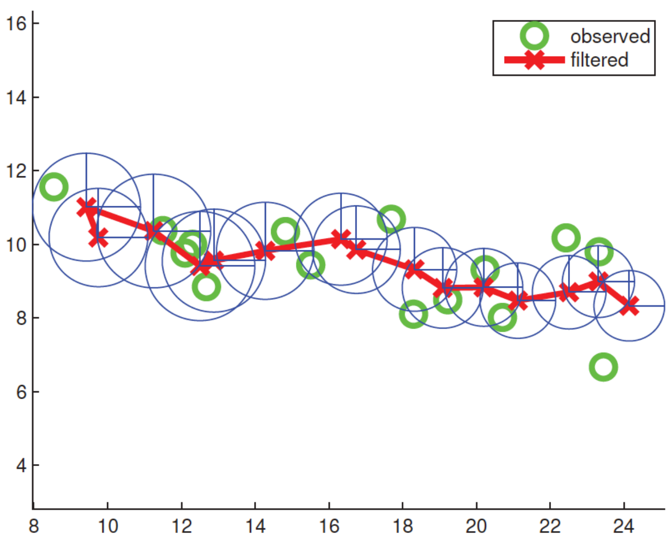
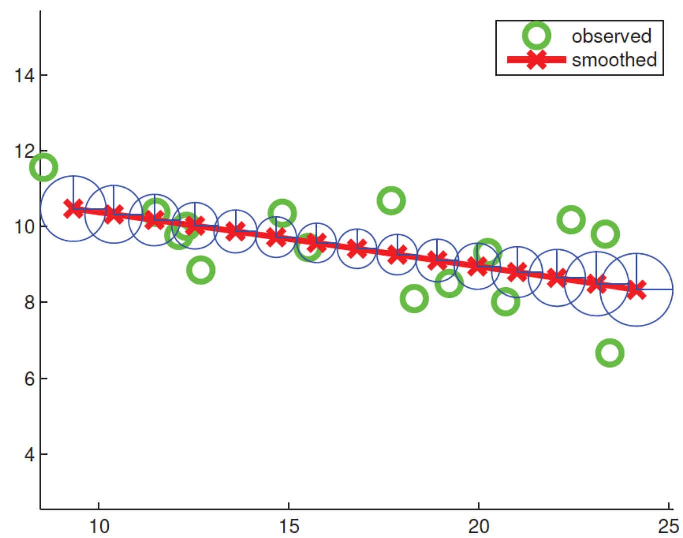
Ch 18 State Space Models

18.3.2 Kalman smoothing algorithm

- kalman filter
  - p(z_t|y_1:t), no future data, online like tracking
- kalman smoothing
  - p(z_t|y_1:T), with future data, offline
  - 미래 데이터 활용으로 인해 불확실성 감소
    - 아래 (c) 처럼 neighbor가 많은 부분은 불확실성이 감소
    - 시작과 끝부분은 상대적으로 neighbor가 적어서 감소 효과가 적다



(b)



(c)

notation recall

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,

algorithm

- HMM의 forward-backward 알고리즘과 유사
  - 즉 UGM에서의 message passing의 원리
  - 왼쪽에서 시작해서 오른쪽으로 그래프 끝까지 전파, p(z_T|y_1:T)
  - 이를 다시 거꾸로 역전파(future back to the past)
  - 정방향 정보와 역방향 정보를 결합
- HMM의 forward-backward 와의 차이점
  - HMM의 backward는 forward를 선행하지 않고 단독으로 진행 가능
    - 그 과정중 observation data 필요
  - kalman smoothing에서는 forward가 반드시 선행되야 함.

- 반면 observation data 필요없고, 정방향 정보만 필요

$$
\begin{aligned}
p(\mathbf{z}_t|\mathbf{y}_{1:T}) &= \mathcal{N}(\boldsymbol{\mu}_{t|T}, \boldsymbol{\Sigma}_{t|T}) && (18.56)\\
\boldsymbol{\mu}_{t|T} &= \boldsymbol{\mu}_{t|t} + \mathbf{J}_t(\boldsymbol{\mu}_{t+1|T} - \boldsymbol{\mu}_{t+1|t}) && (18.57)\\
\boldsymbol{\Sigma}_{t|T} &= \boldsymbol{\Sigma}_{t|t} + \mathbf{J}_t(\boldsymbol{\Sigma}_{t+1|T} - \boldsymbol{\Sigma}_{t+1|t})\mathbf{J}_t^T && (18.58)\\
\mathbf{J}_t &\triangleq \boldsymbol{\Sigma}_{t|t}\mathbf{A}_{t+1}^T\boldsymbol{\Sigma}_{t+1|t}^{-1} && (18.59)
\end{aligned}
$$

derivation - skip

18.4 Learning for LG-SSM

- how to estimate parameters of of LG-SSM, namely, A,C,Q,R below
  - in control theory, called system identification

- The transition model is a linear function

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

- The observation model is a linear function

$$\mathbf{y}_t = \mathbf{C}_t\mathbf{z}_t + \mathbf{D}_t\mathbf{u}_t + \boldsymbol{\delta}_t$$

- The system noise is Gaussian

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

- The observation noise is Gaussian

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

- special case - well known A, C
  - ex) time series forcasting, or physical state estimation
  - 단지 Q와 R만 추정하면 된다.

Identifiability and numerical stability

- without loss of generality
  - assume Q = I => noise 는 A로 적절히 표현 가능
  - assume R to be diagonal => reduce DOF => improve numerical stability
- eigenvalue of A
  - if no system noise, and eigenvalue greater than 1 => blow up
  - all eigenvalues be less than 1 with non-zero noise

$$\mathbf{z}_t = \mathbf{A}^t\mathbf{z}_1 = \mathbf{U}\boldsymbol{\Lambda}^t\mathbf{U}^{-1}\mathbf{z}_1$$

where $\mathbf{U}$ is the matrix of eigenvectors for $\mathbf{A}$, and $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_i)$ c

EM for LG-SSM

- HMM 의 Baum-welch 와 유사,
  - 참고 : https://web.stanford.edu/~lmackey/stats306b/doc/stats306b-spring14-lecture11_scribed.pdf (https://web.stanford.edu/~lmackey/stats306b/doc/stats306b-spring14-lecture11_scribed.pdf)

$$\log p(x_{0:T}, z_{0:T}; \theta) = -\frac{1}{2}\Bigg(\log|\Sigma_0| + z_0^T\Sigma_0^{-1}z_0 + \sum_{t=1}^{T}\log|Q| + (z_t -$$

$$+\sum_{t=0}^{T}\log|R| + (x_t - Cz_t)^T R^{-1}(x_t - Cz_t)\Bigg) + \text{co}$$

In the E step we form the expected complete log likelihood under tl

$$q_s(z_{0:T}) = p(z_{0:T}|x_{0:T}; \theta^{(s)}),$$

11.5.2 M step

Now, we optimize the ECLL. As in factor analysis, there exists a cl

$$C^{(s+1)} = \left(\sum_{t=0}^{T} x_t \mathbb{E}_{q_s}[z_t^T]\right)\left(\sum_{t=0}^{T}\mathbb{E}_{q_s}[z_t z_t^T]\right)^{-1}$$

$$A^{(s+1)} = \left(\sum_{t=1}^{T}\mathbb{E}_{q_s}[z_t z_{t-1}^T]\right)\left(\sum_{t=1}^{T}\mathbb{E}_{q_s}[z_{t-1}z_{t-1}^T]\right.$$

Subspace methods for LG-SSM

- EM 방식의 단점
  - 초기 parameter 추정
- Subspace methods for system identification
  - project y onto subspace of z, and using PCA, identify subspace
  - https://www.youtube.com/watch?v=EMdrHvYd_Zs (https://www.youtube.com/watch?v=EMdrHvYd_
  - https://simsee.org/simsee/biblioteca/Springer,%20Subspace%20Methods%20For%20System%20Io (https://simsee.org/simsee/biblioteca/Springer,%20Subspace%20Methods%20For%20System%20I

Baysian methods for LG-SSM

- offline baysian alternatives
  - Variational Bayes EM (Beal 2003)
  - Blocked Gibbs Sampling (Carter 1994)

18.5 Approximate online inference for non-linearm non-gaussian SSMs

- 지금까지는 easy-case
  - linear and gauusian model
  - gaussian noise
- 일반적으로는
  - non-linear, non-gaussian model
  - non-gaussian noise

18.5.1 Extended Kalman filter(EKF)

- 약간 어려운 case
  - non-liear model
  - gassuian nose
- approximate inference
  - approximate the posterior by a Gaussian
  - Y=f(X)
    - X be gaussian
    - f be non-linear function
    - Y be approximate gaussian
- 기본 아이디어
  - 각 시점에서 선형 근사화(first-order taylor approx)
  - 그런다음 standard kalman filter 적용

$$\mathbf{z}_t = g(\mathbf{u}_t, \mathbf{z}_{t-1}) + \mathcal{N}(\mathbf{0}, \mathbf{Q}_t) \quad (18.80)$$
$$\mathbf{y}_t = h(\mathbf{z}_t) + \mathcal{N}(\mathbf{0}, \mathbf{R}_t) \quad (18.81)$$

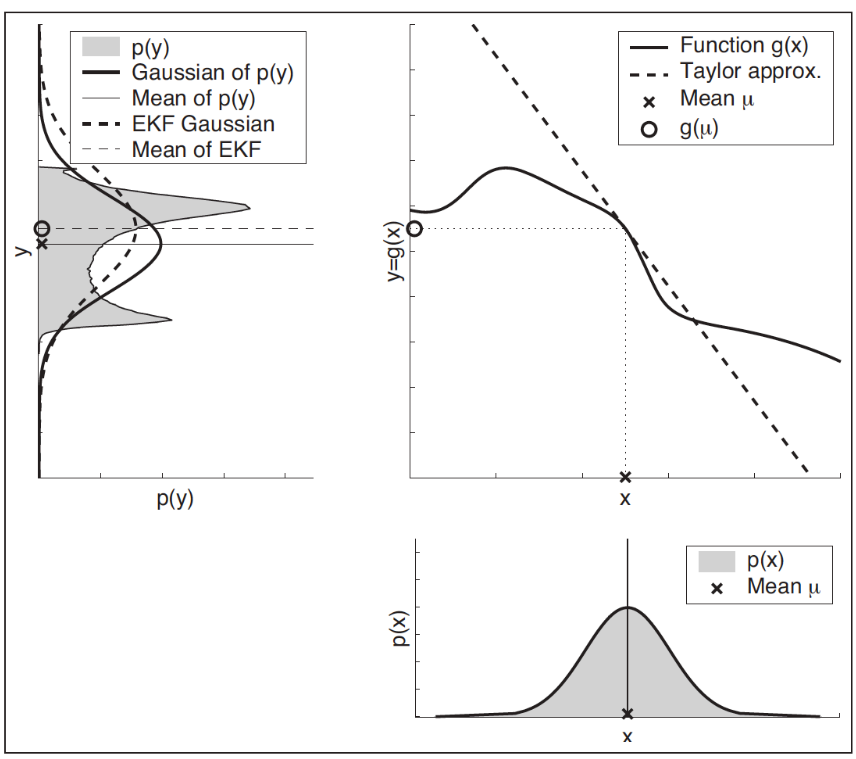


Figure 18.9 Nonlinear transformation of a Gaussian random variable. The prior $p(x)$ is shown on the bottom right. The function $y = g(x)$ is shown on the top right. The transformed distribution $p(y)$ is shown in the top left. A linear function induces a Gaussian distribution, but a non-linear function induces a complex distribution. The solid line is the best Gaussian approximation to this; the dotted line is the EKF approximation to this. Source: Figure 3.4 of (Thrun et al. 2006). Used with kind permission of Sebastian

$$p(\mathbf{y}_t|\mathbf{z}_t) \approx \mathcal{N}(\mathbf{y}_t|\mathbf{h}(\boldsymbol{\mu}_{t|t-1}) + \mathbf{H}_t(\mathbf{y}_t - \boldsymbol{\mu}_{t|t-1}), \mathbf{R}_t) \tag{18.82}$$

where $\mathbf{H}_t$ is the Jacobian matrix of $\mathbf{h}$ evaluated at the prior mode:

$$H_{ij} \triangleq \frac{\partial h_i(\mathbf{z})}{\partial z_j} \tag{18.83}$$

$$\mathbf{H}_t \triangleq \mathbf{H}|_{\mathbf{z}=\boldsymbol{\mu}_{t|t-1}} \tag{18.84}$$

$$p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{u}_t) \approx \mathcal{N}(\mathbf{z}_t|\mathbf{g}(\mathbf{u}_t, \boldsymbol{\mu}_{t-1}) + \mathbf{G}_t(\mathbf{z}_{t-1} - \boldsymbol{\mu}_{t-1}), \mathbf{Q}_t) \tag{18.85}$$

where

$$G_{ij}(\mathbf{u}) \triangleq \frac{\partial g_i(\mathbf{u}, \mathbf{z})}{\partial z_j} \tag{18.86}$$

$$\mathbf{G}_t \triangleq \mathbf{G}(\mathbf{u}_t)|_{\mathbf{z}=\boldsymbol{\mu}_{t-1}} \tag{18.87}$$

Given this, we can then apply the Kalman filter to compute the posterior as follows:

$$\boldsymbol{\mu}_{t|t-1} = \mathbf{g}(\mathbf{u}_t, \boldsymbol{\mu}_{t-1}) \tag{18.88}$$

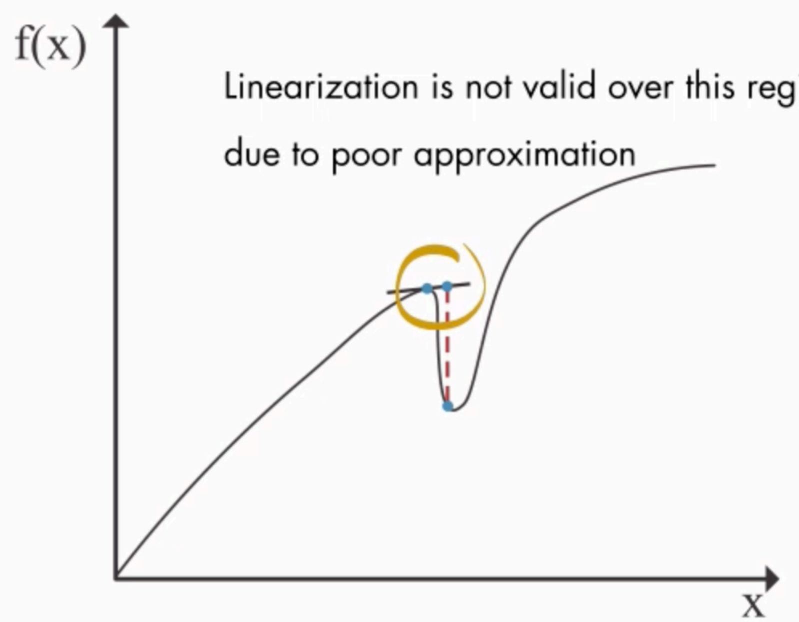
$$\mathbf{V}_{t|t-1} = \mathbf{G}_t\mathbf{V}_{t-1}\mathbf{G}_t^T + \mathbf{Q}_t \tag{18.89}$$

$$\mathbf{K}_t = \mathbf{V}_{t|t-1}\mathbf{H}_t^T(\mathbf{H}_t\mathbf{V}_{t|t-1}\mathbf{H}_t^T + \mathbf{R}_t)^{-1} \tag{18.90}$$

$$\boldsymbol{\mu}_t = \boldsymbol{\mu}_{t|t-1} + \mathbf{K}_t(\mathbf{y}_t - \mathbf{h}(\boldsymbol{\mu}_{t|t-1})) \tag{18.91}$$

$$\mathbf{V}_t = (\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)\mathbf{V}_{t|t-1} \tag{18.92}$$

- 잘 작동하지 않는 경우
  - prior covariancerk 너무 펑퍼짐할 때, 초기부터 많은 probability mass를 linearized 과정 중에 유실
  - mapping function이 nonlinear near the current mean 일 때

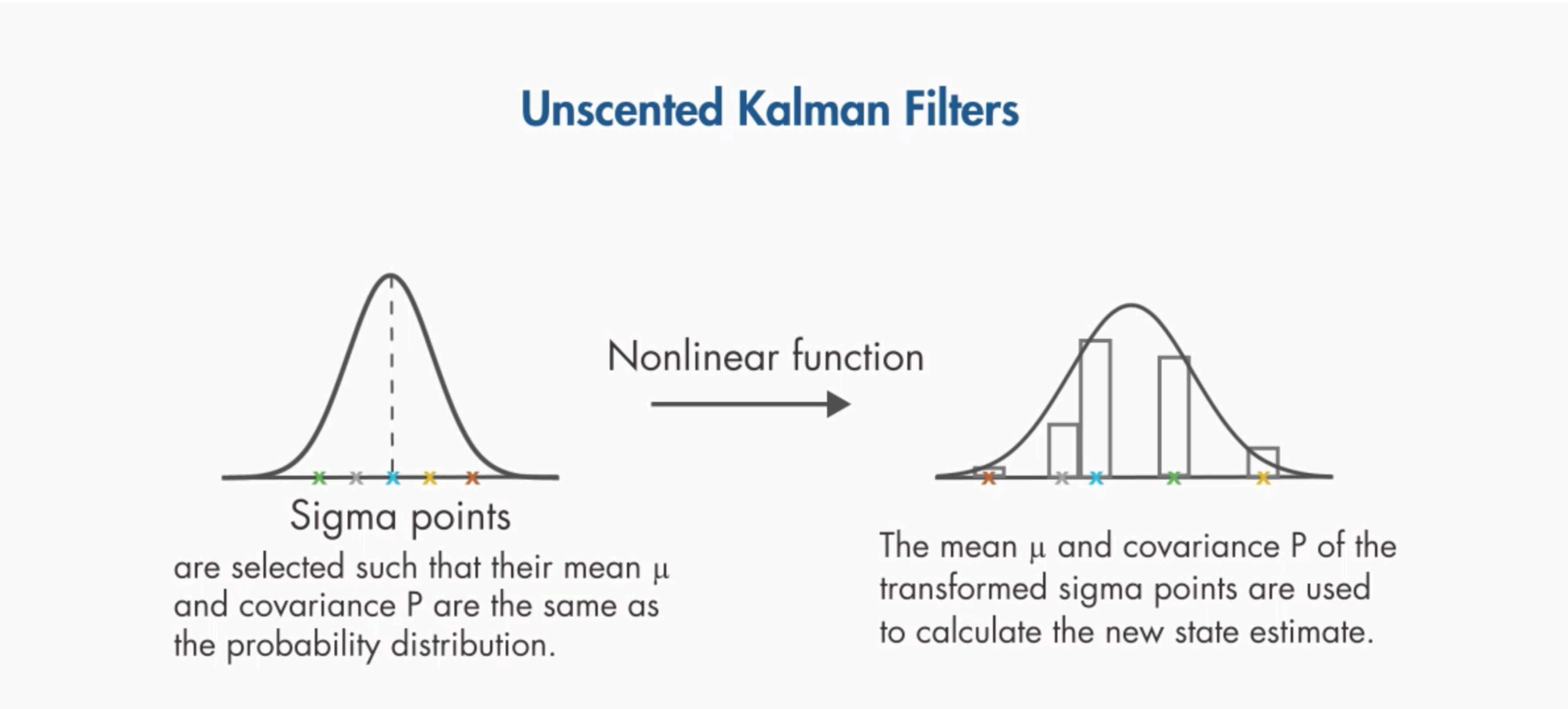


18.5.2. unscented kalman filter(UKF)

- better version of EKF
- approximate a Gaussian than to approximate a function
  - EKF : approximate a function as linear, then pass a gaussian through it
  - UKF : choose sigma points, and pass through function, and fit a gaussian to the transformed points

Unscented transform

- assume
  - p(x) is gassuian
  - y = f(x) as non-linear
- steps
  - choose 2d + 1 sigma points in x-domain such that it represent well underlying dist. i.e gaussian
  - pass through non-linear function
  - new mean and convariance are computed on transformed points



$$\mathbf{x} = \left(\boldsymbol{\mu}, \{\boldsymbol{\mu} + (\sqrt{(d+\lambda)\boldsymbol{\Sigma}})_{:i}\}_{i=1}^{d}, \{\boldsymbol{\mu} - (\sqrt{(d+\lambda)\boldsymbol{\Sigma}})_{:i}\}_{i=1}^{d}\right) \tag{18.93}$$

$$\boldsymbol{\mu}_y = \sum_{i=0}^{2d} w_m^i \mathbf{y}_i \tag{18.94}$$

$$\boldsymbol{\Sigma}_y = \sum_{i=0}^{2d} w_c^i (\mathbf{y}_i - \boldsymbol{\mu}_y)(\mathbf{y}_i - \boldsymbol{\mu}_y)^T \tag{18.95}$$

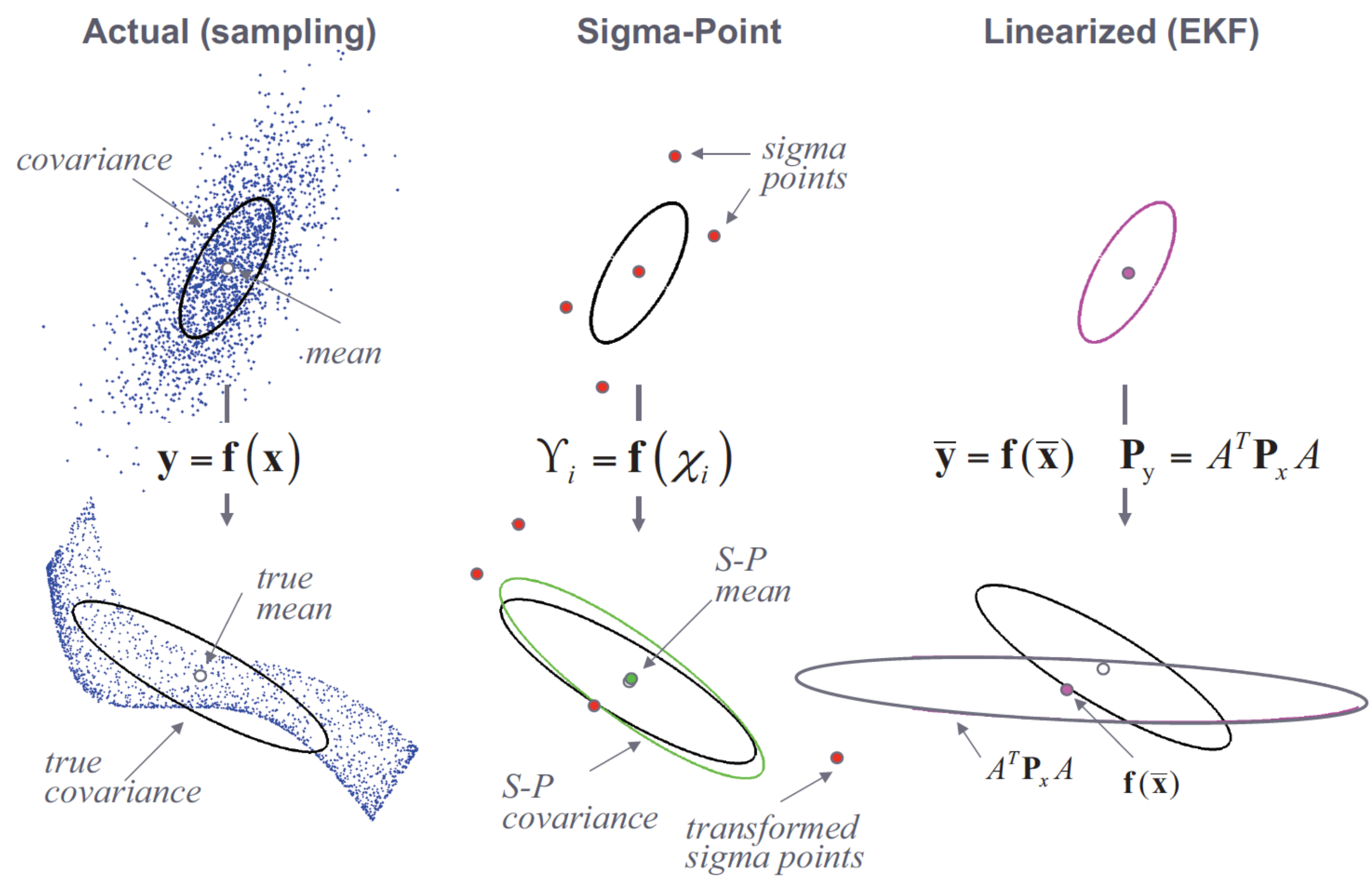


Figure 18.10 An example of the unscented transform in two dimensions. Source: (Wan and der Merwe 2001). Used with kind permission of Eric Wan.

unscented kalman filter

- aprroximate two non-gassuaian dist by two unscented transforms
- first is prediction step

The first step is to approximate the predictive density $p(\mathbf{z}_t|\mathbf{y}_{1:t-1}, \mathbf{u}_{1:t}) \approx \mathcal{N}(\mathbf{z}_t|\overline{\boldsymbol{\mu}}_t, \overline{\boldsymbol{\Sigma}}_t)$ by passing the old belief state $\mathcal{N}(\mathbf{z}_{t-1}|\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1})$ through the system model $\mathbf{g}$ as follows:

$$\mathbf{z}^0_{t-1} = \left(\boldsymbol{\mu}_{t-1}, \{\boldsymbol{\mu}_{t-1} + \gamma(\sqrt{\boldsymbol{\Sigma}_{t-1}})_{:i}\}_{i=1}^d, \{\boldsymbol{\mu}_{t-1} - \gamma(\sqrt{\boldsymbol{\Sigma}_{t-1}})_{:i}\}_{i=1}^d\right) \quad (18.99)$$

$$\overline{\mathbf{z}}^{*i}_t = \mathbf{g}(\mathbf{u}_t, \mathbf{z}^{0i}_{t-1}) \quad (18.100)$$

$$\overline{\boldsymbol{\mu}}_t = \sum_{i=0}^{2d} w^i_m \overline{\mathbf{z}}^{*i}_t \quad (18.101)$$

- second is approximate local evidence

The second step is to approximate the likelihood $p(\mathbf{y}_t|\mathbf{z}_t) \approx \mathcal{N}(\mathbf{y}_t|\hat{\mathbf{y}}_t, \mathbf{S}_t)$ by passing the prior $\mathcal{N}(\mathbf{z}_t|\overline{\boldsymbol{\mu}}_t, \overline{\boldsymbol{\Sigma}}_t)$ through the observation model $\mathbf{h}$:

$$\overline{\mathbf{z}}^0_t = \left(\overline{\boldsymbol{\mu}}_t, \{\overline{\boldsymbol{\mu}}_t + \gamma(\sqrt{\overline{\boldsymbol{\Sigma}}_t})_{:i}\}_{i=1}^d, \{\overline{\boldsymbol{\mu}}_t - \gamma(\sqrt{\overline{\boldsymbol{\Sigma}}_t})_{:i}\}_{i=1}^d\right) \quad (18.103)$$

$$\overline{\mathbf{y}}^{*i}_t = \mathbf{h}(\overline{\mathbf{z}}^{0i}_t) \quad (18.104)$$

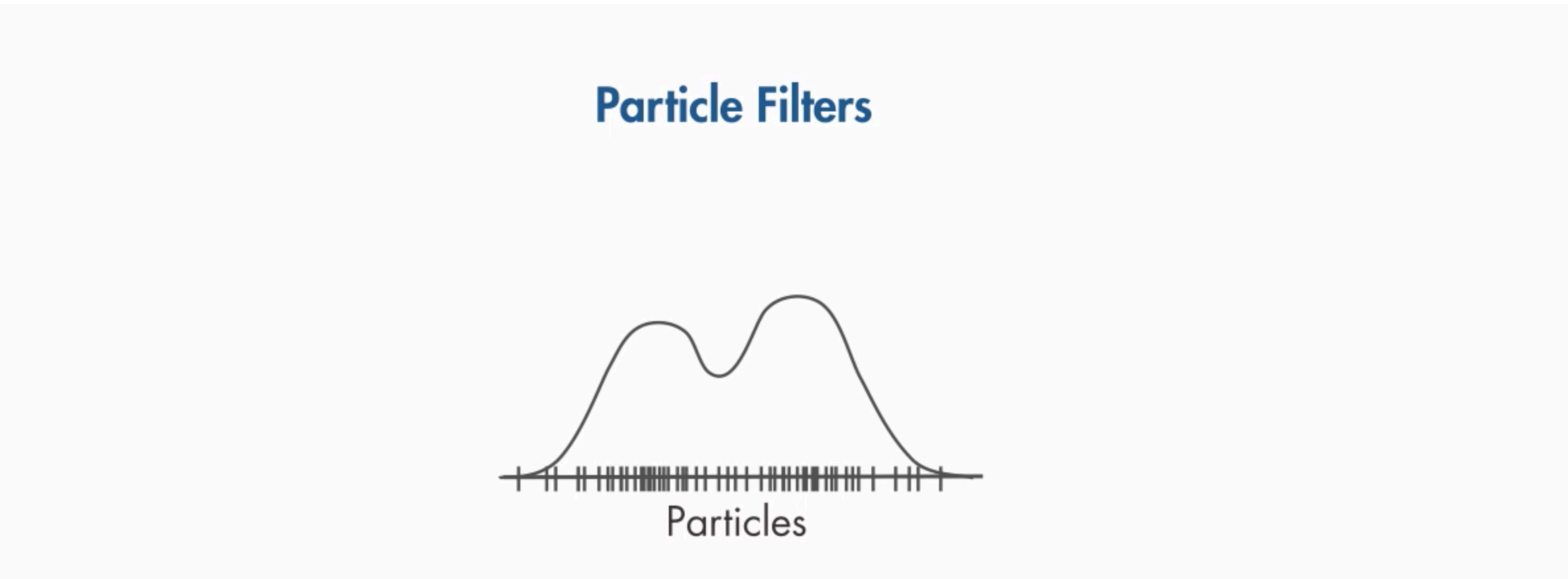
$$\hat{\mathbf{y}}_t = \sum_{i=0}^{2d} w^i_m \overline{\mathbf{y}}^{*i}_t \quad (18.105)$$

- last is update (error-based correction like KF)

Finally, we use Bayes rule for Gaussians to get the posterior $p(\mathbf{z}_t|\mathbf{y}_{1:t}, \mathbf{u}_{1:t}) \approx \mathcal{N}(\mathbf{z}_t|\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$:

$$
\begin{aligned}
\overline{\boldsymbol{\Sigma}}_t^{z,y} &= \sum_{i=0}^{2d} w_c^i (\overline{\mathbf{z}}_t^{*i} - \overline{\boldsymbol{\mu}}_t)(\overline{\mathbf{y}}_t^{*i} - \hat{\mathbf{y}}_t)^T && (18.107) \\
\mathbf{K}_t &= \overline{\boldsymbol{\Sigma}}_t^{z,y} \mathbf{S}_t^{-1} && (18.108) \\
\boldsymbol{\mu}_t &= \overline{\boldsymbol{\mu}}_t + \mathbf{K}_t(\mathbf{y}_t - \hat{\mathbf{y}}_t) && (18.109) \\
\boldsymbol{\Sigma}_t &= \overline{\boldsymbol{\Sigma}}_t - \mathbf{K}_t \mathbf{S}_t \mathbf{K}_t^T && (18.110)
\end{aligned}
$$

comparision of KF, EKF, UKF, Particle filter



State Estimator	Model	Assumed distribution	Computational cost
Kalman filter (KF)	Linear	Gaussian	Low
Extended Kalman filter (EKF)	Locally linear	Gaussian	Low (if the Jacobians need to be computed analytically) Medium (if the Jacobians can be computed numerically)
Unscented Kalman filter (UKF)	Nonlinear	Gaussian	Medium
Particle filter (PF)	Nonlinear	Non-Gaussian	High

18.5.3. Assumed density filtering (ADF)

18.6 Hybrid discrete/continuous SSMs

- hidden state가 discrete와 continous가 mixing 되어 있는 경우
- HMM + LG-SSM => switching lineary dynamic system

$$
\begin{aligned}
p(q_t = k | q_{t-1} = j, \boldsymbol{\theta}) &= A_{ij} & (18.131) \\
p(\mathbf{z}_t | \mathbf{z}_{t-1}, q_t = k, \mathbf{u}_t, \boldsymbol{\theta}) &= \mathcal{N}(\mathbf{z}_t | \mathbf{A}_k \mathbf{z}_{t-1} + \mathbf{B}_k \mathbf{u}_t, \mathbf{Q}_k) & (18.132) \\
p(\mathbf{y}_t | \mathbf{z}_t, q_t = k, \mathbf{u}_t, \boldsymbol{\theta}) &= \mathcal{N}(\mathbf{y}_t | \mathbf{C}_k \mathbf{z}_t + \mathbf{D}_k \mathbf{u}_t, \mathbf{R}_k) & (18.133)
\end{aligned}
$$

Inference

- unfortunetely, is is intractable
- exponential explosion of mode
    - prior (2 mixG) -> 4 mixG -> 8 mixG

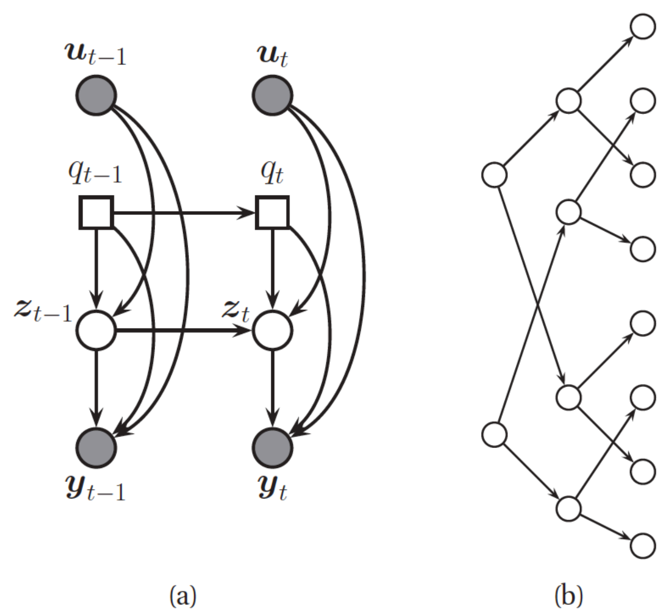
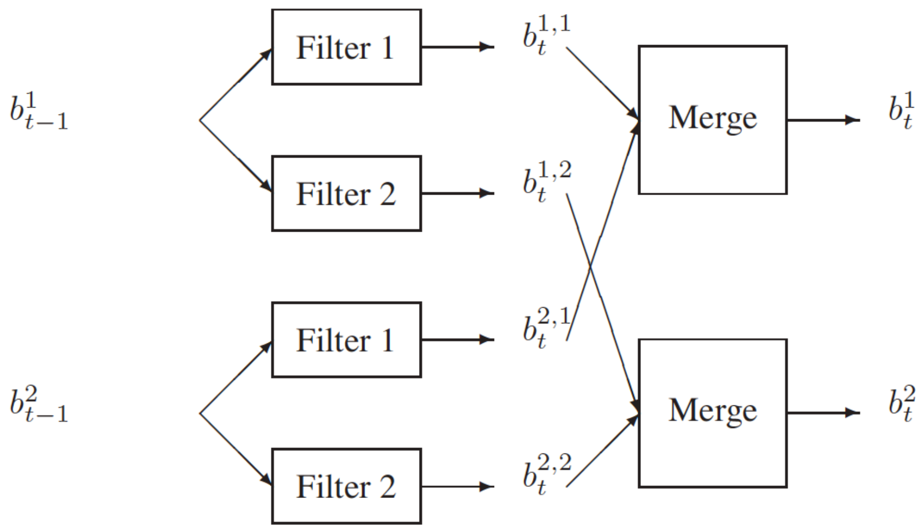


Figure 18.12 A switching linear dynamical system. (a) Squares represent discrete nodes, circles represent continuous nodes. (b) Illustration of how the number of modes in the belief state grows exponentially over time. We assume there are two binary states.

- how to approximate
    - prune low prob of trajectories in discrete-tree
    - sample in tree (section 23.6)
    - Use ADF : appроimate large mixG with small mixG

A gaussian sum filter for switching SSMs

- approximate the belief state at each step by mixture of gaussians
- implemented by running K kalman filters
- algorithm
    - given K mixG
    - pass through K different KF, and acquire K^2 beliefs (explode)
    - collapse K mixG into single gaussian



(a)

- how to merge mixture of gassian into single gaussian
  - by moment matching (only match first/second moment)
  - weak marginalization

The optimal way to approximate a mixture of Gaussians with a single Gaussian is given by $q = \arg\min_q \mathbb{KL}(q||p)$, where $p(\mathbf{z}) = \sum_k \pi_k \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ and $q(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$. This can be solved by moment matching, that is,

$$\boldsymbol{\mu} = \mathbb{E}[\mathbf{z}] = \sum_k \pi_k \boldsymbol{\mu}_k \tag{18.137}$$

$$\boldsymbol{\Sigma} = \text{cov}[\mathbf{z}] = \sum_k \pi_k \left(\boldsymbol{\Sigma}_k + (\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^T\right) \tag{18.138}$$

18.6.2 Application: data association and multi-target tracking

- 레이더 상의 K개의 물체를 K'개의 detect event를 통해 tracking
- K > K' 인 경우 : occlusion or missed detected
- K < K' 인 경우 : due to clutter or false alarms
- data association 문제
  - 서로 다른 cardinality의 z 와 y를 상호 연관시키는 문제

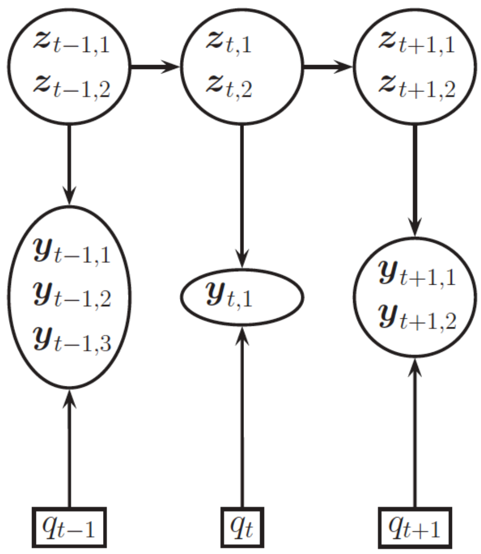
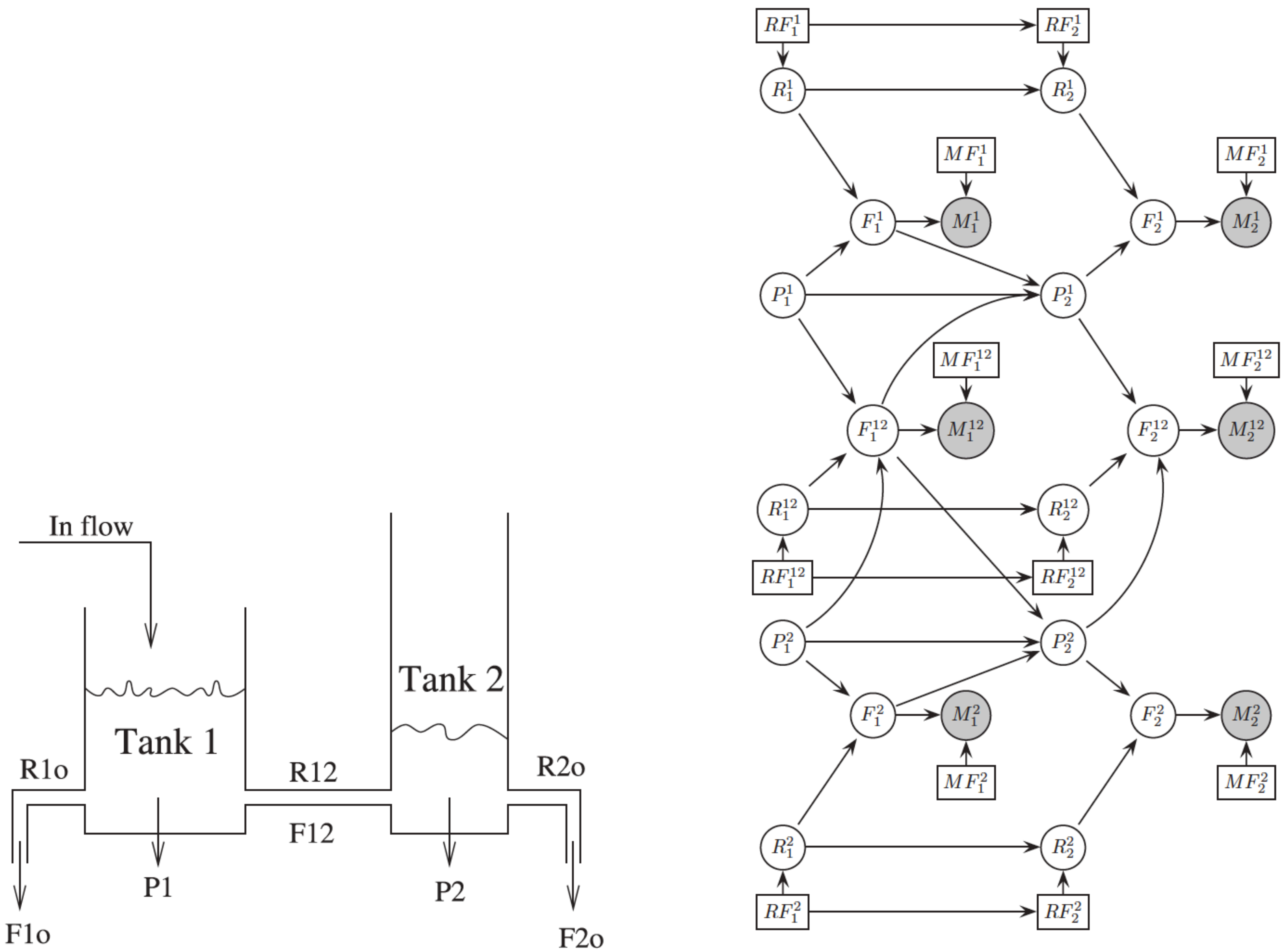


Figure 18.14 A model for tracking two objects in the presence of data-assocation ambiguity. We observe 3, 1 and 2 detections in the first three time steps.

- Hungarian alforithm
  - https://youtu.be/dQDZNHwuuOY (https://youtu.be/dQDZNHwuuOY)
  - typically, association reprsent K x K' matrix
  - add dummy => N x N, where N = max(K,K')
  - explain all false alarms, missed detections
  - compute bypartite matching, O(N^3)
- KF update does not work on dummy observations

18.6.3 Application: Falut diagnosis

- two-tank model
- benchmark in falut-diagnosis community
- latent
  - pressure inside tank(continuous)
  - resistence in pipe (continous)
  - whether or not of resistence failures (discrete)
- measurements
  - flow in-out
  - whether or not of measurement failures (discrete)



(a) (b)

In [ ]:

```

```