Ch 18 State Space Models

18.1 Introduction

- hidden state가 continuous인 점만 빼고는 HMM과 거의 동일
- 구성
  - state transition model with system-error
  - observation model with observation-error

$$
\begin{aligned}
\mathbf{z}_t &= g(\mathbf{u}_t, \mathbf{z}_{t-1}, \boldsymbol{\epsilon}_t) &\quad (18.1) \\
\mathbf{y}_t &= h(\mathbf{z}_t, \mathbf{u}_t, \boldsymbol{\delta}_t) &\quad (18.2)
\end{aligned}
$$

- 주요 과제
  - exact/approximate inference
  - learning : estimate model parameters
- linear-gaussian SSM (LG-SSM)
  - easy case
    - hard case : non-linear case
  - support exact inference
    - kalman filtering

- The transition model is a linear function

$$\mathbf{z}_t = \mathbf{A}_t \mathbf{z}_{t-1} + \mathbf{B}_t \mathbf{u}_t + \boldsymbol{\epsilon}_t \quad (18.3)$$

- The observation model is a linear function

$$\mathbf{y}_t = \mathbf{C}_t \mathbf{z}_t + \mathbf{D}_t \mathbf{u}_t + \boldsymbol{\delta}_t \quad (18.4)$$

- The system noise is Gaussian

$$\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_t) \quad (18.5)$$

- The observation noise is Gaussian

$$\boldsymbol{\delta}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_t) \quad (18.6)$$

18.2 Application

18.2.1 SSM for object tracking

- tracking object by kalman filter from noisy measurements
- example : random acceleration model
  - object moving in 2D-space
  - move at constant velocity
  - perturbed by random gaussian noise, ex. wind
    - velocity jump by system noise
- model
  - state vector : position and velocity
  - observation vector : position only
  - LG-SSM assumption

$$\mathbf{z}_t^T = \begin{pmatrix} z_{1t} & z_{2t} & \dot{z}_{1t} & \dot{z}_{2t} \end{pmatrix}. \tag{18.7}$$

$$\mathbf{z}_t = \mathbf{A}_t \mathbf{z}_{t-1} + \boldsymbol{\epsilon}_t \tag{18.8}$$

$$\begin{pmatrix} z_{1t} \\ z_{2t} \\ \dot{z}_{1t} \\ \dot{z}_{2t} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \Delta & 0 \\ 0 & 1 & 0 & \Delta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} z_{1,t-1} \\ z_{2,t-1} \\ \dot{z}_{1,t-1} \\ \dot{z}_{2,t-1} \end{pmatrix} + \begin{pmatrix} \epsilon_{1t} \\ \epsilon_{2t} \\ \epsilon_{3t} \\ \epsilon_{4t} \end{pmatrix} \tag{18.9}$$

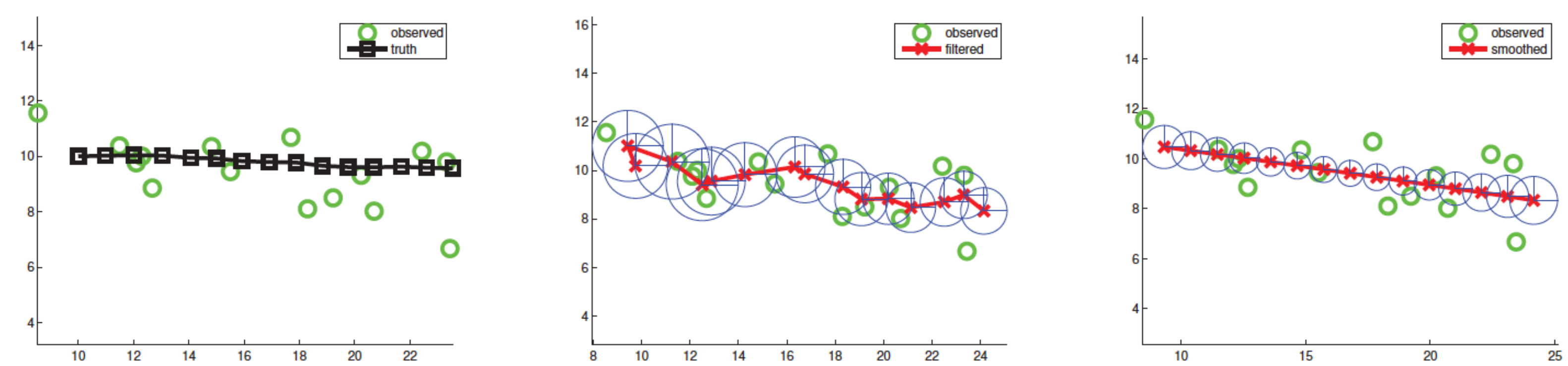
$$\mathbf{y}_t = \mathbf{C}_t \mathbf{z}_t + \boldsymbol{\delta}_t \tag{18.10}$$

$$\begin{pmatrix} y_{1t} \\ y_{2t} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} z_{1t} \\ z_{2t} \\ \dot{z}_{1t} \\ \dot{z}_{2t} \end{pmatrix} + \begin{pmatrix} \delta_{1t} \\ \delta_{2t} \\ \delta_{3t} \\ \delta_{4t} \end{pmatrix} \tag{18.11}$$

- initial belief and notation

if the initial belief state is Gaussian, $p(\mathbf{z}_1) = \mathcal{N}(\boldsymbol{\mu}_{1|0}, \boldsymbol{\Sigma}_{1|0})$, then all subsequent belief states will also be Gaussian; we will denote them by $p(\mathbf{z}_t|\mathbf{y}_{1:t}) = \mathcal{N}(\boldsymbol{\mu}_{t|t}, \boldsymbol{\Sigma}_{t|t})$. (The notation $\boldsymbol{\mu}_{t|\tau}$ denotes $\mathbb{E}[\mathbf{z}_t|\mathbf{y}_{1:\tau}]$, and similarly for $\boldsymbol{\Sigma}_{t|t}$; thus $\boldsymbol{\mu}_{t|0}$ denotes the prior for $\mathbf{z}_1$ before we have seen any data. For brevity we will denote the posterior belief states using $\boldsymbol{\mu}_{t|t} = \boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_{t|t} = \boldsymbol{\Sigma}_t$.) We can compute these quantities efficiently using the celebrated Kalman filter,

- estimation result
  - a) raw observation 은 measurement error 때문에 매우 noisy하다.
  - b) kalman filtering 적용 결과
    - smmoth 해졌다.
    - 원은 p(z|y), 즉 posterior가 gaussian임을 의미, 원 중심은 posterior mode에 해당
  - c) kalman smoothing 적용 결과



(a) (b) (c)

18.2.2 Robotic SLAM (simultaneous localization and mapping)

- robot moving around unknown 2d world
- learn a map and keep track of its location within that map
- model
  - state vector
    - 로봇의 위치 at time t
    - L개의 landmark의 location at time t
      - fixed, no system error
  - observation
    - measurement of distances from robot location to the set of closest landmark

- How it works
  - inital guess about robot and landmark location
  - move arount and find some of landmark
  - measurement and update prior belief

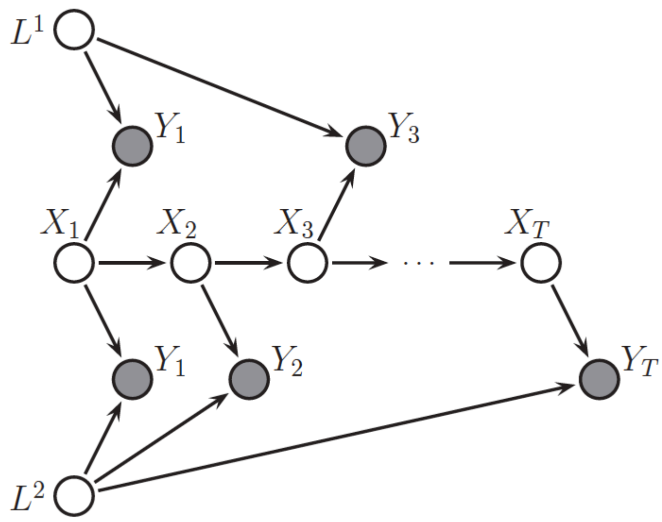
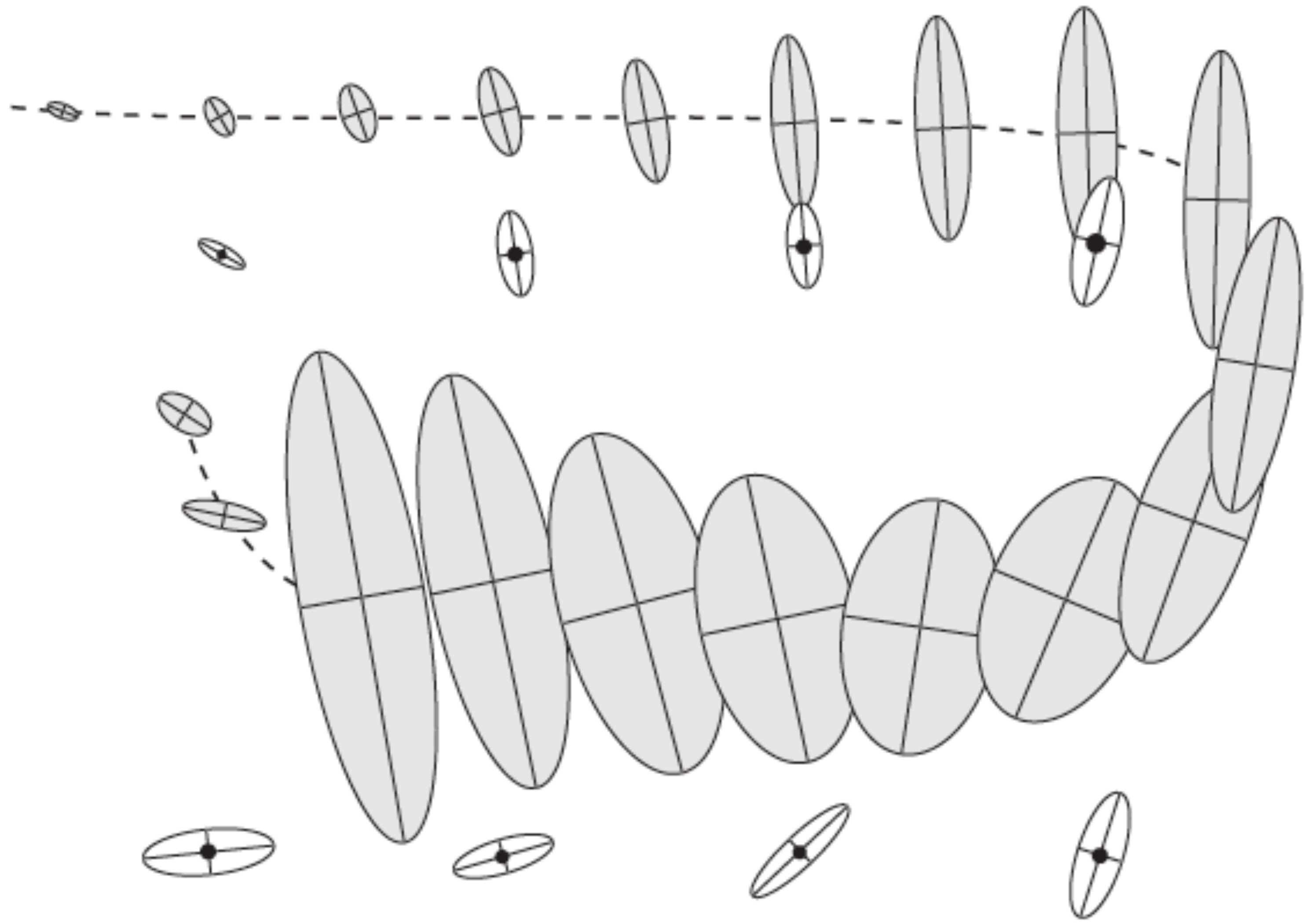


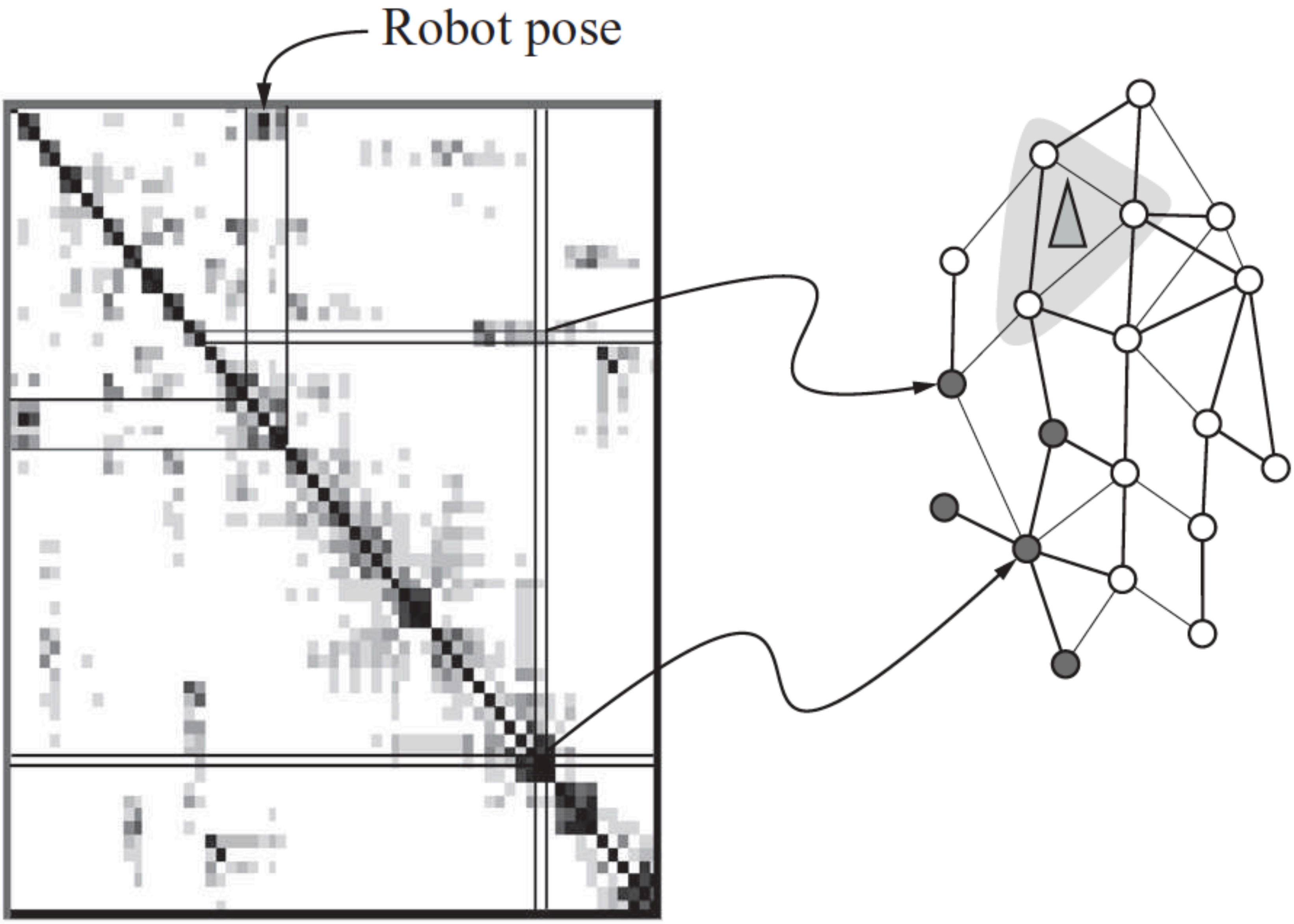
Figure 18.2 Illustration of graphical model underlying SLAM. L^i is the fixed location of landmark i, $\mathbf{x}_t$ is the location of the robot, and $\mathbf{y}_t$ is the observation. In this trace, the robot sees landmarks 1 and 2 at time step 1, then just landmark 2, then just landmark 1, etc. Based on Figure 15.A.3 of (Koller and Friedman 2009).

- Close-loop 현상
  - over time, uncertainty in robot location increase due to system-error
  - but when robot return to a familiar location, its uncertainty will decrease agamin
  - with smoothing, this shrink propagate back to the past trajectory



(a)

- Convert to undirected graphical model
  - due to sparcity of posterior precision matrix
  - zero in matrix means no edges in UGM
  - at prior, only diagonal terms non-zero => UGM as disconnected graph
  - as robot moves, correlation between near-by landmarks induced



(b)

- Entanglement problem
  - estimated landmark locations are not independent
  - because all depends on estimated robot locations
    - not d-seperated by observation : L1-Y1-X1-Y1-L2 경로
  - over time, precision matrix denser
  - hard to perform exact inference, O(K^3)
  - solution
    - dynamically prune out ewak edegs (junction tree in section 20.4)
    - Fast SLAM : combination of kalman filter and particle filtering (section 23.6.3)

18.3 Inference in LG-SSM

- on-line case => analgous to the forward algorithm for HMMs
  - predict and correct
- off-line case => analgous to the forward-backward algorithm for HMMs

18.3.1 Kalman filtering algorithm

- 기본 idea
  - predict : 현재까지의 정보로 다음 time-step을 예측
  - correct : 실제 관측된 사실을 바탕으로 잘못된 예측 모형을 바로잡음 (error-based correction)

- 사실상 강화학습의 TD-based 와 동일한 컨셉

- marginal posterior at time t

$$p(\mathbf{z}_t|\mathbf{y}_{1:t}, \mathbf{u}_{1:t}) = \mathcal{N}(\mathbf{z}_t|\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \tag{18.24}$$

- prediction-step

$$\begin{aligned} p(\mathbf{z}_t|\mathbf{y}_{1:t-1}, \mathbf{u}_{1:t}) &= \int \mathcal{N}(\mathbf{z}_t|\mathbf{A}_t\mathbf{z}_{t-1} + \mathbf{B}_t\mathbf{u}_t, \mathbf{Q}_t)\mathcal{N}(\mathbf{z}_{t-1}|\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1})d\mathbf{z}_{t-1} && (18.25) \\ &= \mathcal{N}(\mathbf{z}_t|\boldsymbol{\mu}_{t|t-1}, \boldsymbol{\Sigma}_{t|t-1}) && (18.26) \\ \boldsymbol{\mu}_{t|t-1} &\triangleq \mathbf{A}_t\boldsymbol{\mu}_{t-1} + \mathbf{B}_t\mathbf{u}_t && (18.27) \\ \boldsymbol{\Sigma}_{t|t-1} &\triangleq \mathbf{A}_t\boldsymbol{\Sigma}_{t-1}\mathbf{A}_t^T + \mathbf{Q}_t && (18.28) \end{aligned}$$

- update by bayes rule
  - bayes rule : P(A|B) = (P(B)*P(B|A)) / (P(A)

The measurement step can be computed using Bayes rule, as follows

$$p(\mathbf{z}_t|\mathbf{y}_t, \mathbf{y}_{1:t-1}, \mathbf{u}_{1:t}) \propto p(\mathbf{y}_t|\mathbf{z}_t, \mathbf{u}_t)p(\mathbf{z}_t|\mathbf{y}_{1:t-1}, \mathbf{u}_{1:t}) \tag{18.29}$$

- error-signal
  - true measurement - guessed measurement

$$\begin{aligned} \mathbf{r}_t &\triangleq \mathbf{y}_t - \hat{\mathbf{y}}_t && (18.33) \\ \hat{\mathbf{y}}_t &\triangleq \mathbb{E}\left[\mathbf{y}_t|\mathbf{y}_{1:t-1}, \mathbf{u}_{1:t}\right] = \mathbf{C}_t\boldsymbol{\mu}_{t|t-1} + \mathbf{D}_t\mathbf{u}_t && (18.34) \end{aligned}$$

- kalman gain factor
  - 역할 : update rule에서 일종의 learning rate에 해당
  - 이것도 정의는 아니고 원래 유도되는 것인데, 책에서는 그냥 define 으로 표기
  - prediction error를 minimize하는 objective를 설정해서 유도할 수 있음
  - 보다 직관적인 유도 과정은 다음 비디오 playlist 참고
    - https://www.youtube.com/playlist?list=PLsrI8QPOTtbbxi-Y5FyLOIX8dBmOtQMoU (https://www.youtube.com/playlist?list=PLsrI8QPOTtbbxi-Y5FyLOIX8dBmOtQMoU)

$$\mathbf{K}_t \triangleq \boldsymbol{\Sigma}_{t|t-1}\mathbf{C}_t^T\mathbf{S}_t^{-1} \tag{18.35}$$

where

$$\begin{aligned} \mathbf{S}_t &\triangleq \operatorname{cov}\left[\mathbf{r}_t|\mathbf{y}_{1:t-1}, \mathbf{u}_{1:t}\right] && (18.36) \\ &= \mathbb{E}\left[(\mathbf{C}_t\mathbf{z}_t + \boldsymbol{\delta}_t - \hat{\mathbf{y}}_t)(\mathbf{C}_t\mathbf{z}_t + \boldsymbol{\delta}_t - \hat{\mathbf{y}}_t)^T|\mathbf{y}_{1:t-1}, \mathbf{u}_{1:t}\right] && (18.37) \\ &= \mathbf{C}_t\boldsymbol{\Sigma}_{t|t-1}\mathbf{C}_t^T + \mathbf{R}_t && (18.38) \end{aligned}$$

- final update rule
  - mean update만 생각하면 일종의 gradient-descent와 유사
  - kalman gain factor가 일종의 time-varying learning rate가 됨
    - ratio between prior and measurement error
    - strong prior and/or noisy measurement => small learning rate
    - weak prior and precise measurement => large learning rate

$$
\begin{aligned}
p(\mathbf{z}_t|\mathbf{y}_{1:t}, \mathbf{u}_t) &= \mathcal{N}(\mathbf{z}_t|\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) && (18.30)\\
\boldsymbol{\mu}_t &= \boldsymbol{\mu}_{t|t-1} + \mathbf{K}_t\mathbf{r}_t && (18.31)\\
\boldsymbol{\Sigma}_t &= (\mathbf{I} - \mathbf{K}_t\mathbf{C}_t)\boldsymbol{\Sigma}_{t|t-1} && (18.32)
\end{aligned}
$$

- posterior predictive
  - usefule for time-serires forcasting

$$
\begin{aligned}
p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \mathbf{u}_{1:t}) &= \int \mathcal{N}(\mathbf{y}_t|\mathbf{C}\mathbf{z}_t, \mathbf{R})\mathcal{N}(\mathbf{z}_t|\boldsymbol{\mu}_{t|t-1}, \boldsymbol{\Sigma}_{t|t-1})d\mathbf{z}_t && (18.42)\\
&= \mathcal{N}(\mathbf{y}_t|\mathbf{C}\boldsymbol{\mu}_{t|t-1}, \mathbf{C}\boldsymbol{\Sigma}_{t|t-1}\mathbf{C}^T + \mathbf{R}) && (18.43)
\end{aligned}
$$

- log-likelihood of specific measurement sequences

$$
\log p(\mathbf{y}_{1:T}|\mathbf{u}_{1:T}) = \sum_t \log p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \mathbf{u}_{1:t}) \tag{18.40}
$$

where

$$
p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \mathbf{u}_{1:t}) = \mathcal{N}(\mathbf{y}_t|\mathbf{C}_t\boldsymbol{\mu}_{t|t-1}, \mathbf{S}_t) \tag{18.41}
$$

Kalman filter의 시간 복잡도와 SMM의 Application 등은 다음 발표시간에 보충