightarrow \boxed{\text{AAF}\ \ \frac{1}{\Delta T}\int_{t_k-\Delta T}^{t_k} y(\tau)\,d\tau} \xrightarrow{\Delta T} y(k)$$

Models an "average & dump" operator

Autocorr fxn for $v(k)$: let $R(k) = R(t_k)$ → use mean-value theorem

$$R(k) = \frac{1}{\Delta T^2}\iint_{t_k-\Delta T}^{t_k} \underbrace{E[\tilde{v}(\tau_1)\tilde{v}^T(\tau_2)]}_{\text{apply def'n w/ Dirac Delta properties}}\,d\tau_1\,d\tau_2 = \boxed{\frac{\underline{V}}{\Delta T}\cdot\delta(j,k) = R(k)}$$

Additive White Gaussian Noise (AWGN)

- Most often we assume w(k) and v(k) are distributed as zero-mean Gaussian random vectors for some known intensity covar matrices Q(k) and R(k)
- Then we say that DT LTI system is driven by **additive white Gaussian noise (AWGN)**

$w(k) \underset{\text{iid}}{\sim} \mathcal{N}(0, Q(k)) \longleftrightarrow$ AWG proc. noise

$v(k) \underset{\text{iid}}{\sim} \mathcal{N}(0, R(k)) \longleftrightarrow$ AWG meas. noise

* w(k) & v(k) must be zero mean to be AWGN

- Justification: Central Limit Theorem
  - Although: in many cases can still have noise that is not additive, non-white, non-Gaussian, yet still has E[w(k)] = 0, E[v(k)] =0, E[w(k)w(j)^T] = Q(k), E[v(k)v(j)^T]=R(k)

Calculating Q (DT process noise intensity covariance) from W (CT process noise PSD intensity covariance) – Van Loan's Method

- We showed for CT → DT LTI conversion: $E[w(k)] = 0$ and $E[w(k)w(j)^T] = Q \cdot \delta(k,j)$

$$\text{where } Q = \int_0^{\Delta t} e^{A(\Delta t - \tau)}\ \underline{\Gamma\ W\ \Gamma^T}\ e^{A^T(\Delta t - \tau)}\ d\tau$$

- But how to actually compute Q?
- We can use **Van Loan's method**: starting with CT LTI system specifications:

Given:
$$\begin{aligned} \dot{x} &= \underline{A}x(t) + Bu(t) + \underline{\Gamma}\tilde{w}(t) \\ E[\tilde{w}(t)] &= 0,\ E[\tilde{w}(t)\tilde{w}^T(\tau)] = \underline{W} \cdot \delta(t-\tau) \end{aligned} \rightarrow \underline{\text{given}}\ \ (\underline{A},\ \underline{\Gamma},\ \underline{W}) \text{ for sample time } \underline{\Delta t},$$

Step 1: form block matrix $Z = \underline{\Delta t} \cdot \begin{bmatrix} -A & \Gamma W \Gamma^T \\ 0 & A^T \end{bmatrix}$

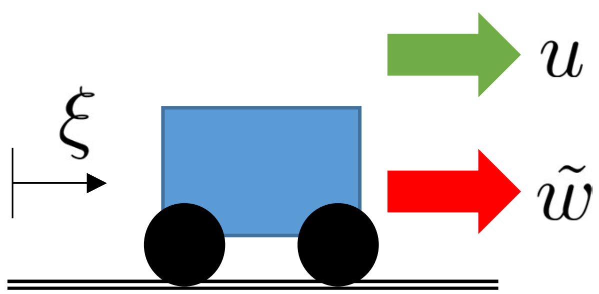
same size as $\underbrace{\Gamma W \Gamma^T}_{n \times n}$

Step 2: compute matrix exponential $e^Z = \text{expm(Z)} \rightarrow e^Z = \begin{bmatrix} (\cdots) & F^{-1}Q \\ 0 & F^T \end{bmatrix}$

Step 3: use matrices in right block column to solve for $Q \rightarrow \underset{[n \times n]}{Q} = (F^T)^T \cdot [F^{-1}Q]$

Example: Converting Stochastic Robot Cart Dynamics from CT to DT

- **Given:**



$$\dot{x} = Ax(t) + Bu(t) + \Gamma\tilde{w}(t), \quad A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$x = \begin{bmatrix} \xi \\ \dot{\xi} \end{bmatrix}$ $\rightarrow$ want convert to DT dynamics model for given Δt:

$$x(k+1) = Fx(k) + Gu(k) + w(k),$$

$\rightarrow$ for deterministic part of system, easy to show: $F = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \quad G = \begin{bmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \end{bmatrix}$

$\rightarrow$ Now, suppose $\tilde{w}$ is modeled as AWGN: $\tilde{w}(t) \sim \mathcal{N}(0, W \cdot \delta(t-\tau))$ (CT white process noise)

$\rightarrow$ find corresponding Q for DT model: $w(k) \sim \mathcal{N}(0, Q)$ (DT white process noise)

(i.e. DT process noise will effectively 'sample' a Gaussian random vector $w(k)$
at each time step k to summarize effect of $\tilde{w}(t)$ on $x(t)$ from $t = k \rightarrow t = k+1$)

$\rightarrow$ let's assume $\Delta t = 0.1$ sec and $W = 1 \; (\frac{\text{m}}{\text{s}^2})^2$

Example: Converting Stochastic Robot Cart from CT to DT

- Now apply Van Loan's method to find Q:

Step 1: form block matrix $Z = \Delta t \cdot \begin{bmatrix} -A & \Gamma W \Gamma^T \\ 0 & A^T \end{bmatrix} = (0.1) \cdot \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

Step 2: compute matrix exponential $e^Z = \text{expm(Z)}$

$$\rightarrow e^Z = \begin{bmatrix} (\cdots) & (F^{-1}Q) \\ 0 & F^T \end{bmatrix} = \begin{bmatrix} 1 & -0.1 & -1.67 \times 10^{-4} & 5 \times 10^{-3} \\ 0 & 1 & 5 \times 10^{-3} & 0.1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0.1 & 1 \end{bmatrix}$$

Step 3: use matrices in right block column to solve for $Q \rightarrow Q = (F^T)^T \cdot [F^{-1}Q]$

[m²] total variance in pos'n state disturbance added in by proc. noise accel. $\tilde{w}$ over Δt time step.

$$\rightarrow Q = \begin{bmatrix} 3 \times 10^{-4} & 5 \times 10^{-3} \\ 5 \times 10^{-3} & 0.1 \end{bmatrix}$$

linear integration coupling

cross/covariance of pos'n & velocity disturbance due to proc. noise

$[m \cdot \frac{m}{s}]$

$[\frac{m}{s}]^2$ total variance in velocity disturbance added in by proc. noise accel $\tilde{w}$ over Δt time step.

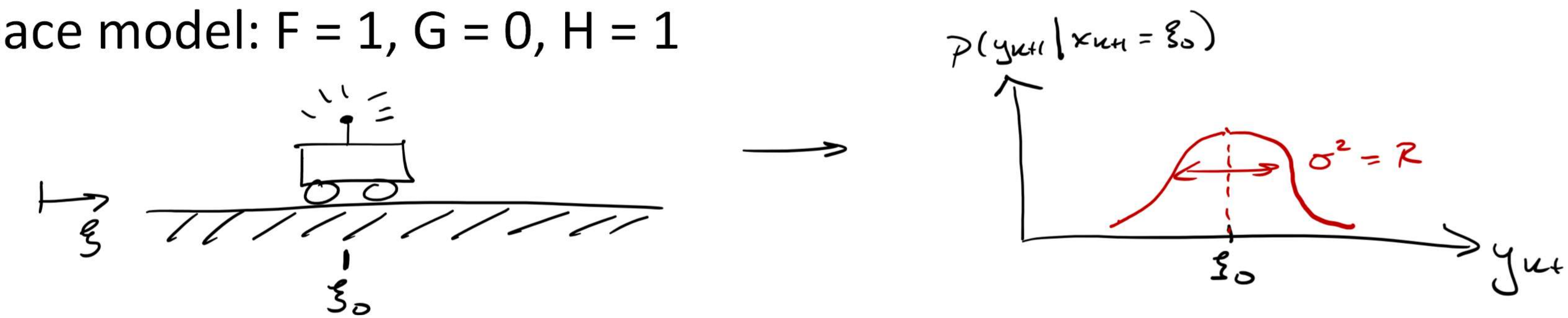
Monte Carlo Sampling of Gaussian Random Vectors

- How to simulate AWGN?
- Example: suppose our 1D robot is sitting perfectly still, collecting GPS position data

$$x(k+1) = x(k) = \xi_0 \text{ (position } \xi_0 \text{ is fixed/constant for all time, no proc. noise)}$$

$$y(k+1) = x(k+1) + v(k+1),\ v(k+1) \sim \mathcal{N}(0,R) \text{ (suppose } R = 1 \text{ m}^2)$$

- DT state space model: F = 1, G = 0, H = 1

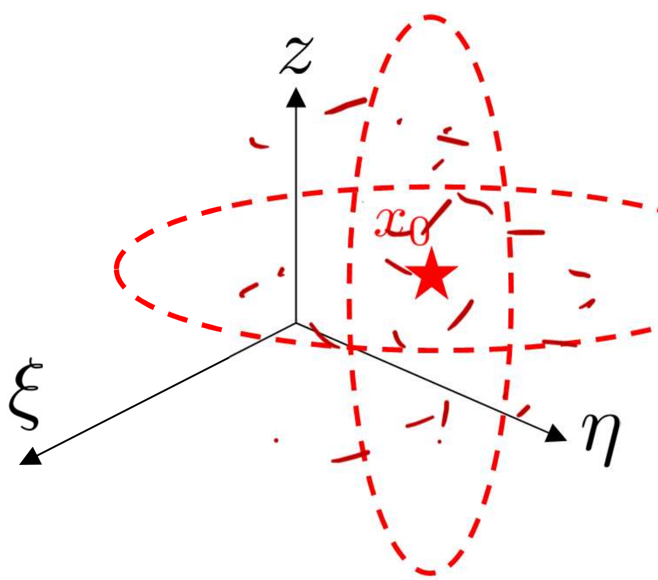


- How to simulate noisy position measurements y(k+1)?
  - Need to draw from scalar Gaussian distribution: $y(k+1) \sim \mathcal{N}(Hx_{k+1}, R)$ $= \mathcal{N}(\xi_0, R) = p(y_{k+1} | x_{k+1} = \xi_0)$
- Recall formula for scalar case (use “randn” in Matlab):

$$\text{for time } k = 1:T,\ \text{ compute } y(k) = Hx_k + \sqrt{R}\cdot q_k,\ \text{ where } q_k \sim N(0,1)\ \text{ (i.e. } q_k = \text{randn in Matlab)}$$

Monte Carlo Sampling of Gaussian Random Vectors

- Now suppose we look at 3D position measurements for static position vector



$$x(k) = \begin{bmatrix} \xi(k) \\ \eta(k) \\ z(k) \end{bmatrix} \quad \begin{pmatrix} \text{Easting} \\ \text{Northing} \\ \text{height} \end{pmatrix} \qquad x(k+1) = x(k) = \text{const.} = \begin{bmatrix} \xi(0) \\ \eta(0) \\ z(0) \end{bmatrix} \begin{matrix} = x(0) \\ = x_0 \end{matrix}$$

$$y(k+1) = x(k+1) + v(k+1) \longleftrightarrow H = I_{3\times 3}$$

[$x_0 + v(k+1)$ in this case)

$$v(k+1) \sim \mathcal{N}(0, R), \qquad R = \begin{bmatrix} 3 & 0.1 & 0.01 \\ 0.1 & 3 & 0.01 \\ 0.01 & 0.01 & 10 \end{bmatrix}$$

- In this case, we have position error correlations from GPS receiver via R matrix

(errors in diff. directions are not $\not\perp$ of each other)

- So to simulate T measurements, need to draw T i.i.d. Gaussian vectors $y_k \sim N(Hx_k, R)$

(following logic of prev. slide: $p(y_{k+1} \mid x_{k+1} = x_0) = N(Hx_{k+1}, R)$)

Monte Carlo Sampling of Gaussian Random Vectors

- Can generalize sampling algorithm from Gaussian scalar case to Gaussian vector case
- A "matrix square root" of positive definite symmetric R matrix can be obtained via the **Cholesky decomposition** ("chol.m" in Matlab):

$$\mathrm{chol}(R, \text{'}lower') = S_v, \text{ such that } S_v S_v^T = R \; (= \underline{S_v} \cdot \underline{I} \cdot \underline{S_v^T})$$

where $R \in \mathbb{R}^{p \times p}$, $S_v \in \mathbb{R}^{p \times p} =$ (lower triangular) 'matrix square root of R'

pxl random vector

$\rightarrow$ so apply linear transformation to standard normal random vector $\underline{q} \sim \mathcal{N}(0, \underline{\underline{I_{p\times p}}})$

such that $y = \underline{m_y} + \underline{\underline{S_v}} \cdot q$ $\rightarrow$ we know from linear transformations of Gaussian random vectors (Lecture 16) that $y \sim \mathcal{N}(\underline{\underline{m_y}}, S_v \cdot \underline{\underline{I}} \cdot S_v^T) = \mathcal{N}(m_y, R)$

($H x_{k+1}$) for prev slide

$\rightarrow$ general algorithm for simulating $p-$dim. Gaussian random vector $y_k \sim \mathcal{N}(m_y, R)$:

1. compute $S_v = \mathrm{chol}(R, \text{'}lower')$, such that $S_v S_v^T = R$
2. draw sample $q_k = \mathcal{N}(0, I_{p\times p})$ [e.g. using randn in Matlab] [or mvnrnd.m in Matlab]
3. compute $y_k = m_y + S_v \cdot q_k$, where $m_y \in \mathbb{R}^p$ is the desired mean