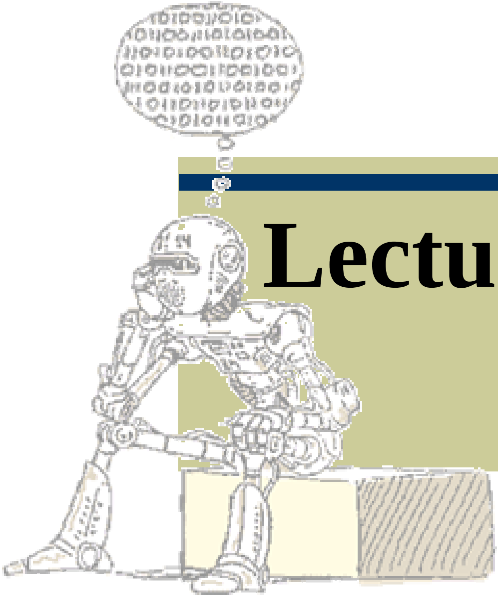


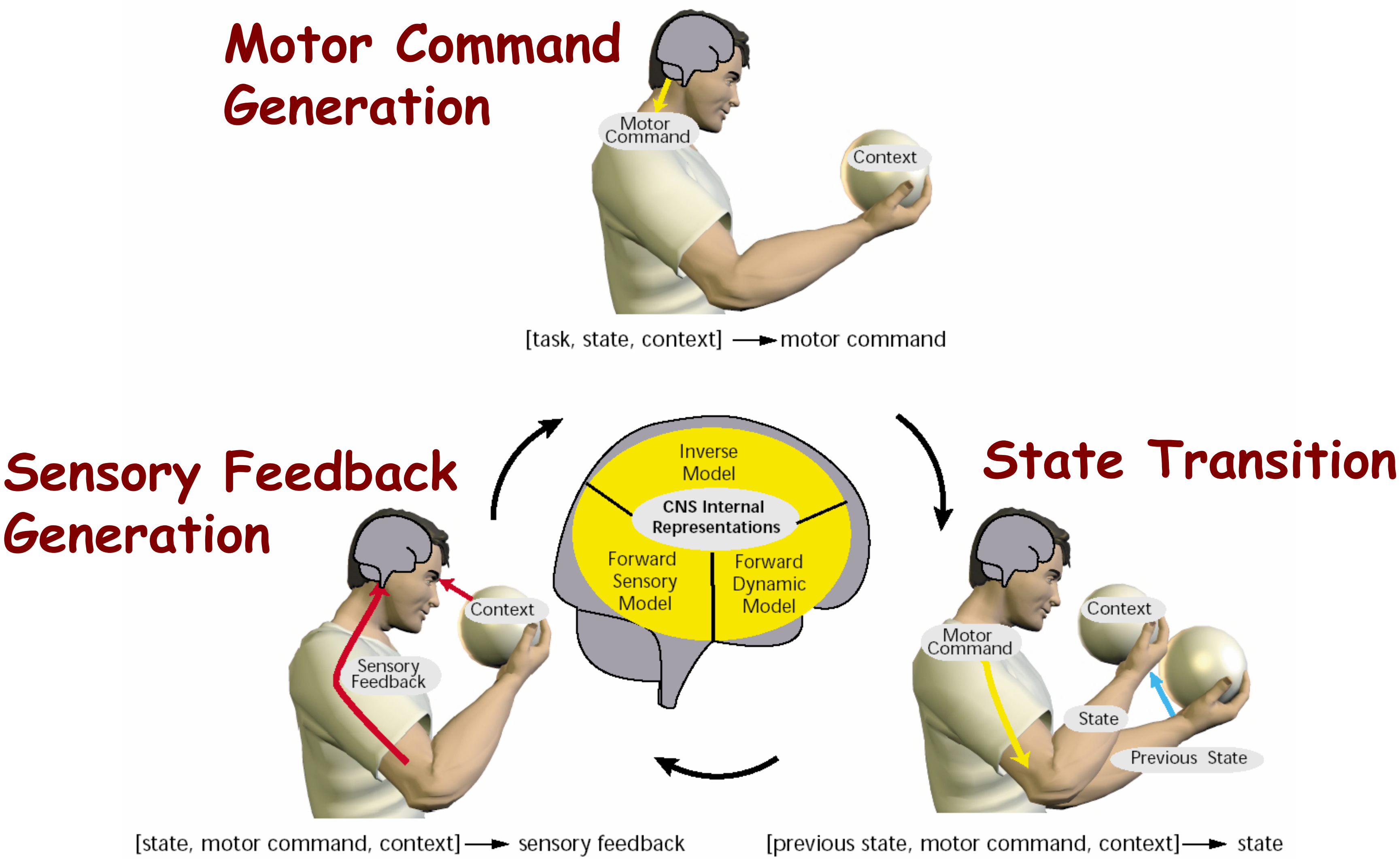
# Lecture XV

## Predictive Control

## Contents:

- State Estimation & Prediction
  - Kalman Filters
  - Extended Kalman Filters
  - Particle Filters

# Internal Models: Various Types



# State Estimation & Prediction

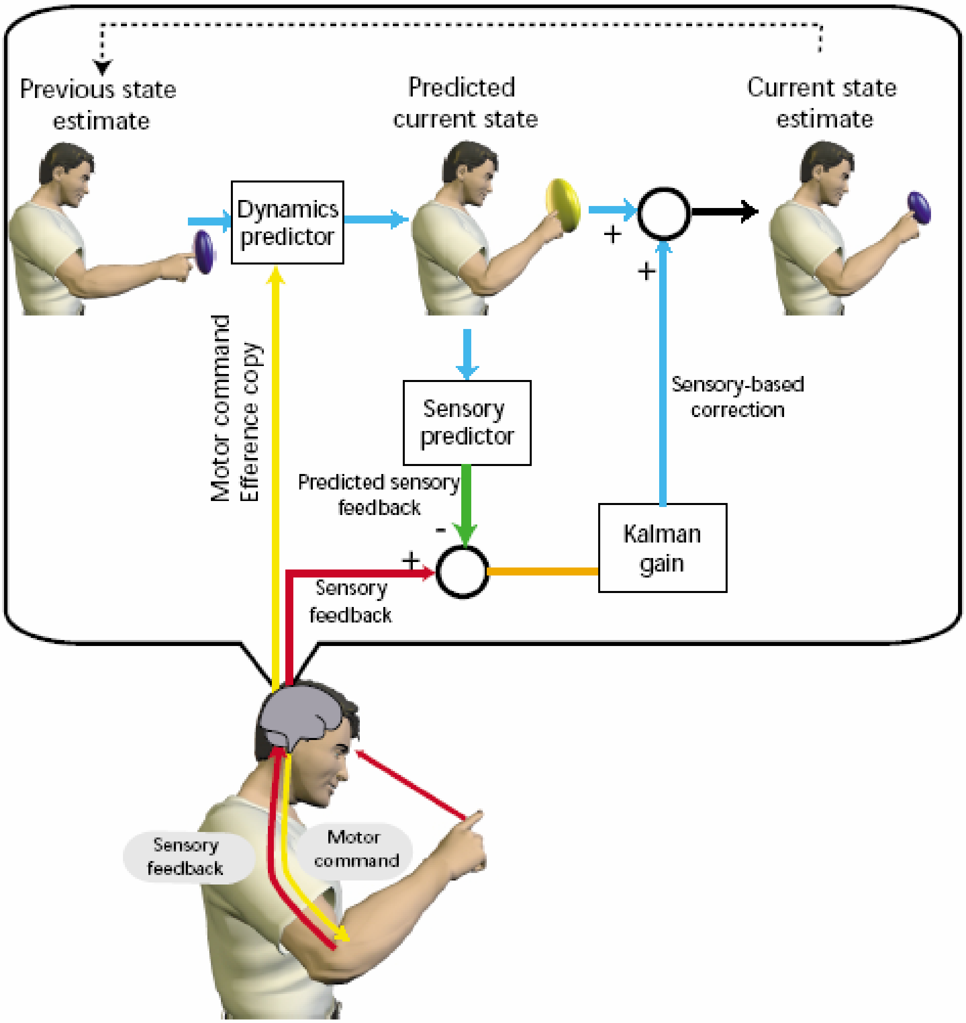
- **All the Internal models require estimation of the current state of the system (which is problematic because …):**
  - The CNS has to deal with huge sensorimotor delays
  - The sensory signals are contaminated by noise and may provide only partial information about the state
- **The Tennis Ball Example**

  Assume we just hit a tennis ball. If we use just the retinal position of the ball, we have to deal with a sensory delay of about 100ms → ***Use a forward model to predict where the ball is to generate a better estimate.***

  The ball's spin cannot be directly observed but can be calculated by integrating sensory information over time (ball's trajectory) → ***This estimate can be improved by knowing how the ball was hit (motor commands) and the internal model of the ball's dynamics.***

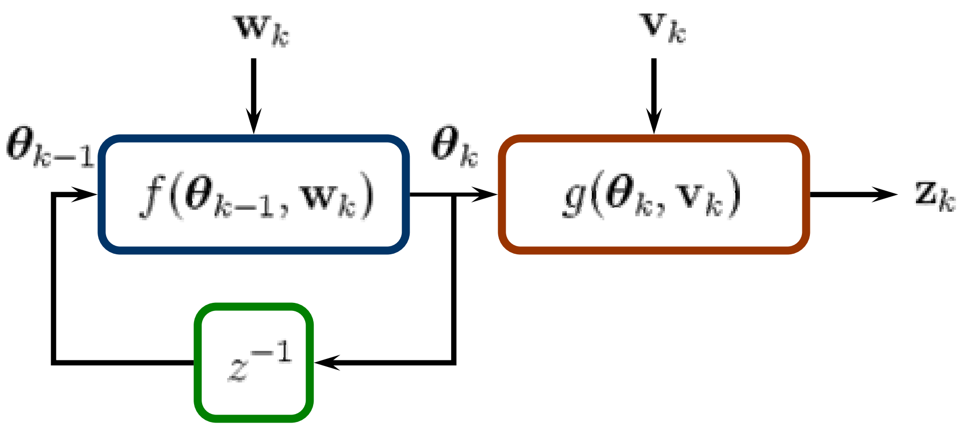
**This combination of using sensory feedback and forward models is known as the 'observer' : Kalman filters are an example.**

# Predictive Control




- An efference copy of the ***motor command*** and ***dynamics model*** enables us to construct the current state distribution from the previous state estimate.
- This new estimate is refined by using it to predict the current sensory feedback. The error between the prediction and the ***actual sensory feedback*** is used to correct the state estimate.
- The Kalman gain changes this ***sensory error into state error*** and also determines the ***relative importance*** placed on the efference motor copy vs. the sensory feedback

# Probabilistic State Estimation




State/observation equations

$$\boldsymbol{\theta}_k = f\left(\boldsymbol{\theta}_{k-1}, \mathbf{w}_k\right)$$
$$\mathbf{z}_k = g\left(\boldsymbol{\theta}_k, \mathbf{v}_k\right)$$

Goal: Given $\mathcal{D}_k = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k\}$ estimate $p(\boldsymbol{\theta}_k | \mathcal{D}_k)$

Recursive state estimation:

Initial: $p(\boldsymbol{\theta}_0 | \mathcal{D}_0) = p(\boldsymbol{\theta}_0)$

Prior: $p(\boldsymbol{\theta}_k | \mathcal{D}_{k-1}) = \int p(\boldsymbol{\theta}_k | \boldsymbol{\theta}_{k-1}) p(\boldsymbol{\theta}_{k-1} | \mathcal{D}_{k-1}) d\boldsymbol{\theta}_{k-1}$

Posterior: $p(\boldsymbol{\theta}_k | \mathcal{D}_k) \propto p(\mathbf{z}_k | \boldsymbol{\theta}_k) p(\boldsymbol{\theta}_k | \mathcal{D}_{k-1})$

# A Special Case: The Kalman Filter

Assume *linear* state/observation equations, and *Gaussian* noise distributions

State/observation equations:

$$\boldsymbol{\theta}_k = \mathbf{F}\boldsymbol{\theta}_{k-1} + \boldsymbol{\Phi}\mathbf{w}_k$$
$$\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$$
$$\mathbf{z}_k = \mathbf{G}\boldsymbol{\theta}_k + \boldsymbol{\Psi}\mathbf{v}_k$$
$$\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$$

Recursive state estimation:

Prior:

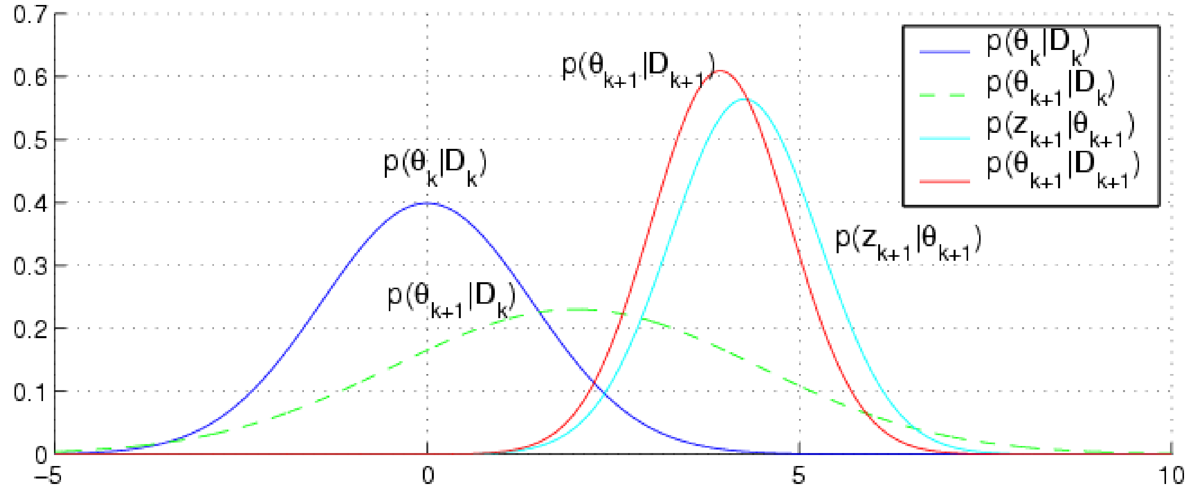
$$\mathbf{a}_{k|k-1} = \mathbf{F}\mathbf{a}_{k-1}$$
$$\mathbf{P}_{k|k-1} = \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^T + \boldsymbol{\Phi}\mathbf{Q}\boldsymbol{\Phi}^T$$

Posterior:

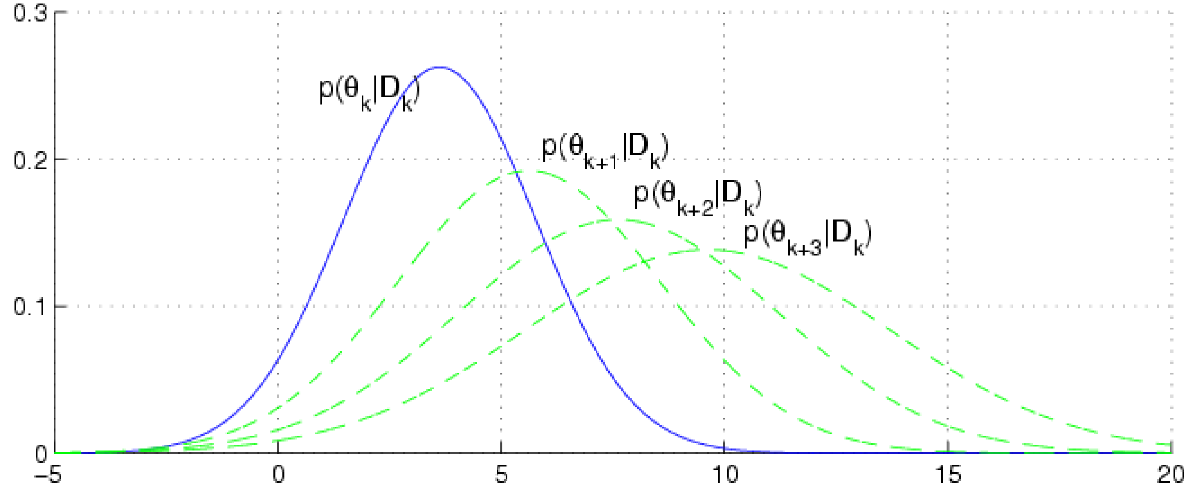
$$\mathbf{a}_k = \mathbf{a}_{k|k-1} + \mathbf{P}_{k|k-1}\mathbf{G}^T\mathbf{K}_k^{-1}\left(\mathbf{z}_k - \mathbf{G}\mathbf{a}_{k|k-1}\right)$$
$$\mathbf{P}_k = \mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\mathbf{G}^T\mathbf{K}_k^{-1}\mathbf{G}\mathbf{P}_{k|k-1}$$
$$\mathbf{K}_k = \mathbf{G}\mathbf{P}_{k|k-1}\mathbf{G}^T + \boldsymbol{\Psi}\mathbf{R}\boldsymbol{\Psi}^T$$

# Kalman Filter: Prediction / Estimation Cycle

Single prediction-estimation cycle




Multiple prediction cycles



# Stimulus Tracking - Formulation

State:

$$\boldsymbol{\theta} = \begin{bmatrix} x & y & dx & dy \end{bmatrix}^T$$

State Transition:

$$x_k = x_{k-1} + dx_{k-1}$$
$$y_k = y_{k-1} + dy_{k-1}$$
$$dx_k = dx_{k-1}$$
$$dy_k = dy_{k-1}$$

For a linear, Gaussian system:

$$\boldsymbol{\theta}_k = \mathbf{F}\boldsymbol{\theta}_{k-1} + \boldsymbol{\Phi}\mathbf{w}_k$$
$$\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$$
$$\mathbf{z}_k = \mathbf{G}\boldsymbol{\theta}_k + \boldsymbol{\Psi}\mathbf{v}_k$$
$$\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$$

State transition matrix:

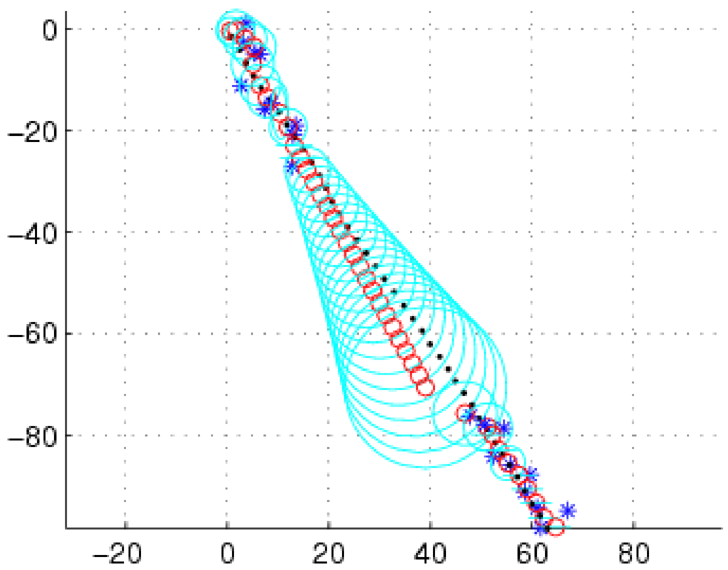
$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Observation matrix:

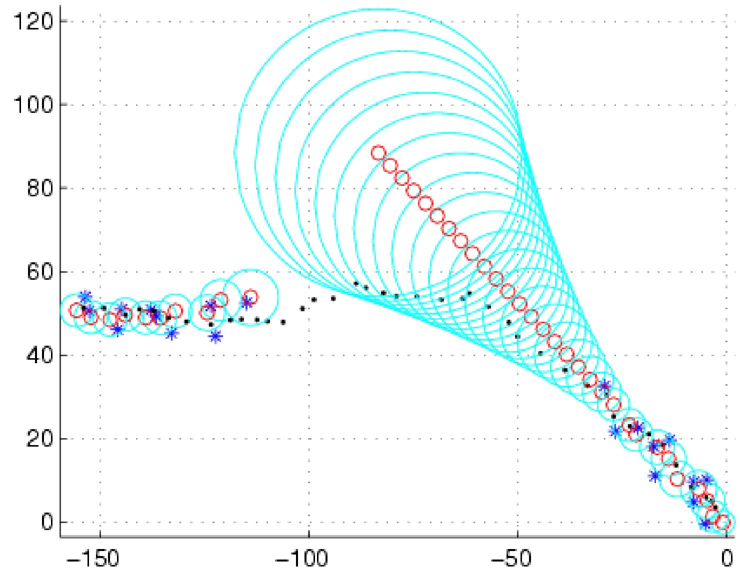
$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

# Stimulus Tracking - Examples

**Tracking a target with temporary occlusion**




Low deviation from linear system behavior




High deviation from linear system behavior

# Extended Kalman Filter

Linearize state equations:

$$\boldsymbol{\theta}_k = f(\mathbf{a}_{k-1}) + \mathbf{F}_{k-1}\left[\boldsymbol{\theta}_{k-1} - \mathbf{a}_{k-1}\right] + \boldsymbol{\Phi}_{k-1}\mathbf{w}_k$$

$$\mathbf{F}_{k-1} = \left.\frac{\partial f(\boldsymbol{\theta}, \mathbf{w})}{\partial \boldsymbol{\theta}}\right|_{(\mathbf{a}_{k-1}, \mathbf{0})}, \quad \boldsymbol{\Phi}_{k-1} = \left.\frac{\partial f(\boldsymbol{\theta}, \mathbf{w})}{\partial \mathbf{w}}\right|_{(\mathbf{a}_{k-1}, \mathbf{0})}$$

Linearize obs. equations:

$$\mathbf{z}_k = \mathbf{G}_k\boldsymbol{\theta}_k + \boldsymbol{\Psi}_k\mathbf{v}_k$$

$$\mathbf{G}_k = \left.\frac{\partial g(\boldsymbol{\theta}, \mathbf{v})}{\partial \boldsymbol{\theta}}\right|_{(\mathbf{a}_{k|k-1}, \mathbf{0})}, \quad \boldsymbol{\Psi}_k = \left.\frac{\partial g(\boldsymbol{\theta}, \mathbf{v})}{\partial \mathbf{v}}\right|_{(\mathbf{a}_{k|k-1}, \mathbf{0})}$$

Filter prediction-update:

$$\begin{aligned}
\mathbf{a}_{k|k-1} &= f(\mathbf{a}_{k-1}) \\
\mathbf{P}_{k|k-1} &= \mathbf{F}_{k-1}\mathbf{P}_{k-1}\mathbf{F}_{k-1}^T + \boldsymbol{\Phi}_{k-1}\mathbf{Q}\boldsymbol{\Phi}_{k-1}^T \\
\mathbf{a}_k &= \mathbf{a}_{k|k-1} + \mathbf{P}_{k|k-1}\mathbf{G}_k^T\mathbf{K}_k^{-1}\left[\mathbf{z}_k - g(\mathbf{a}_{k|k-1})\right] \\
\mathbf{P}_k &= \mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\mathbf{G}_k^T\mathbf{K}_k^{-1}\mathbf{G}_k\mathbf{P}_{k|k-1} \\
\mathbf{K}_k &= \mathbf{G}_k\mathbf{P}_{k|k-1}\mathbf{G}_k^T + \boldsymbol{\Psi}_k\mathbf{R}\boldsymbol{\Psi}_k^T
\end{aligned}$$

# Particle Filters

Assume N samples: $\boldsymbol{\theta}_{k-1}(i) \sim p(\boldsymbol{\theta}_{k-1}|\mathcal{D}_{k-1})$

Prediction:

$$\boldsymbol{\theta}_{k|k-1}(i) = f\left(\boldsymbol{\theta}_{k-1}(i), \mathbf{w}_{k-1}(i)\right) \text{ where } \mathbf{w}_{k-1}(i) \sim p(\mathbf{w}_{k-1})$$

Update:

- Compute: $q_i = \dfrac{p(\mathbf{z}_k|\boldsymbol{\theta}_{k|k-1}(i))}{\sum_{i=1}^{N} p(\mathbf{z}_k|\boldsymbol{\theta}_{k|k-1}(i))}$
- Resample (multinomial dist.) $\Pr\left(\boldsymbol{\theta}_k(j) = \boldsymbol{\theta}_{k|k-1}(i)\right) = q_i$

Algorithm requirements:

- We can sample from $p(\mathbf{w})$
- We can evaluate (up to a constant factor) $p(\mathbf{z}|\boldsymbol{\theta})$

Gordon et al (1993)

# Other variations on a theme…

- **Switching Kalman Filters**
- **Variational Kalman Filters**
  - Variational approximation to posterior distribution of filter parameters (Ghahramani & Beal – 1998)
- **Unscented Kalman Filters**
- **Unscented Particle Filters**