Robotics
Spring 2023

Abhishek Gupta

TAs: Yi Li, Srivatsa GS

Recap: Course Overview

- Filtering/Smoothing
- Localization
- Mapping
- SLAM

- Search
- Motion Planning
- TrajOpt
- Stability/Certification

- MDPs and RL
- Imitation Learning
- Solving POMDPs

Lecture Outline

Kalman Filtering

↓

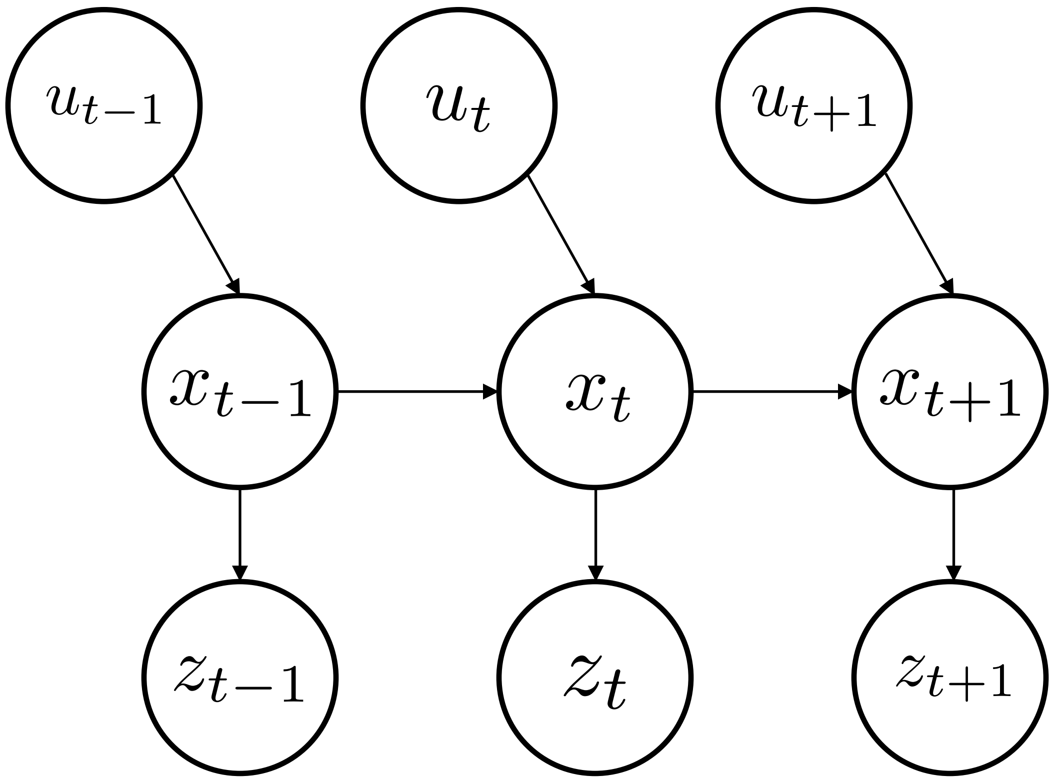
Extended Kalman Filter

↓

Unscented Kalman Filter

Discrete Kalman Filter

Kalman filter = Bayes filter with Linear Gaussian dynamics and sensor models



Discrete Kalman Filter

Estimates the state ***x*** of a discrete-time controlled process that is governed by the linear stochastic difference equation

$$x_{t+1} = Ax_t + Bu_t + \epsilon_t$$
$$\epsilon_t \sim \mathcal{N}(0, Q)$$

Linear Gaussian

with a measurement

$$z_{t+1} = Cx_{t+1} + \delta_t$$
$$\delta_t \sim \mathcal{N}(0, R)$$

Components of a Kalman Filter

A Matrix (n x n) that describes how the state evolves from ***t-1*** to ***t*** without controls or noise.

B Matrix (n x l) that describes how the control $\boldsymbol{u_{t-1}}$ changes the state from ***t-1*** to ***t***

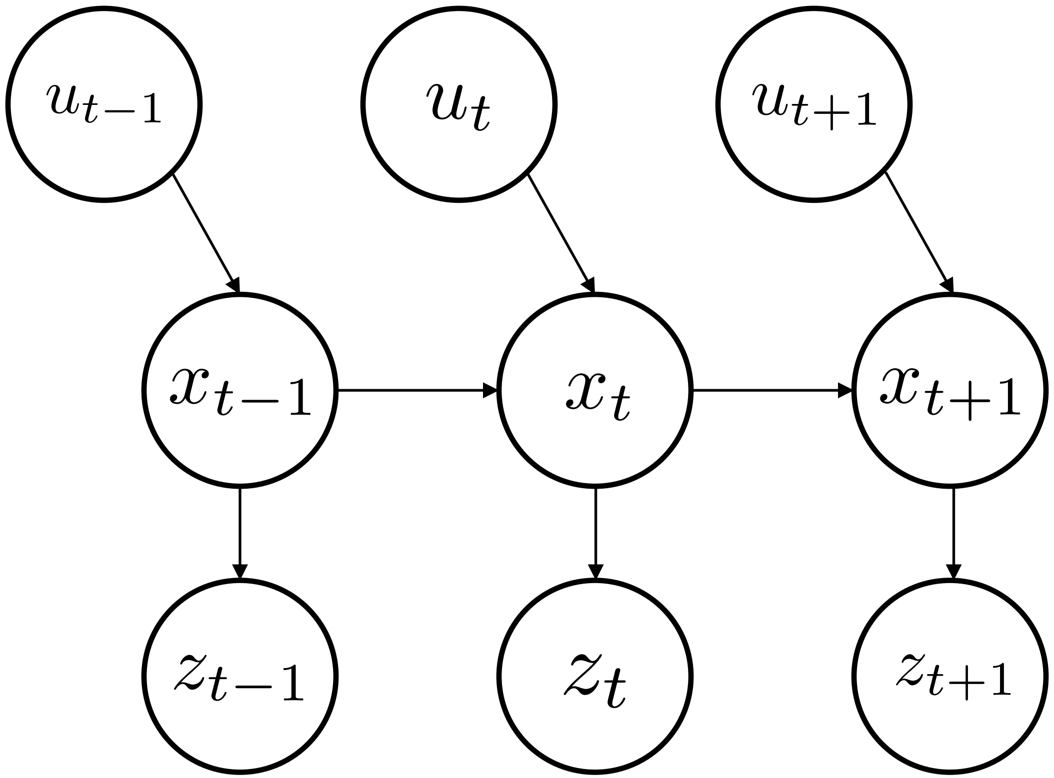
C Matrix (k x n) that describes how to map the state $\boldsymbol{x_t}$ to an observation $\boldsymbol{z_t}$.

ϵ_t

δ_t

Random variables representing the process and measurement noise that are assumed to be independent and normally distributed with covariance ***R*** and ***Q*** respectively.

Goal of the Kalman Filter



Belief

$$p(x_t | z_{0:t}, u_{0:t-1})$$

Idea: recursive update for Bayes filter

$$\propto p(z_t|x_t) \int p(x_t|x_{t-1}, u_{t-1}) p(x_{t-1}|z_{0:t-1}, u_{0:t-2}) dx_{t-1}$$

Measurement

Dynamics

Recursive Belief

2 step process:

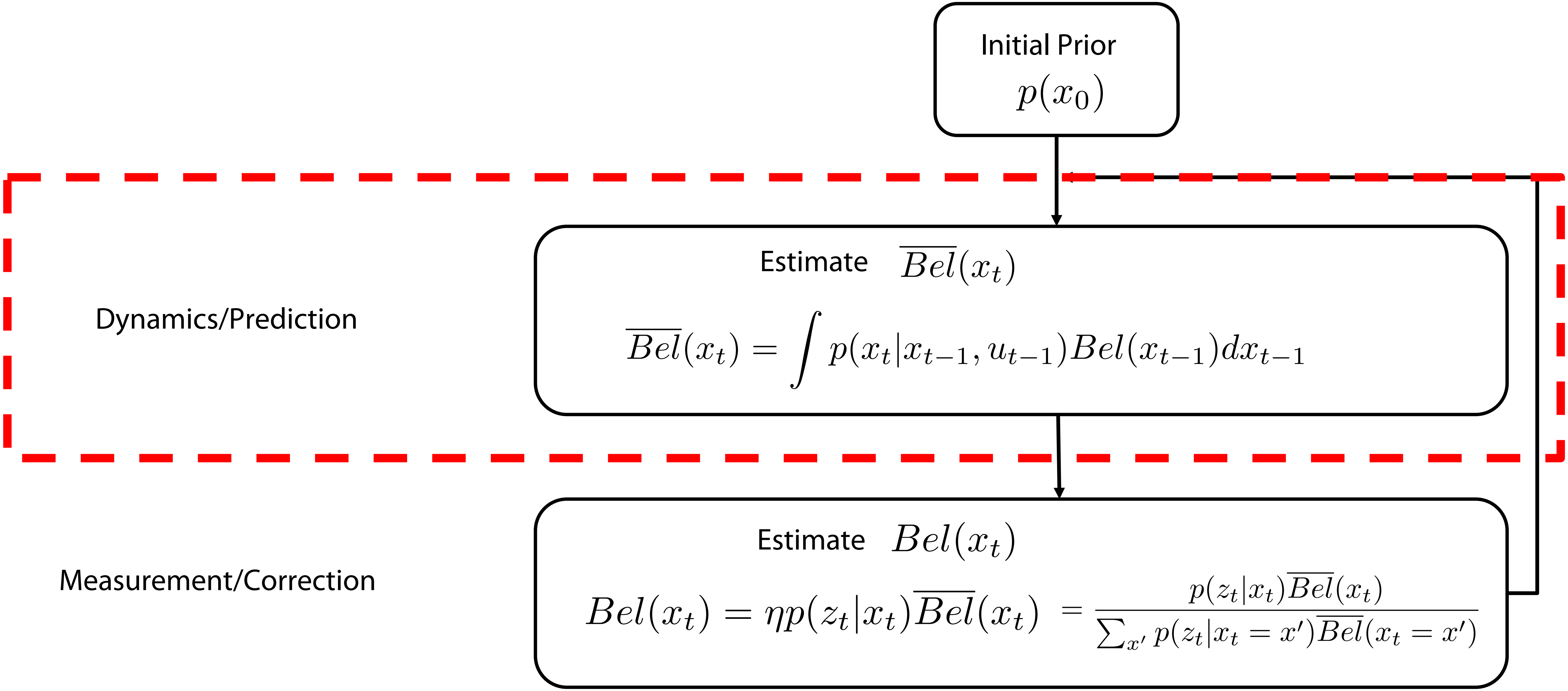
- Dynamics update (incorporate action)
- Measurement update (incorporate sensor reading)

Linear Gaussian Systems: Initialization

- Initial belief is normally distributed:

$$bel(x_0) = N\left(x_0; \mu_0, \Sigma_0\right)$$

Linear Gaussian Systems: Dynamics



Linear Gaussian Systems: Dynamics

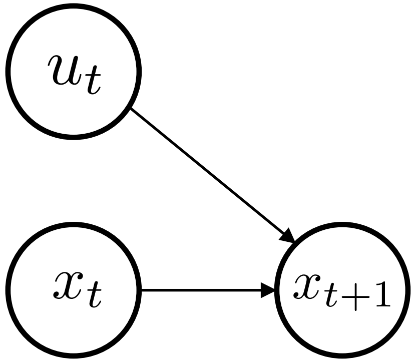
- Integrate the effect of one action under the dynamics, before measurement comes in

$$x_{t+1} = Ax_t + Bu_t + \epsilon_t \qquad \epsilon_t \sim \mathcal{N}(0, Q_t)$$

$$p(x_{t+1}|x_t, u_t) = \mathcal{N}(Ax_t + Bu_t, Q_t)$$

$$p(x_{t+1}|z_{0:t}, u_{0:t}) = \int p(x_t|u_{0:t-1}, z_{0:t})p(x_{t+1}|x_t, u_t)dx_t$$

Gaussian, easy!

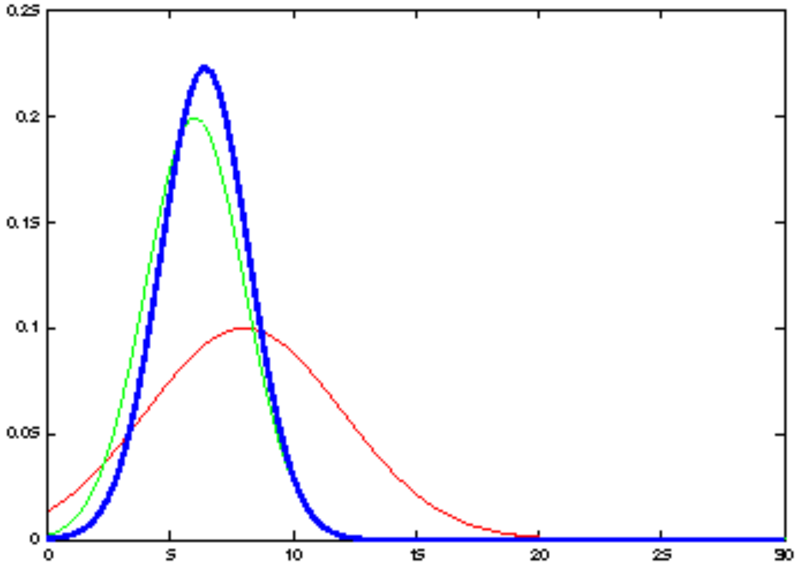


Linear Gaussian Systems: Dynamics Intuition

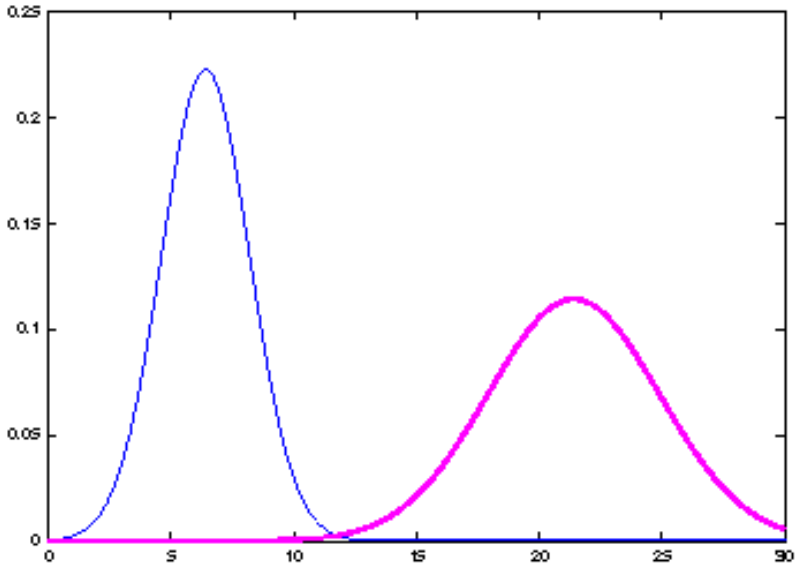
Previous belief $p(x_t|u_{0:t}, z_{0:t}) \sim \mathcal{N}(\mu_{t|0:t}, \Sigma_{t|0:t})$

Belief Update $p(x_{t+1}|z_{0:t}, u_{0:t+1}) \sim \mathcal{N}(A\mu_{t|0:t} + Bu_t, A\Sigma_{t|0:t}A^T + Q_t)$

Intuition: Scale and shift the mean according to dynamics, uncertainty grows quadratically!



Belief at x_t



Belief post dynamics → shifted mean, scaled and shifted variance

Linear Gaussian Systems: Dynamics

- Integrate the effect of one action under the dynamics, before measurement comes in

$$x_{t+1} = Ax_t + Bu_t + \epsilon_t \qquad \epsilon_t \sim \mathcal{N}(0, Q_t)$$

$$p(x_{t+1}|x_t, u_t) = \mathcal{N}(Ax_t + Bu_t, Q_t)$$

$$p(x_{t+1}|z_{0:t}, u_{0:t}) = \int p(x_t|u_{0:t-1}, z_{0:t})p(x_{t+1}|x_t, u_t)dx_t$$

Previous belief $p(x_t|u_{0:t-1}, z_{0:t}) \sim \mathcal{N}(\mu_{t|0:t}, \Sigma_{t|0:t})$

Belief Update $p(x_{t+1}|z_{0:t}, u_{0:t}) \sim \mathcal{N}(A\mu_{t|0:t} + Bu_t, A\Sigma_{t|0:t}A^T + Q_t)$

How??

Kalman Dynamics Update: Intuitive

- Integrate the effect of one action under the dynamics, before measurement comes in

$$x_{t+1} = Ax_t + Bu_t + \epsilon_t \qquad \epsilon_t \sim \mathcal{N}(0, Q_t)$$

$$p(x_{t+1}|x_t, u_t) = \mathcal{N}(Ax_t + Bu_t, Q_t)$$

$$p(x_{t+1}|z_{0:t}, u_{0:t}) \sim \mathcal{N}(A\mu_{t|0:t} + Bu_t, A\Sigma_{t|0:t}A^T + Q_t)$$

Kalman Dynamics Update: Full

- Integrate the effect of one action under the dynamics, before measurement comes in

$$p(x_{t+1}|z_{0:t}, u_{0:t}) = \int p(x_t|u_{0:t-1}, z_{0:t})p(x_{t+1}|x_t, u_t)dx_t$$

Stays in Gaussian world

$$(X_{t+1}, X_t)|_{z_{0:t}, u_{0:t}} \sim \mathcal{N}\left(\begin{bmatrix} \mu_{t|0:t} \\ \mu_{t+1|0:t} \end{bmatrix}, \begin{bmatrix} \Sigma_{t|0:t} & \Sigma_{t,t+1|0:t} \\ \Sigma_{t+1,t|0:t} & \Sigma_{t+1|0:t} \end{bmatrix} \right)$$

Current belief at time t

Now compute the mean and covariance and then marginalize

Kalman Dynamics Update: Full

$$(X_{t+1}, X_t)|_{z_{0:t}, u_{0:t}} \sim \mathcal{N}\left(\begin{bmatrix} \mu_{t|0:t} \\ \mu_{t+1|0:t} \end{bmatrix}, \begin{bmatrix} \Sigma_{t|0:t} & \Sigma_{t,t+1|0:t} \\ \Sigma_{t+1,t|0:t} & \Sigma_{t+1|0:t} \end{bmatrix} \right)$$

$$\mu_{t+1|0:t} = \mathbb{E}\left[X_{t+1} | z_{0:t}, u_{0:t}\right]$$

Mean

$$\Sigma_{t+1|0:t} = \mathbb{E}\left[(X_{t+1|0:t} - \mu_{t+1|0:t})(X_{t+1|0:t} - \mu_{t+1|0:t})^T\right]$$

Diagonal Covariance

$$\Sigma_{t,t+1|0:t} = \mathbb{E}\left[(X_{t|0:t} - \mu_{t|0:t})(X_{t+1|0:t} - \mu_{t+1|0:t})^T\right]$$

Cross Covariance

Kalman Dynamics Update: Full

$$(X_{t+1}, X_t)|_{z_{0:t}, u_{0:t}} \sim \mathcal{N}\left(\begin{bmatrix} \mu_{t|0:t} \\ \mu_{t+1|0:t} \end{bmatrix}, \begin{bmatrix} \Sigma_{t|0:t} & \Sigma_{t,t+1|0:t} \\ \Sigma_{t+1,t|0:t} & \Sigma_{t+1|0:t} \end{bmatrix} \right)$$

Mean

$$\begin{aligned} \mu_{t+1|0:t} &= \mathbb{E}\left[X_{t+1} | z_{0:t}, u_{0:t}\right] \\ &= \mathbb{E}\left[AX_t + Bu_t + \epsilon_t | z_{0:t}, u_{0:t}\right] \\ &= A\mathbb{E}\left[X_t | z_{0:t}, u_{0:t}\right] + Bu_t + \mathbb{E}\left[\epsilon_t | z_{0:t}, u_{0:t}\right] \\ &= A\mu_{t|0:t} + Bu_t \end{aligned}$$

Diagonal Covariance

$$\begin{aligned} \Sigma_{t+1|0:t} &= \mathbb{E}\left[(X_{t+1|0:t} - \mu_{t+1|0:t})(X_{t+1|0:t} - \mu_{t+1|0:t})^T\right] \\ &= \mathbb{E}\left[(AX_{t|0:t} + Bu_t + \epsilon_t - A\mu_{t|0:t} - Bu_t)(AX_{t|0:t} + Bu_t + \epsilon_t - A\mu_{t|0:t} - Bu_t)^T\right] \\ &= A\mathbb{E}\left[(X_{t|0:t} - \mu_{t|0:t})(X_{t|0:t} - \mu_{t|0:t})^T\right] A^T + Q_t \\ &= A\Sigma_{t|0:t}A^T + Q_t \end{aligned}$$

Cross Covariance

$$\Sigma_{t,t+1|0:t} = \mathbb{E}\left[(X_{t|0:t} - \mu_{t|0:t})(X_{t+1|0:t} - \mu_{t+1|0:t})^T\right]$$

$$\Sigma_{t,t+1|0:t} = \Sigma_{t|0:t}A^T$$

Linear Gaussian Systems: Dynamics

$$(X_{t+1}, X_t)|_{z_{0:t}, u_{0:t}} \sim \mathcal{N}\left(\begin{bmatrix} \mu_{t|0:t} \\ \mu_{t+1|0:t} \end{bmatrix}, \begin{bmatrix} \Sigma_{t|0:t} & \Sigma_{t,t+1|0:t} \\ \Sigma_{t+1,t|0:t} & \Sigma_{t+1|0:t} \end{bmatrix} \right)$$

Mean $$\mu_{t+1|0:t} = A\mu_{t|0:t} + Bu_t$$

Diagonal Covariance $$\Sigma_{t+1|0:t} = A\Sigma_{t|0:t}A^T + Q_t$$

Previous belief $$p(x_t|u_{0:t-1}, z_{0:t}) \sim \mathcal{N}(\mu_{t|0:t}, \Sigma_{t|0:t})$$

Belief Update $$p(x_{t+1}|z_{0:t}, u_{0:t}) \sim \mathcal{N}(A\mu_{t|0:t} + Bu_t, A\Sigma_{t|0:t}A^T + Q_t)$$

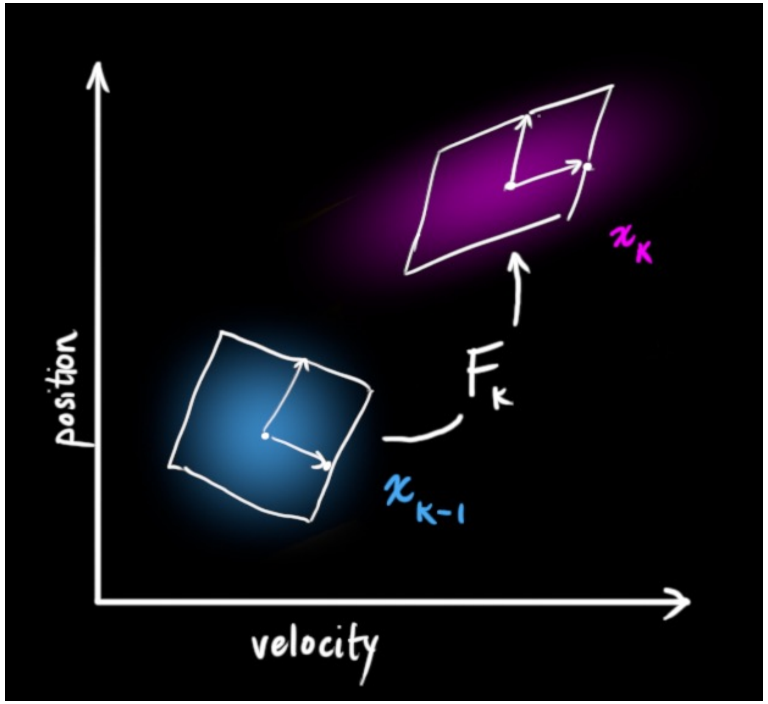
Intuition: Scale and shift the mean according to dynamics, uncertainty grows quadratically!

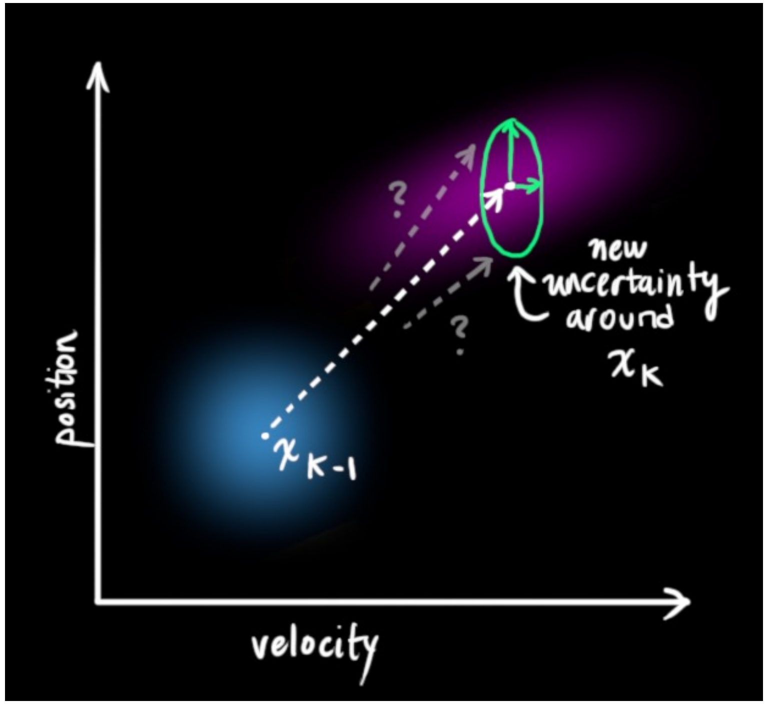
Linear Gaussian Systems: Dynamics

Previous belief $$p(x_t | u_{0:t}, z_{0:t}) \sim \mathcal{N}(\mu_{t|0:t}, \Sigma_{t|0:t})$$

Belief Update $$p(x_{t+1} | z_{0:t}, u_{0:t+1}) \sim \mathcal{N}(A\mu_{t|0:t} + Bu_t, A\Sigma_{t|0:t}A^T + Q_t)$$

Intuition: Scale and shift the mean according to dynamics, uncertainty grows!



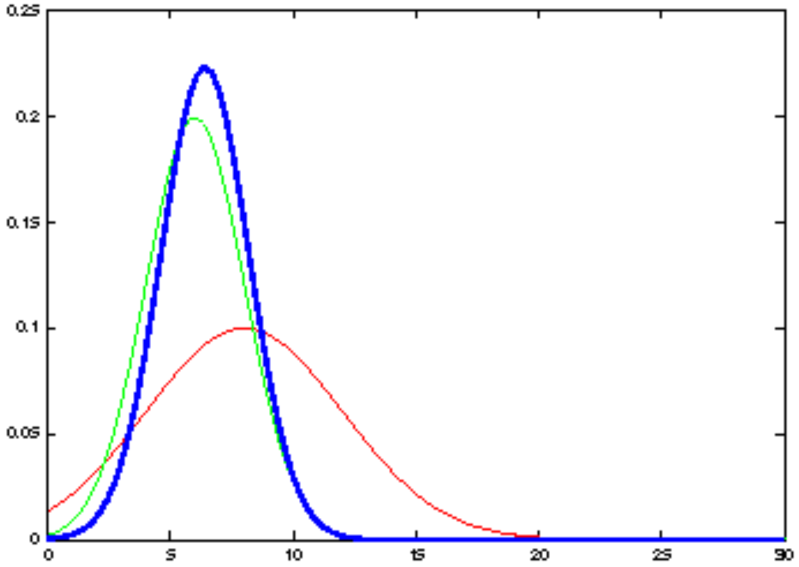


Intuition Behind Prediction Step

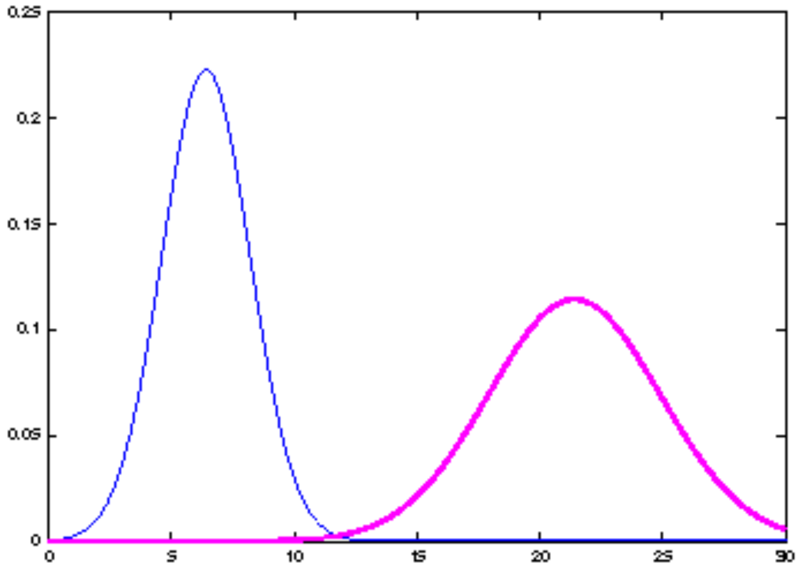
Previous belief $$p(x_t|u_{0:t}, z_{0:t}) \sim \mathcal{N}(\mu_{t|0:t}, \Sigma_{t|0:t})$$

Belief Update $$p(x_{t+1}|z_{0:t}, u_{0:t+1}) \sim \mathcal{N}(A\mu_{t|0:t} + Bu_t, A\Sigma_{t|0:t}A^T + Q_t)$$

Intuition: Scale and shift the mean according to dynamics, uncertainty grows!

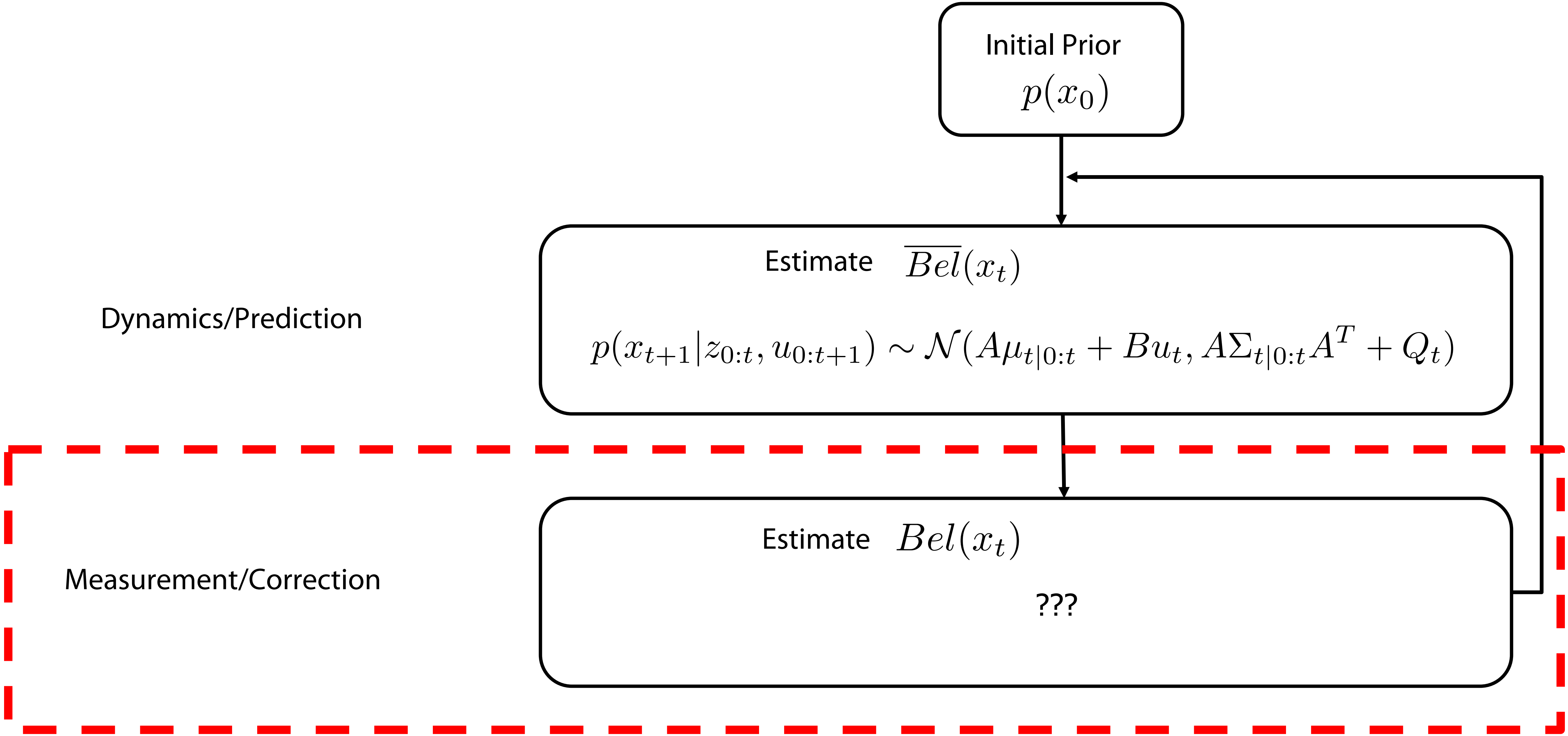


Belief at x_t



Belief post dynamics → shifted mean, scaled and shifted variance

Kalman Filter: Where are we?



Linear Gaussian Systems: Observations

Measurement/Correction

Estimate $Bel(x_t)$

???

$$p(x_{t+1}|z_{0:t+1}, u_{0:t}) = \eta\, p(z_t|x_t)p(x_{t+1}|z_{0:t}, u_{0:t})$$

Need to do conditioning/normalization with Linear Gaussians

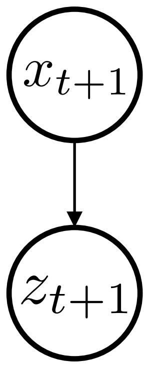
Linear Gaussian Systems: Observations

- Integrate the effect of an observation using sensor model, after dynamics

$$z_{t+1} = Cx_{t+1} + \delta_t \qquad \delta_t \sim \mathcal{N}(0, R_t)$$

$$p(z_{t+1}|x_{t+1}) = \mathcal{N}(Cx_{t+1}, R_t)$$

$$p(x_{t+1}|z_{0:t+1}, u_{0:t}) \propto p(z_{t+1}|x_{t+1})p(x_{t+1}|z_{0:t}, u_{0:t})$$

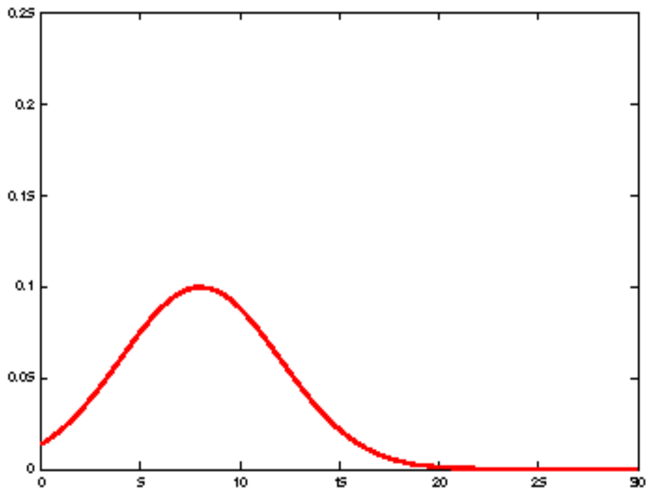


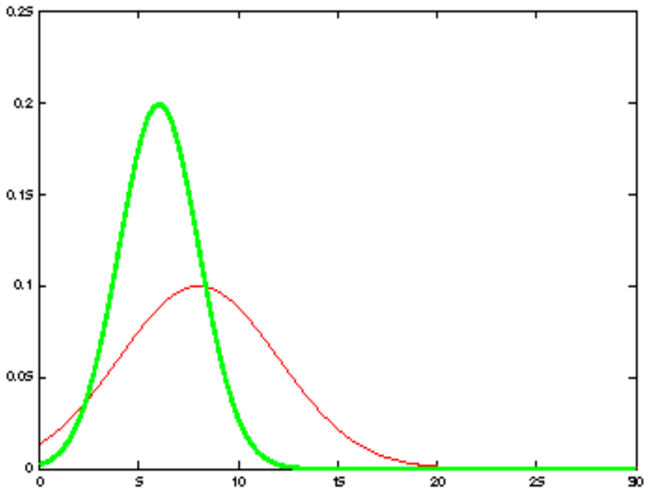
Gaussian, easy to normalize

Linear Gaussian Systems: Observations Intuition

Previous belief (post dynamics)

New Measurement





For the sake of simplicity, let's say C = I

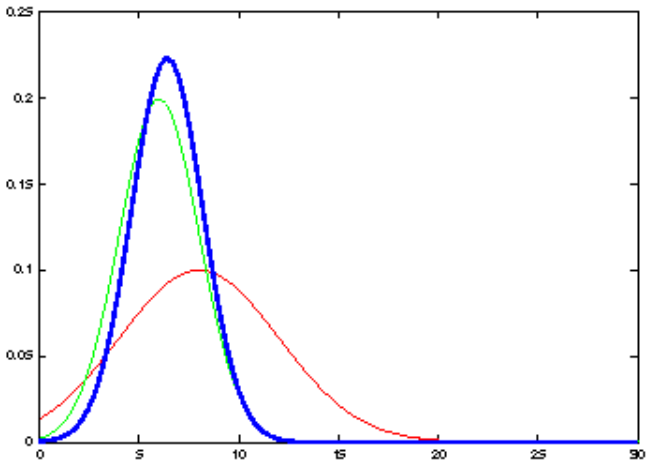
Previous belief

$$p(x_{t+1}|z_{0:t}, u_{0:t}) = \mathcal{N}(\mu_{t+1|0:t}, \Sigma_{t+1|0:t})$$

Updated belief

$$p(x_{t+1}|z_{0:t+1}, u_{0:t})$$

$$= \mathcal{N}(\mu_{t+1|0:t} + K_{t+1}(z_{t+1} - \mu_{t+1|0:t}), (I - K_{t+1})\Sigma_{t+1|0:t})$$



Linearly interpolate between measurement and previous mean based on K

Scale down uncertainty based on K

Linear Gaussian Systems: Observations

- Integrate the effect of an observation using sensor model, after dynamics

$$z_{t+1} = Cx_{t+1} + \delta_t \qquad \delta_t \sim \mathcal{N}(0, R_t)$$

$$p(z_{t+1}|x_{t+1}) = \mathcal{N}(Cx_{t+1}, R_t)$$

$$p(x_{t+1}|z_{0:t+1}, u_{0:t}) \propto p(z_{t+1}|x_{t+1})p(x_{t+1}|z_{0:t}, u_{0:t})$$

Previous belief $p(x_{t+1}|z_{0:t}, u_{0:t}) = \mathcal{N}(\mu_{t+1|0:t}, \Sigma_{t+1|0:t})$ Computed from dynamics step

Updated belief $p(x_{t+1}|z_{0:t+1}, u_{0:t})$

$$= \mathcal{N}(\mu_{t+1|0:t} + K_{t+1}(z_{t+1} - C\mu_{t+1|0:t}), (I - K_{t+1}C)\Sigma_{t+1|0:t})$$

How??

Linear Gaussian Systems: Observations

- Integrate the effect of an observation using sensor model, after dynamics

$$p(x_{t+1}|z_{0:t+1}, u_{0:t}) \propto p(z_{t+1}|x_{t+1})p(x_{t+1}|z_{0:t}, u_{0:t})$$

Stays in Gaussian world, but now conditioning instead of marginalization

Sketch:

1. Construct the joint of x_{t+1} and z_{t+1} conditioned on the past
2. Solve for the mean and covariances of this joint
3. Perform conditioning with z_{t+1} equaling a particular value

Linear Gaussian Systems: Observations

- Integrate the effect of an observation using sensor model, after dynamics

$$p(x_{t+1}|z_{0:t+1}, u_{0:t}) \propto p(z_{t+1}|x_{t+1})p(x_{t+1}|z_{0:t}, u_{0:t})$$

Stays in Gaussian world, but now conditioning instead of marginalization

Linear Gaussian Systems: Observations

- Integrate the effect of an observation using sensor model, after dynamics

$$p(x_{t+1}|z_{0:t+1}, u_{0:t}) \propto p(z_{t+1}|x_{t+1})p(x_{t+1}|z_{0:t}, u_{0:t})$$

Stays in Gaussian world, but now conditioning instead of marginalization

$$X_{t+1|0:t}, Z_{t+1|0:t} \sim \mathcal{N}\left(\begin{bmatrix} \mu^X_{t+1|0:t} \\ \mu^Z_{t+1|0:t} \end{bmatrix}, \begin{bmatrix} \Sigma^{XX}_{t+1|0:t} & \Sigma^{XZ}_{t+1|0:t} \\ \Sigma^{ZX}_{t+1|0:t} & \Sigma^{ZZ}_{t+1|0:t} \end{bmatrix} \right)$$

Belief from the dynamics step

Following the same procedure as last time

$$\begin{bmatrix} \mu^X_{t+1|0:t} \\ \mu^Z_{t+1|0:t} = C\mu^X_{t+1|0:t} \end{bmatrix} \begin{bmatrix} \Sigma^{XX}_{t+1|0:t} & \Sigma^{XZ}_{t+1|0:t} = \Sigma^{XX}_{t+1|0:t}C^T \\ \Sigma^{ZX}_{t+1|0:t} = (\Sigma^{XX}_{t+1|0:t}C^T)^T & \Sigma^{ZZ}_{t+1|0:t} = C\Sigma^{XX}_{t+1|0:t}C^T + R_{t+1} \end{bmatrix}$$

Linear Gaussian Systems: Observations

- Integrate the effect of an observation using sensor model, after dynamics

$$\begin{bmatrix} \mu^X_{t+1|0:t} \\ \\ \mu^Z_{t+1|0:t} = C\mu^X_{t+1|0:t} \end{bmatrix} \begin{bmatrix} \Sigma^{XX}_{t+1|0:t} & \Sigma^{XZ}_{t+1|0:t} = \Sigma^{XX}_{t+1|0:t} C^T \\ \\ \Sigma^{ZX}_{t+1|0:t} = (\Sigma^{XX}_{t+1|0:t} C^T)^T & \Sigma^{ZZ}_{t+1|0:t} = C\Sigma^{XX}_{t+1|0:t} C^T + R_{t+1} \end{bmatrix}$$

$$\mu^Z_{t+1|0:t} = \mathbb{E}\left[Z_{t+1} | X_{t+1}\right]$$

$$\Sigma^{ZZ}_{t+1|0:t} = \mathbb{E}\left[(Z_{t+1} - \mu^Z_{t+1|0:t})(Z_{t+1} - \mu^Z_{t+1|0:t})^T\right]$$

$$\Sigma^{ZZ}_{t+1|0:t} = \mathbb{E}\left[(Z_{t+1} - \mu^Z_{t+1|0:t})(X_{t+1|0:t} - \mu_{t+1|0:t})^T\right]$$

Linear Gaussian Systems: Observations

- Integrate the effect of an observation using sensor model, after dynamics

$$\begin{bmatrix} \mu^X_{t+1|0:t} \\ \\ \mu^Z_{t+1|0:t} = C\mu^X_{t+1|0:t} \end{bmatrix} \begin{bmatrix} \Sigma^{XX}_{t+1|0:t} & \Sigma^{XZ}_{t+1|0:t} = \Sigma^{XX}_{t+1|0:t} C^T \\ \\ \Sigma^{ZX}_{t+1|0:t} = (\Sigma^{XX}_{t+1|0:t} C^T)^T & \Sigma^{ZZ}_{t+1|0:t} = C\Sigma^{XX}_{t+1|0:t} C^T + R_{t+1} \end{bmatrix}$$

Now we just condition on $Z_{t+1} = z_{t+1}$

Remember $p(X|Y = y_0) = \mathcal{N}(\mu_X + \Sigma_{XY}\Sigma_{YY}^{-1}(y_0 - \mu_Y), \Sigma_{XX} - \Sigma_{XY}\Sigma_{YY}^{-1}\Sigma_{YX})$

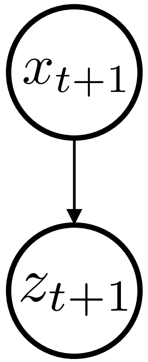
$$p(x_{t+1}|z_{0:t+1}, u_{0:t}) = \mathcal{N}(\mu_{t+1|0:t} + \Sigma_{t+1|0:t}C^T(C\Sigma_{t+1|0:t}C^T + R_{t+1})^{-1}(z_{t+1} - C\mu_{t+1|0:t}),$$
$$\Sigma_{t+1|0:t} - \Sigma_{t+1|0:t}C^T(C\Sigma_{t+1|0:t}C^T + R_{t+1})^{-1}C\Sigma_{t+1|0:t})$$

Linear Gaussian Systems: Observations

- Integrate the effect of an observation using sensor model, after dynamics

$$z_{t+1} = Cx_{t+1} + \delta_t \qquad \delta_t \sim \mathcal{N}(0, R_t)$$

$$p(z_{t+1}|x_{t+1}) = \mathcal{N}(Cx_{t+1}, R_t)$$



$$p(x_{t+1}|z_{0:t+1}, u_{0:t}) = \mathcal{N}(\mu_{t+1|0:t} + \overbrace{\Sigma_{t+1|0:t}C^T(C\Sigma_{t+1|0:t}C^T + R_{t+1})^{-1}}^{\text{Kalman Gain}}(z_{t+1} - C\mu_{t+1|0:t}),$$
$$\Sigma_{t+1|0:t} - \Sigma_{t+1|0:t}C^T(C\Sigma_{t+1|0:t}C^T + R_{t+1})^{-1}C\Sigma_{t+1|0:t})$$

$$K_{t+1} = \Sigma_{t+1|0:t}C^T(C\Sigma_{t+1|0:t}C^T + R_{t+1})^{-1}$$

$$p(x_{t+1}|z_{0:t+1}, u_{0:t}) = \mathcal{N}(\mu_{t+1|0:t} + K_{t+1}(z_{t+1} - C\mu_{t+1|0:t}), (I - K_{t+1}C)\Sigma_{t+1|0:t})$$

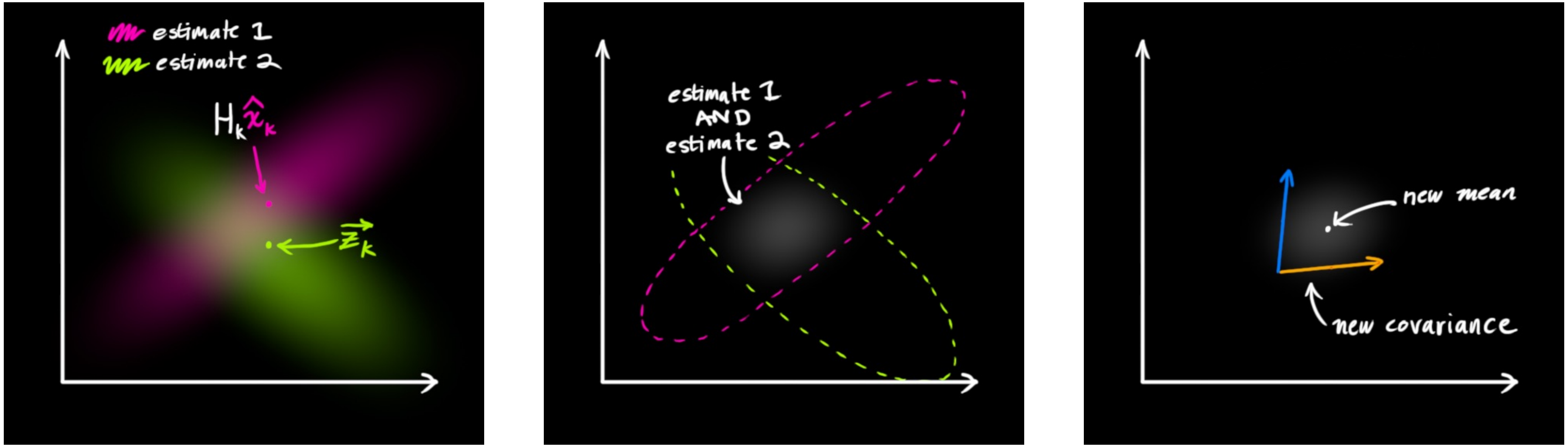
Linear Gaussian Systems: Observations

Previous belief $$p(x_{t+1}|z_{0:t}, u_{0:t}) = \mathcal{N}(\mu_{t+1|0:t}, \Sigma_{t+1|0:t})$$ Computed from dynamics step

Updated belief $$p(x_{t+1}|z_{0:t+1}, u_{0:t})$$

$$= \mathcal{N}(\mu_{t+1|0:t} + K_{t+1}(z_{t+1} - C\mu_{t+1|0:t}), (I - K_{t+1}C)\Sigma_{t+1|0:t})$$

Intuition: Correct the update linearly according to measurement error from expectation, shrink uncertainty accordingly



Unpacking the Kalman Gain

Previous belief

$$p(x_{t+1}|z_{0:t}, u_{0:t}) = \mathcal{N}(\mu_{t+1|0:t}, \Sigma_{t+1|0:t})$$

Computed from dynamics step

Updated belief

$$p(x_{t+1}|z_{0:t+1}, u_{0:t})$$

$$= \mathcal{N}(\mu_{t+1|0:t} + K_{t+1}(z_{t+1} - C\mu_{t+1|0:t}), (I - K_{t+1}C)\Sigma_{t+1|0:t})$$

$$K_{t+1} = \Sigma_{t+1|0:t}C^T(C\Sigma_{t+1|0:t}C^T + R)^{-1}$$

For the sake of simplicity, let's say C = I

in $z_{t+1} = Cx_{t+1} + \delta_t$

$$K_{t+1} = \frac{\Sigma_{t+1|0:t}}{\Sigma_{t+1|0:t} + R}$$

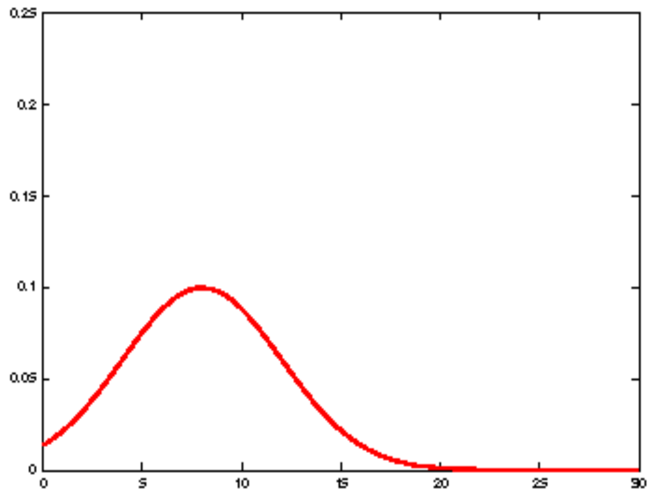
Case 1: Very noisy sensor, R>>Σ

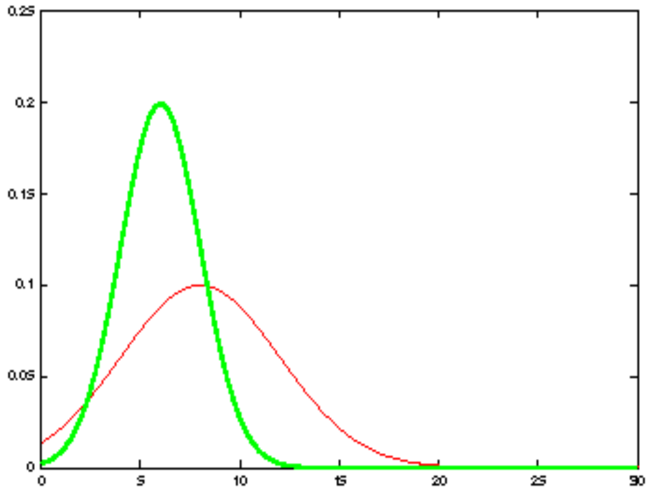
Case 2: Deterministic sensor, R = 0

Intuition Behind Correction Step

Previous belief

New Measurement

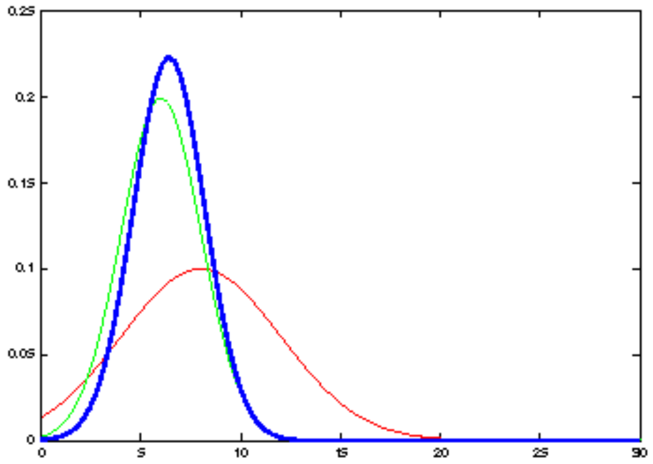




For the sake of simplicity, let's say C = I

$$p(x_{t+1}|z_{0:t+1}, u_{0:t}) = \mathcal{N}(\mu_{t+1|0:t} + K_{t+1}(z_{t+1} - \mu_{t+1|0:t}), (I - K_{t+1})\Sigma_{t+1|0:t})$$

$$K_{t+1} = \frac{\Sigma_{t+1|0:t}}{\Sigma_{t+1|0:t} + R}$$



Corrects belief based on measurement

→ Average between mean and measurement based on K

→ Scale down uncertainty based on K

Kalman Filter Pseudocode

1. def Kalman_filter($\mu_{t|0:t}$, $\Sigma_{t|0:t}$, u_t, z_{t+1}):
2. Prediction:

$$\mu_{t+1|0:t} = A\mu_{t|0:t} + Bu_t$$
$$\Sigma_{t+1|0:t} = A\Sigma_{t|0:t}A^T + Q_t$$

3. Correction:

$$K_{t+1} = \Sigma_{t+1|0:t}C^T(C\Sigma_{t+1|0:t}C^T + R_{t+1})^{-1}$$
$$\mu_{t+1|0:t+1} = \mu_{t+1|0:t} + K_{t+1}(z_{t+1} - C\mu_{t+1|0:t})$$
$$\Sigma_{t+1|0:t+1} = (I - K_{t+1}C)\Sigma_{t+1|0:t}$$

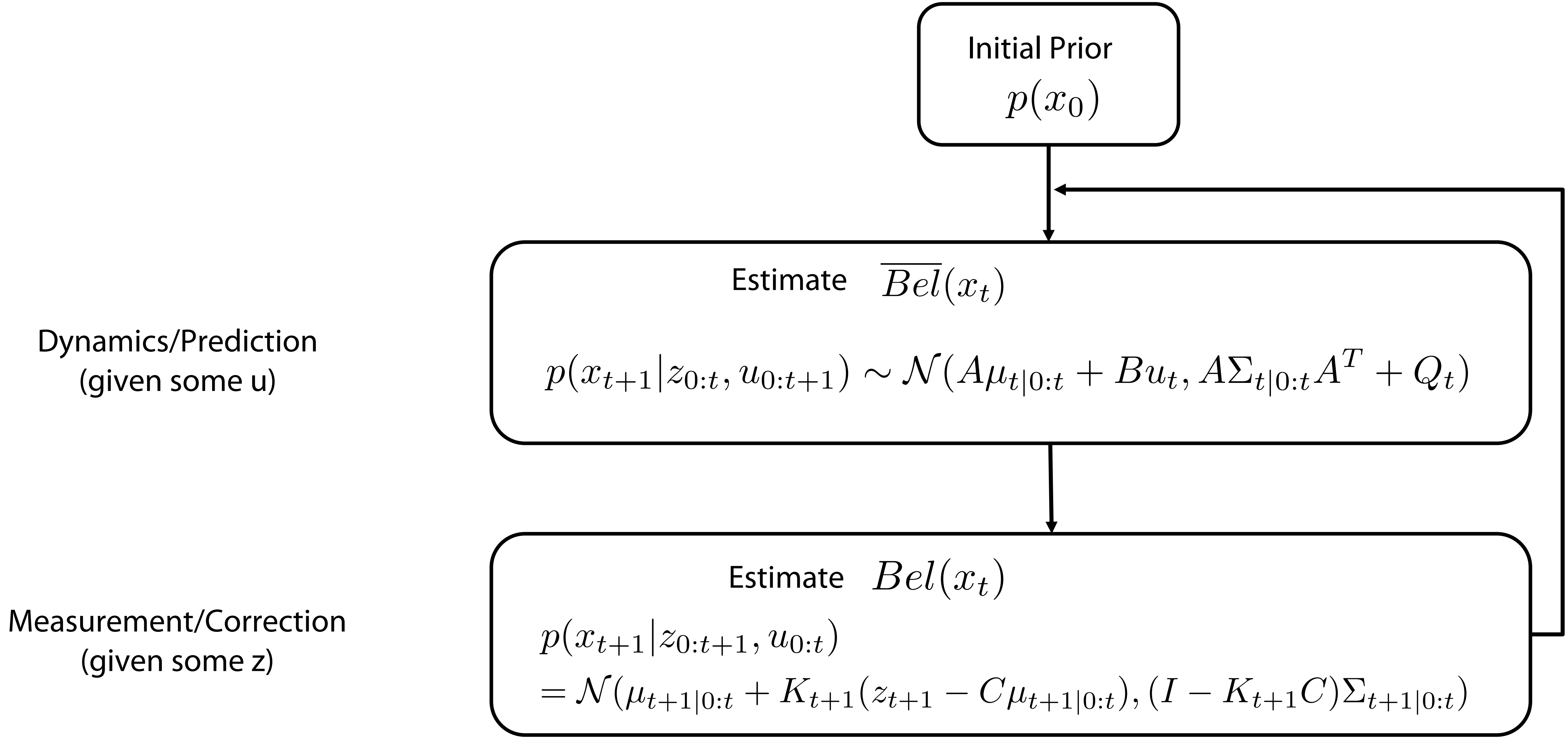
4. Return $\mu_{t+1|0:t+1}$, $\Sigma_{t+1|0:t+1}$

$$x_{t+1} = Ax_t + Bu_t + \epsilon_t$$
$$\epsilon_t \sim \mathcal{N}(0, Q)$$

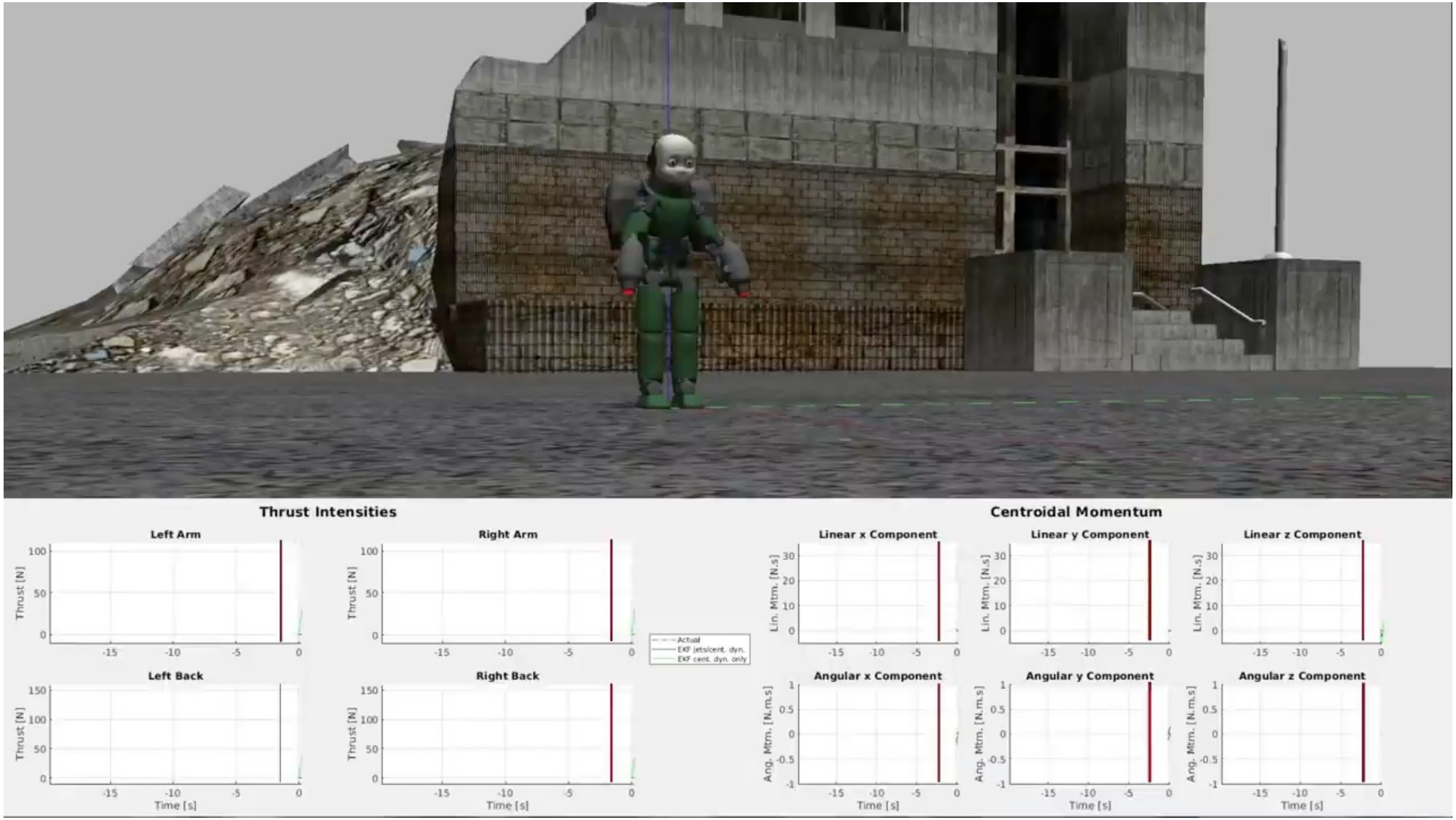
$$z_{t+1} = Cx_{t+1} + \delta_t$$
$$\delta_t \sim \mathcal{N}(0, R)$$

Reminder of the model

Kalman Filter Algorithm



Kalman Filter in Action



Kalman Filter Summary

- Highly efficient: Polynomial in measurement dimensionality ***k*** and state dimensionality ***n***:

  $O(k^{2.376} + n^2)$

Matrix Inversion (Correction)

$$K_{t+1} = \Sigma_{t+1|0:t} C^T (C \Sigma_{t+1|0:t} C^T + R_{t+1})^{-1}$$

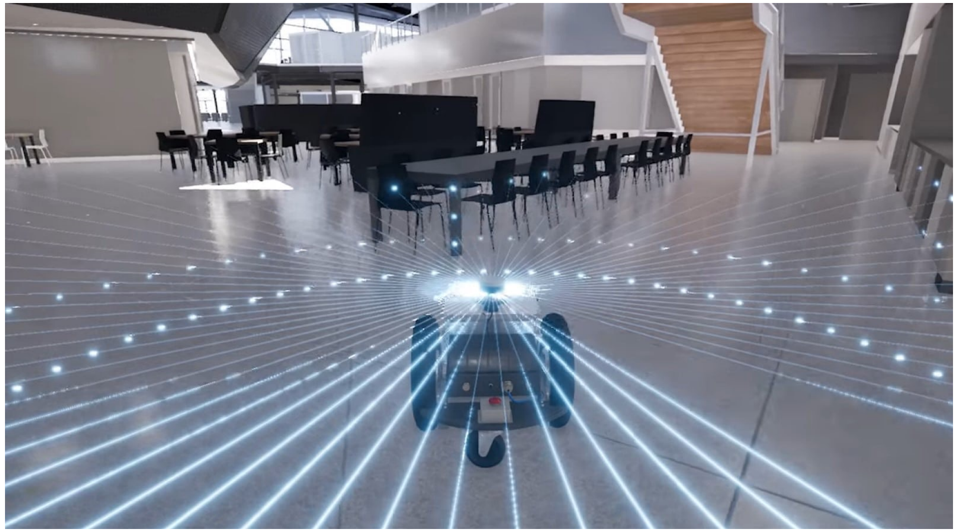
Matrix Multiplication (Prediction)

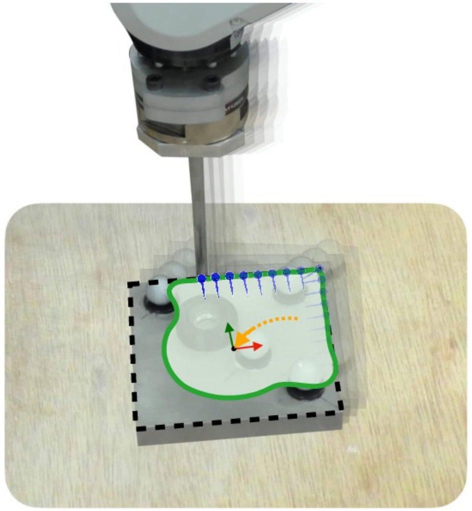
$$p(x_{t+1}|z_{0:t}, u_{0:t+1}) \sim \mathcal{N}(A\mu_{t|0:t} + Bu_t, A\Sigma_{t|0:t}A^T + Q_t)$$

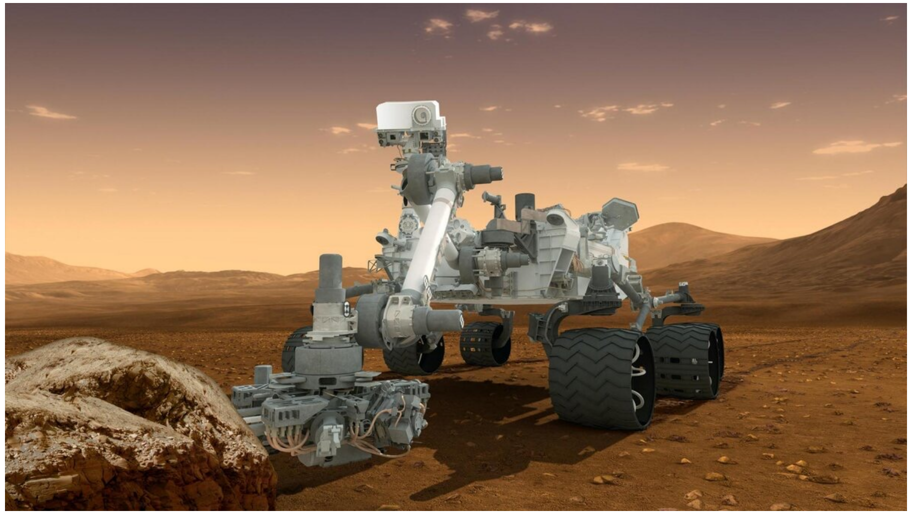
- Optimal for linear Gaussian systems!
- Most robotics systems are nonlinear! → next time

Why should we care in 2023?

Still a very widely used technique for estimation/localization/mapping in real problems







Why should we care in 2023?

Embed to Control: A Locally Li
Dynamics Model for Control from

Manuel Watter* **Jost Tobias Springenberg*** **Martin Riedmiller**
Joschka Boedecker Google DeepMind
University of Freiburg, Germany
{watterm,springj,jboedeck}@cs.uni-freib

Mastering Diverse Domains through World Models

Danijar Hafner[1,2] Jurgis Pasukonis[1] Jimmy Ba[2] Timothy Lillicrap[1]

[1]DeepMind [2]University of Toronto

SOLAR: Deep Structured Representations for Model-Based Reinforcement Learning

Marvin Zhang [*1] **Sharad Vikram** [*2] **Laura Smith** [1] **Pieter Abbeel** [1] **Matthew J. Johnson** [3] **Sergey Levine** [1]

Lecture Outline

Kalman Filtering

↓

Extended Kalman Filter

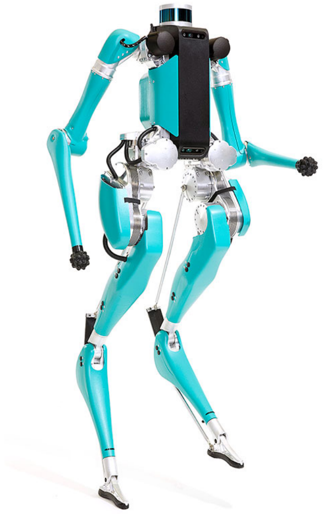
↓

Unscented Kalman Filter

Realistic Robotic Systems

- Most realistic robotic problems involve nonlinear functions

$$x_{t+1} = Ax_t + Bu_t + \epsilon_t$$
$$\epsilon_t \sim \mathcal{N}(0, Q)$$



$$z_{t+1} = Cx_{t+1} + \delta_t$$
$$\delta_t \sim \mathcal{N}(0, R)$$

Nonlinear Dynamic Systems

- Most realistic robotic problems involve nonlinear functions

$$x_{t+1} = g(x_t, u_t) + \epsilon_t$$

$$z_t = h(x_t) + \delta_t$$

Non-linear system

$$\epsilon_t \sim \mathcal{N}(0, Q)$$

$$\delta_t \sim \mathcal{N}(0, R)$$

Additive Gaussian noise

More reasonable assumption than linear Gaussian. More on non-Gaussian systems next time

How do we deal with non-linearity?

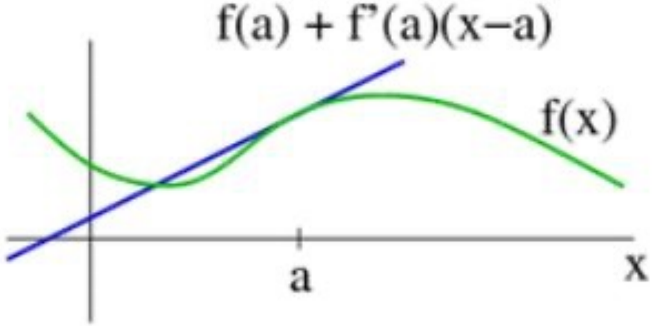
- Differentiable non-linear functions can be expressed via their Taylor expansion

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \cdots,$$

$$f(x) \approx f(a) + \frac{f'(a)}{1!}(x-a)$$

Dropping higher order terms, when x-a is small enough

Linear function in x



Pretend that your function is linear in this neighborhood
→ Reapprox in a new neighborhood

EKF Linearization: First Order Taylor Series Expansion

- Idea behind EKF: Linearize the dynamics and measurement around current μ_t
- Dynamics Model (linearize around previous belief):

$$x_{t+1} = g(x_t, u_t) + \epsilon_t \approx g(\mu_t, u_t) + \left.\frac{\partial g(x_t, u_t)}{\partial x_t}\right|_{x_t=\mu_t} (x_t - \mu_t) + \epsilon_t$$

$$= g(\mu_t, u_t) + G(x_t - \mu_t) + \epsilon_t$$

- Measurement Model (linearize around post dynamics belief):

$$z_t = h(x_t) + \delta_t \approx h(\bar{\mu}_t) + \left.\frac{\partial h(x_t)}{\partial x_t}\right|_{x_t=\bar{\mu}_t} (x_t - \bar{\mu}_t) + \delta_t \approx h(\bar{\mu}_t) + H(x_t - \bar{\mu}_t) + \delta_t$$

Now everything is linear → back to Kalman filtering!

Modified System under EKF Linearization

- Start by linearizing dynamics model under current belief
- Dynamics Model (linearize around previous belief):

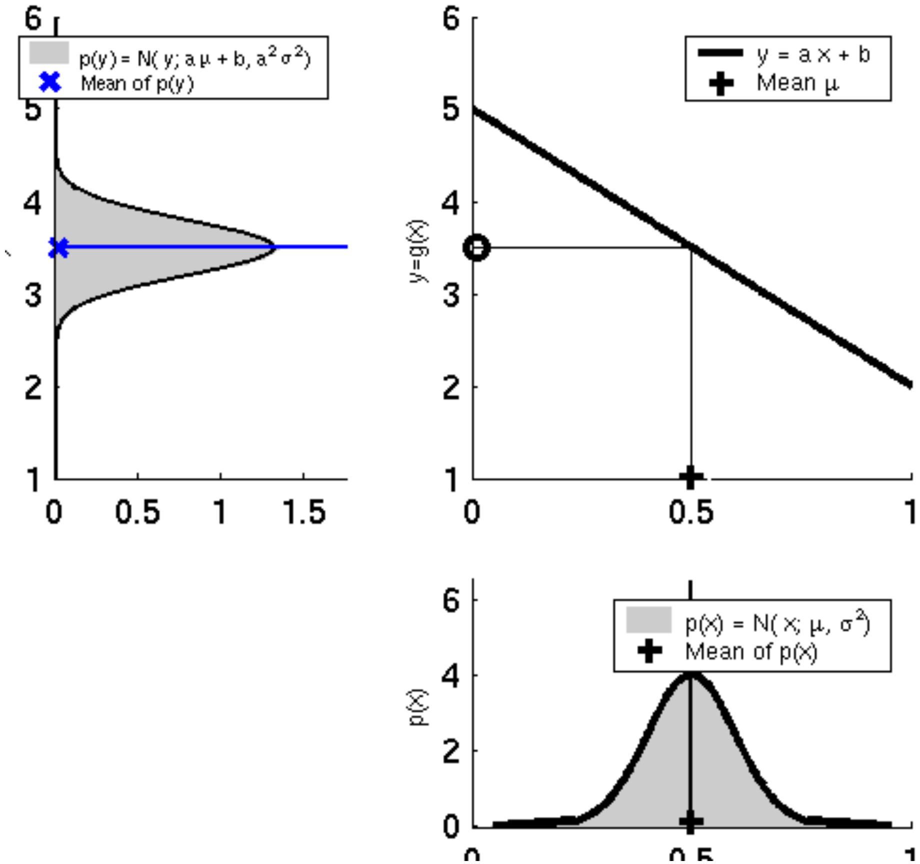
$$x_{t+1} = g(x_t, u_t) + \epsilon_t \qquad \approx g(\mu_t, u_t) + \left.\frac{\partial g(x_t, u_t)}{\partial x_t}\right|_{x_t=\mu_t} (x_t - \mu_t) + \epsilon_t$$

- Perform dynamics update
- Linearize measurement around post dynamics belief

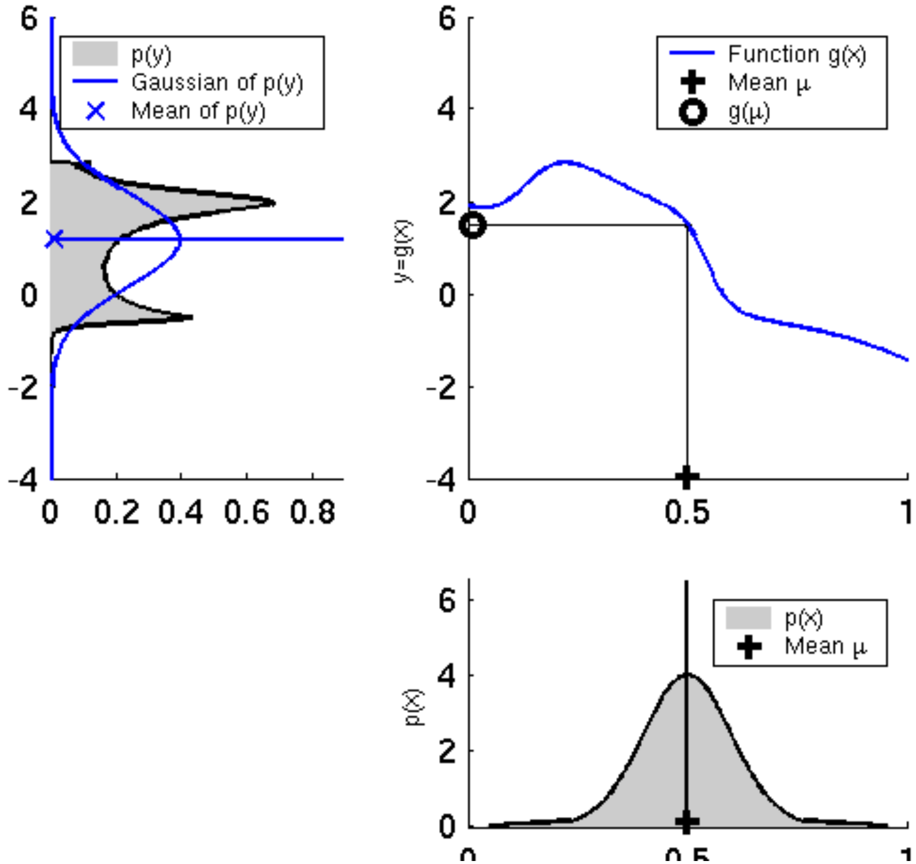
$$z_t = h(x_t) + \delta_t \approx h(\bar{\mu}_t) + \left.\frac{\partial h(x_t)}{\partial x_t}\right|_{x_t=\bar{\mu}_t} (x_t - \bar{\mu}_t) + \delta_t \approx h(\bar{\mu}_t) + H(x_t - \bar{\mu}_t) + \delta_t$$

- Perform measurement update
- Repeat

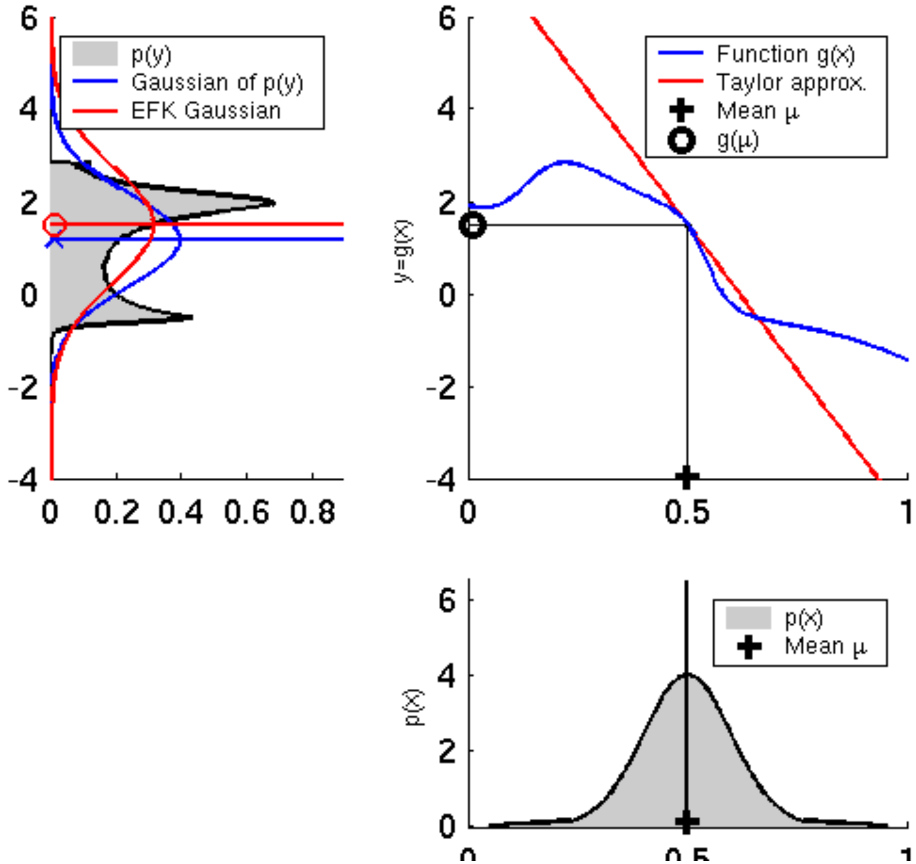
Linearity Assumption Revisited



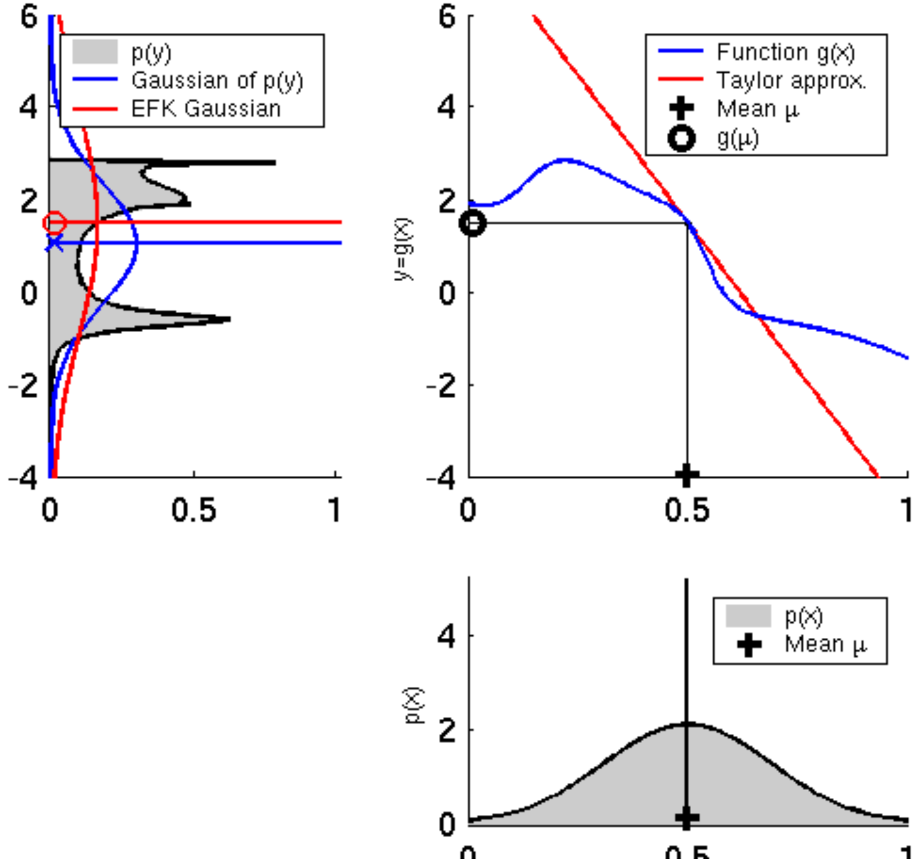
Non-linear Function



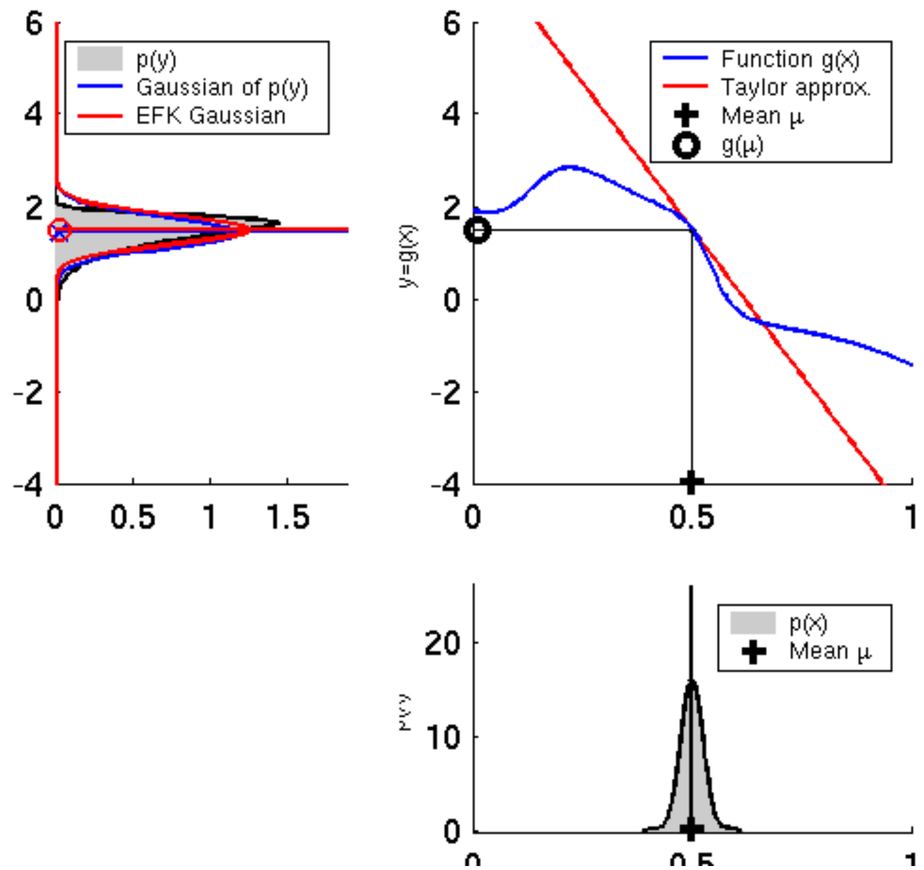
EKF Linearization (1)



EKF Linearization (2)



EKF Linearization (3)



EKF Dynamics Step

- Linearize dynamics around current mean belief

$$x_{t+1} = g(x_t, u_t) + \epsilon_t \approx g(\mu_t, u_t) + \left. \frac{\partial g(x_t, u_t)}{\partial x_t} \right|_{x_t = \mu_t} (x_t - \mu_t) + \epsilon_t$$

$$= g(\mu_t, u_t) + G(x_t - \mu_t) + \epsilon_t$$

- Run standard Kalman filter with A = G and Bu = $g(\mu_t, u_t)$

$$p(x_{t+1} | z_{0:t}, u_{0:t})$$
$$\sim \mathcal{N}(A\mu_{t|0:t} + Bu_t, A\Sigma_{t|0:t}A^T + Q_t)$$

Standard Kalman filter

⇒

$$p(x_{t+1} | z_{0:t}, u_{0:t})$$
$$\sim \mathcal{N}(g(\mu_t, u_t), G\Sigma_{t|0:t}G^T + Q_t)$$

Extended Kalman filter

EKF Measurement Step

- Linearize measurement around post dynamics belief:

$$z_t = h(x_t) + \delta_t \approx h(\bar{\mu}_t) + \left.\frac{\partial h(x_t)}{\partial x_t}\right|_{x_t=\bar{\mu}_t} (x_t - \bar{\mu}_t) + \delta_t \approx h(\bar{\mu}_t) + H(x_t - \bar{\mu}_t) + \delta_t$$

- Run standard Kalman filter with C = H

$$p(x_{t+1}|z_{0:t+1}, u_{0:t})$$
$$= \mathcal{N}(\mu_{t+1|0:t} + K_{t+1}(z_{t+1} - C\mu_{t+1|0:t}), (I - K_{t+1}C)\Sigma_{t+1|0:t})$$
$$K_{t+1} = \Sigma_{t+1|0:t}C^T(C\Sigma_{t+1|0:t}C^T + R)^{-1}$$

Standard Kalman Filter

$$p(x_{t+1}|z_{0:t+1}, u_{0:t})$$
$$= \mathcal{N}(\mu_{t+1|0:t} + K_{t+1}(z_{t+1} - h(\bar{\mu}_t), (I - K_{t+1}H)\Sigma_{t+1|0:t})$$
$$K_{t+1} = \Sigma_{t+1|0:t}H^T(H\Sigma_{t+1|0:t}H^T + R)^{-1}$$

Extended Kalman Filter

EKF Algorithm

Initial Prior

$$p(x_0)$$

Linearize dynamics

$$x_{t+1} = g(x_t, u_t) + \epsilon_t \approx g(\mu_t, u_t) + \left.\frac{\partial g(x_t, u_t)}{\partial x_t}\right|_{x_t=\mu_t} (x_t - \mu_t) + \epsilon_t$$

Dynamics/Prediction
(given some u)

Estimate $\overline{Bel}(x_t)$

$$p(x_{t+1} | z_{0:t}, u_{0:t}) \sim \mathcal{N}(g(\mu_t, u_t), G\Sigma_{t|0:t}G^T + Q_t)$$

Linearize measurement

$$z_t = h(x_t) + \delta_t \approx h(\bar{\mu}_t) + \left.\frac{\partial h(x_t)}{\partial x_t}\right|_{x_t=\bar{\mu}_t} (x_t - \bar{\mu}_t) + \delta_t$$

Measurement/Correction
(given some z)

Estimate $Bel(x_t)$

$$p(x_{t+1} | z_{0:t+1}, u_{0:t}) = \mathcal{N}(\mu_{t+1|0:t} + K_{t+1}(z_{t+1} - h(\bar{\mu}_t), (I - K_{t+1}H)\Sigma_{t+1|0:t})$$

EKF Pseudocode

1. def Extended_Kalman_filter($\mu_{t|0:t}$, $\Sigma_{t|0:t}$, u_t, z_{t+1}):
2. Linearize dynamics: $G = \left.\frac{\partial g(x_t, u_t)}{\partial x_t}\right|_{x_t = \mu_{t|0:t}}$
3. Prediction:

$$\mu_{t+1|0:t} = g(\mu_{t|0:t}, u_t)$$
$$\Sigma_{t+1|0:t} = G\Sigma_{t|0:t}G^T + Q$$

4. Linear measurement: $H = \left.\frac{\partial h(x_t)}{\partial x_t}\right|_{x_t = \mu_{t+1|0:t}}$
5. Correction: C = H

$$K_{t+1} = \Sigma_{t+1|0:t}H^T(H\Sigma_{t+1}H^T + R)^{-1}$$
$$\mu_{t+1|0:t+1} = \mu_{t+1|0:t} + K_{t+1}(z_{t+1} - h(\mu_{t+1|0:t}))$$
$$\Sigma_{t+1|0:t+1} = (I - K_{t+1}H)\Sigma_{t+1|0:t}$$

6. Return $\mu_{t+1|0:t+1}$, $\Sigma_{t+1|0:t+1}$

$$x_{t+1} = g(x_t, u_t) + \epsilon_t$$
$$z_t = h(x_t) + \delta_t$$
$$\epsilon_t \sim \mathcal{N}(0, Q)$$
$$\delta_t \sim \mathcal{N}(0, R)$$

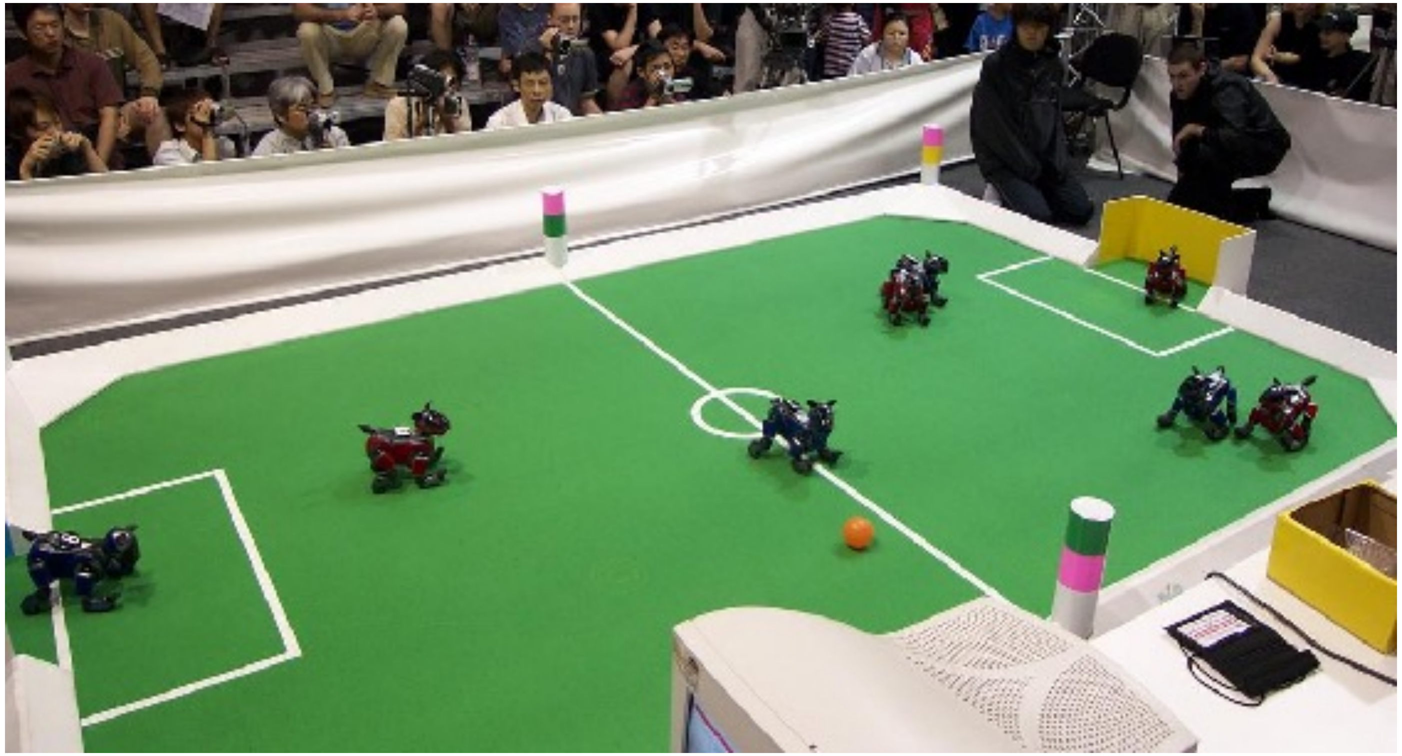
Reminder of the model

Localization

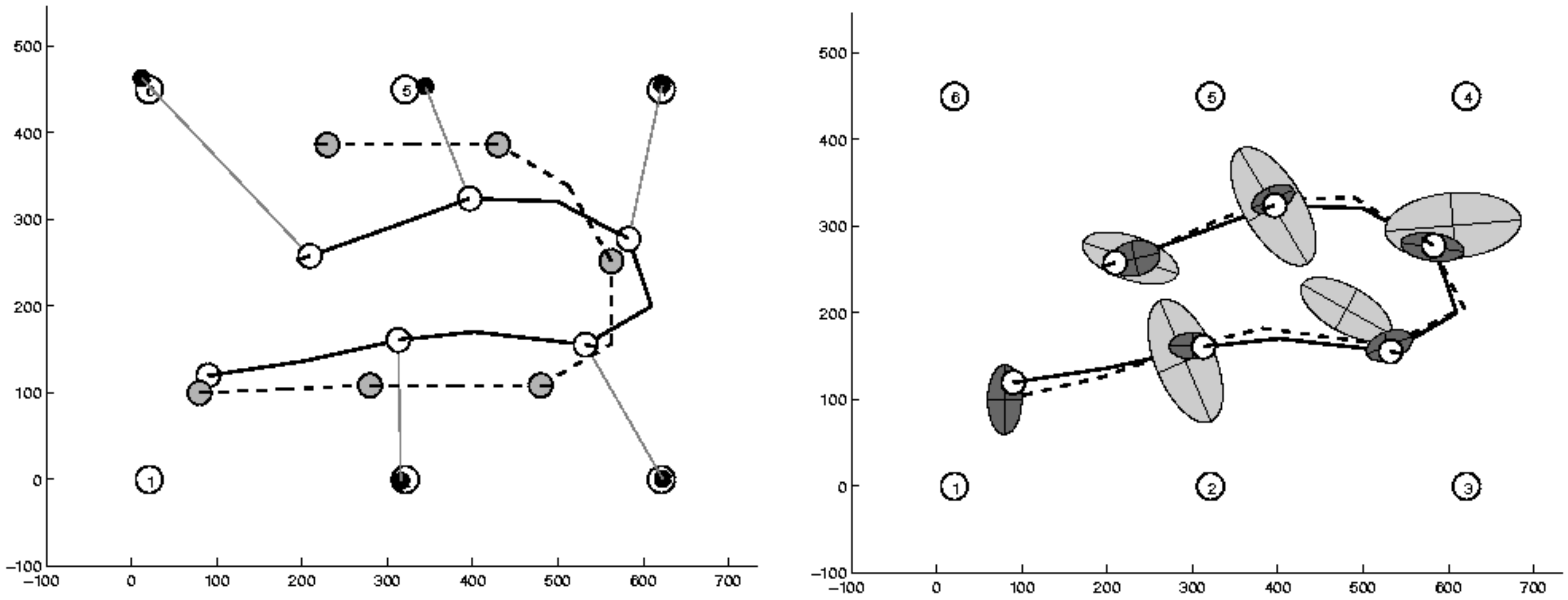
> "Using sensory information to locate the robot in its environment is the most fundamental problem to providing a mobile robot with autonomous capabilities." [Cox '91]

- **Given**
  - Map of the environment.
  - Sequence of sensor measurements.
- **Wanted**
  - Estimate of the robot's position.
- **Problem classes**
  - Position tracking
  - Global localization
  - Kidnapped robot problem (recovery)

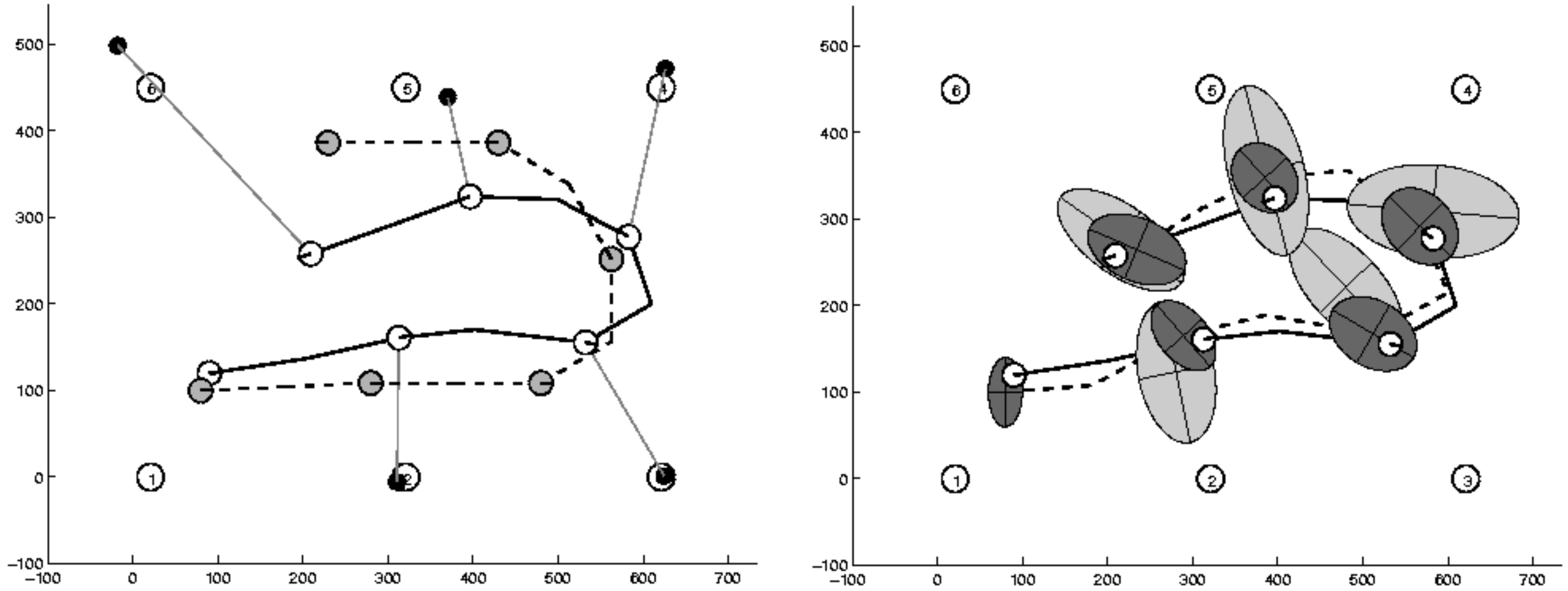
Landmark-based Localization



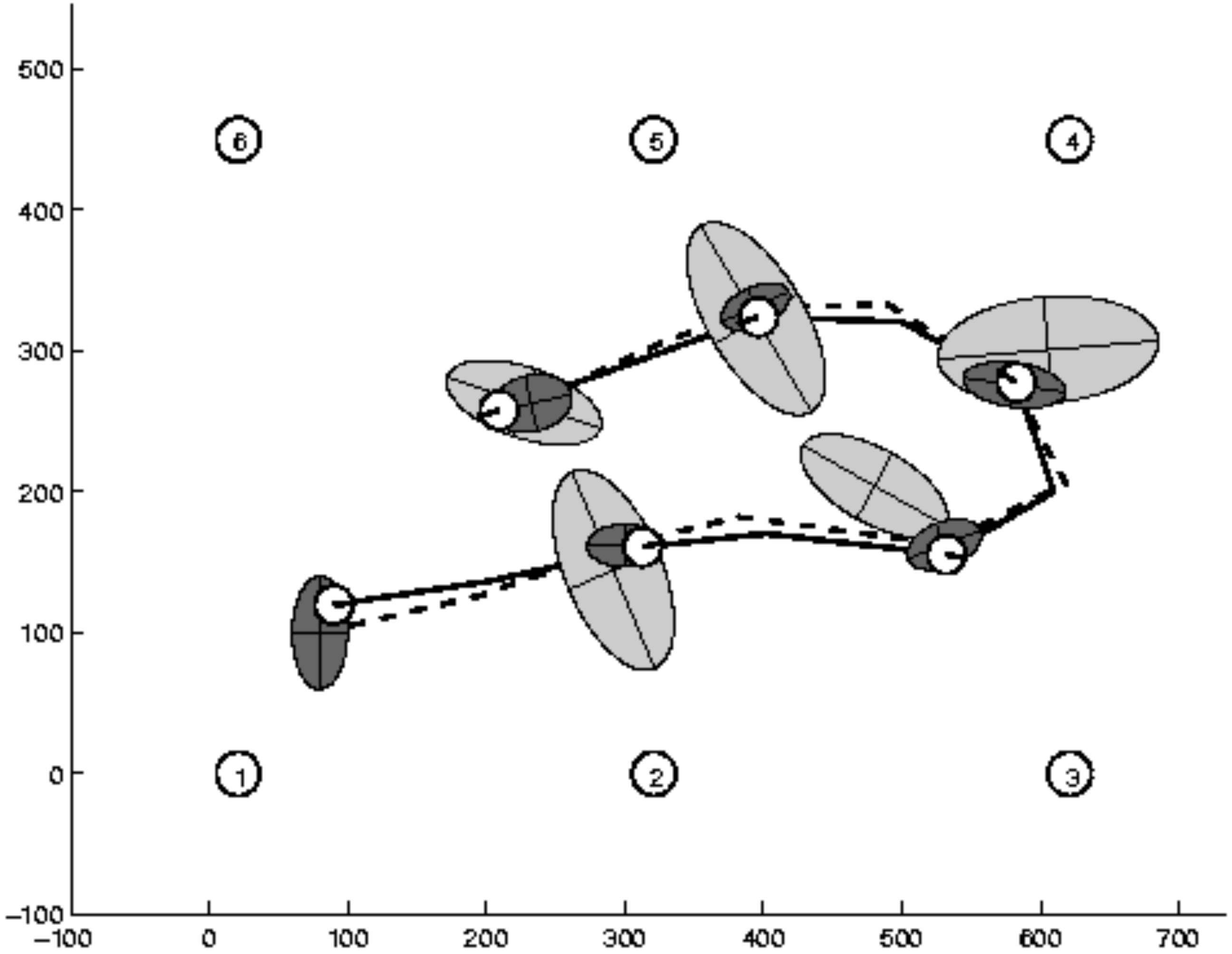
Estimation Sequence (1)



Estimation Sequence (2)



Comparison to GroundTruth



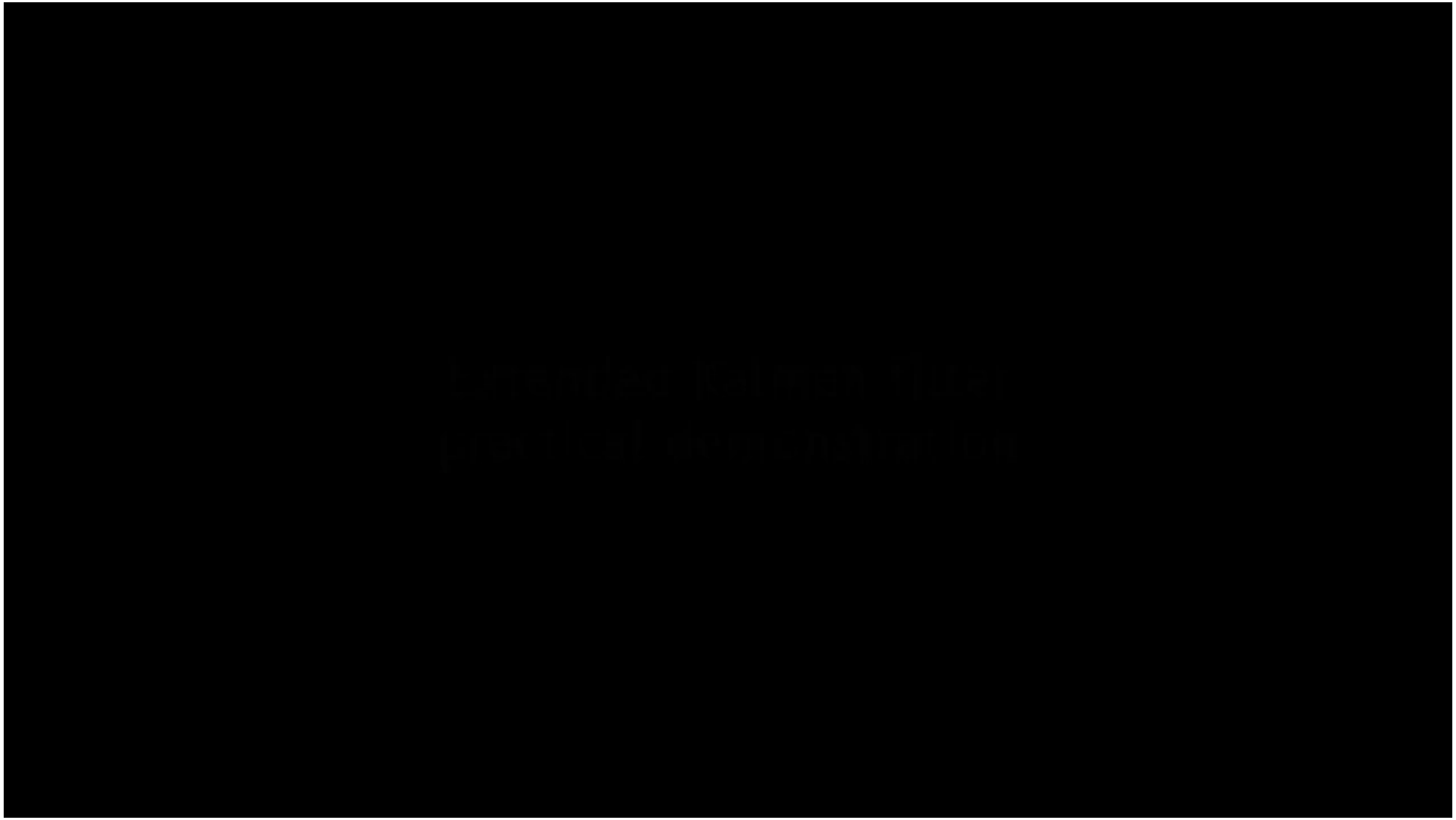
EKF Summary

- Highly efficient: Polynomial in measurement dimensionality ***k*** and state dimensionality ***n***:

  $$O(k^{2.376} + n^2)$$

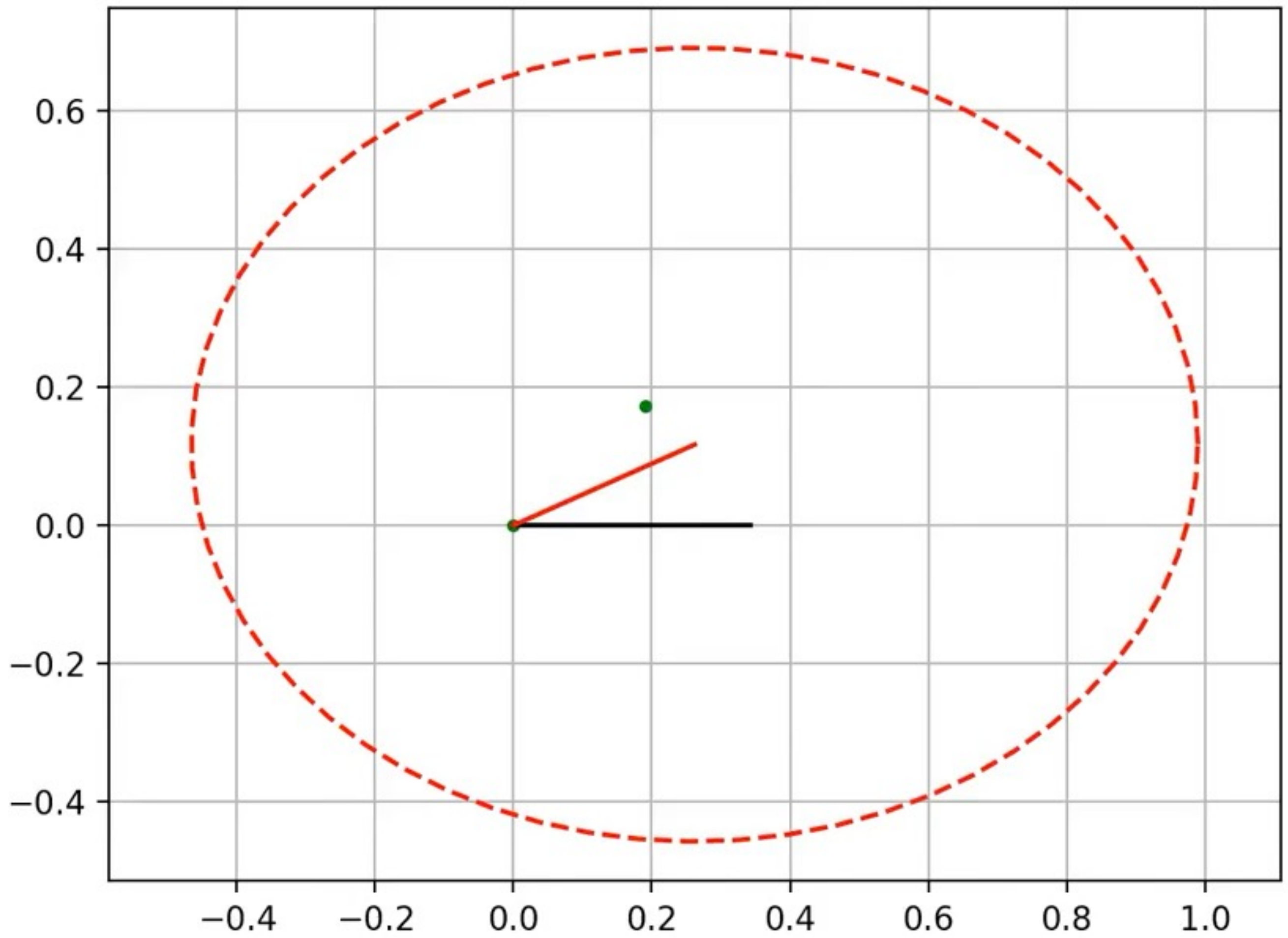
- Not optimal!
- Can diverge if nonlinearities are large!
- Works surprisingly well even when all assumptions are violated!

EKF in Action



https://www.youtube.com/watch?v=dEd4cNfmi0s&ab_channel=ThePoorEngineer

EKF in Action



https://www.youtube.com/watch?v=ju27V142D2o&ab_channel=AtsushiSakai

When does the EKF struggle?

- With discontinuous dynamics, the linearization will not be valid
- For very non-linear functions, the first order Taylor approximation is poor
- The EKF can drift over time because of growing linearization errors
- Jacobian may be very expensive to compute and invert

Lecture Outline

Kalman Filtering

↓

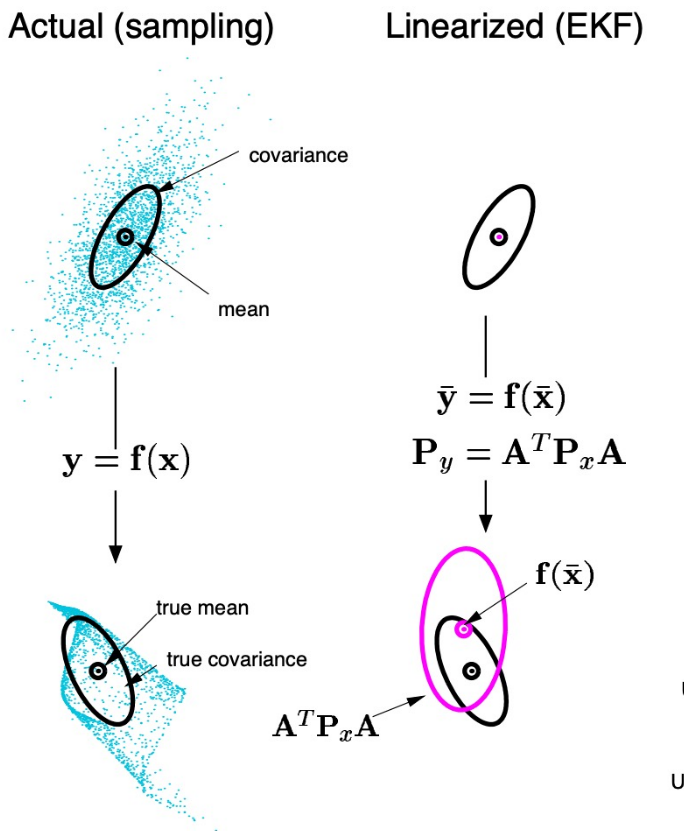
Extended Kalman Filter

↓

Unscented Kalman Filter

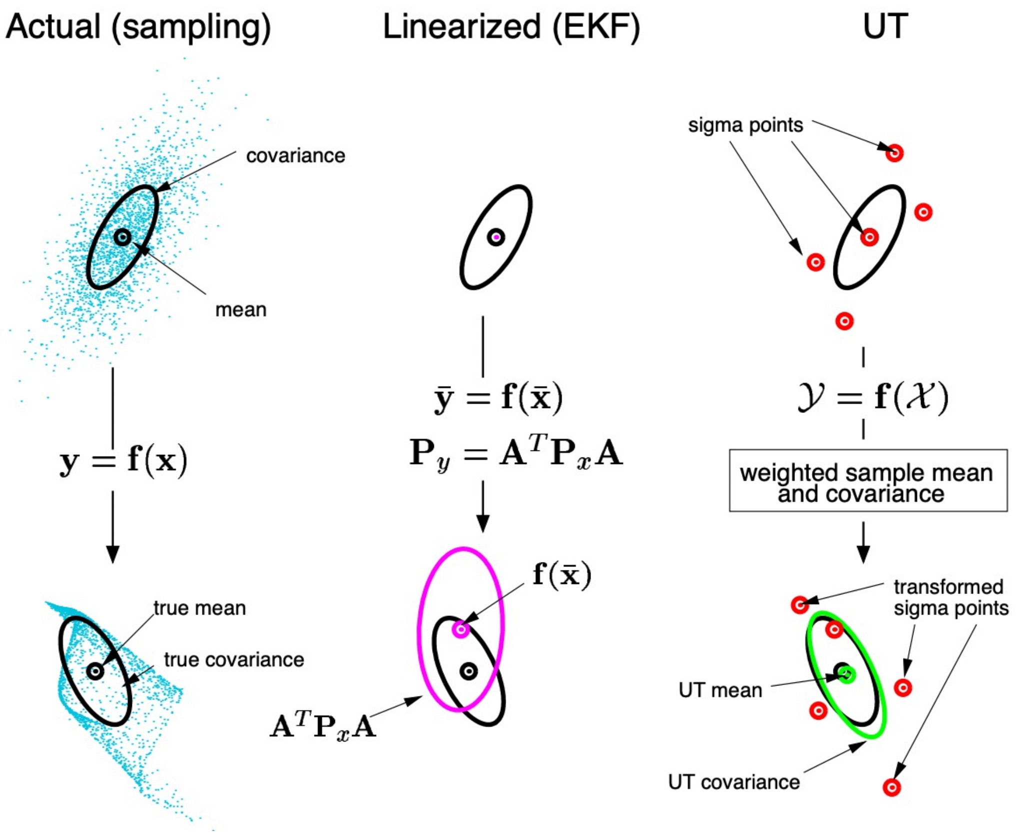
How can we achieve closer approximations?

- Extended Kalman filters first linearize then send through Gaussian, can be quite poor when the dynamics/measurements are quite non-linear



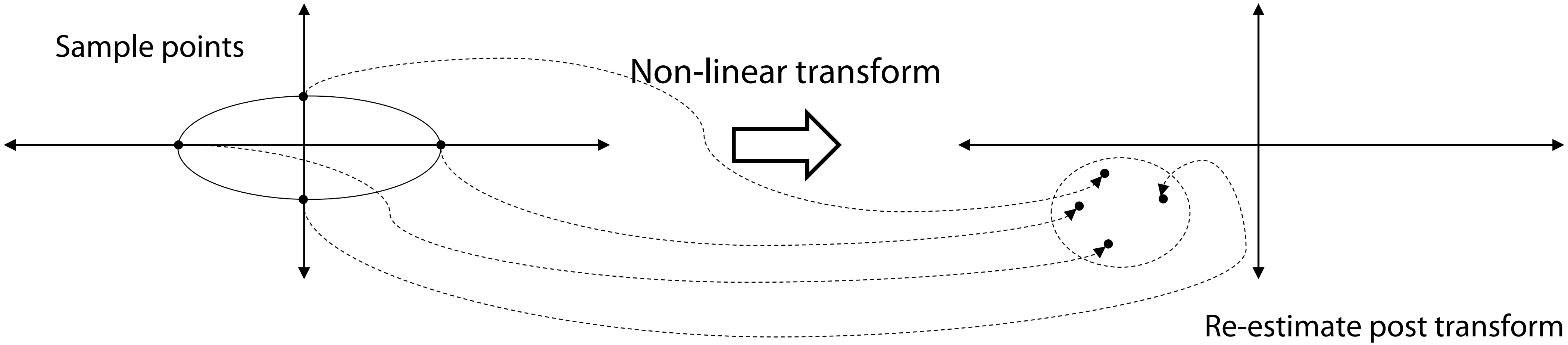
How can we achieve closer approximations?

- Extended Kalman filters first linearize then send through Gaussian, can be quite poor when the dynamics/measurements are quite non-linear



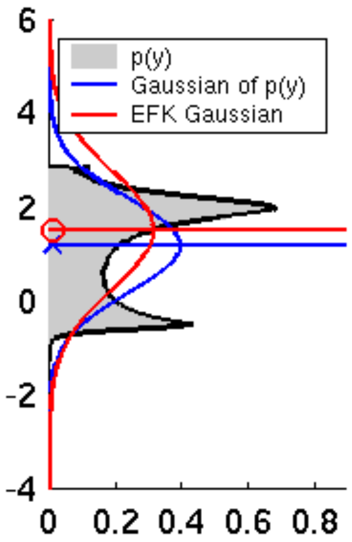
How can we achieve closer approximations?

- Idea: Rather than linearizing first and then propagate, propagate through non-linear transform and re-estimate Gaussian

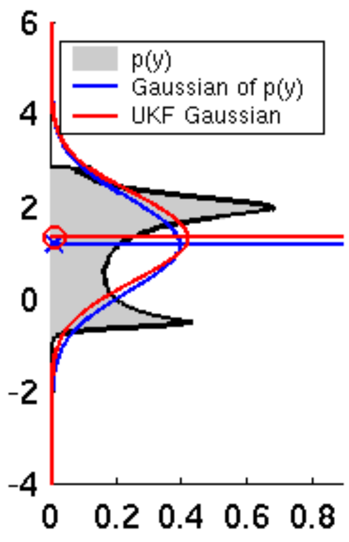


- Ensure that first and second moments (mean and covariance) match as closely as possible on re-estimation

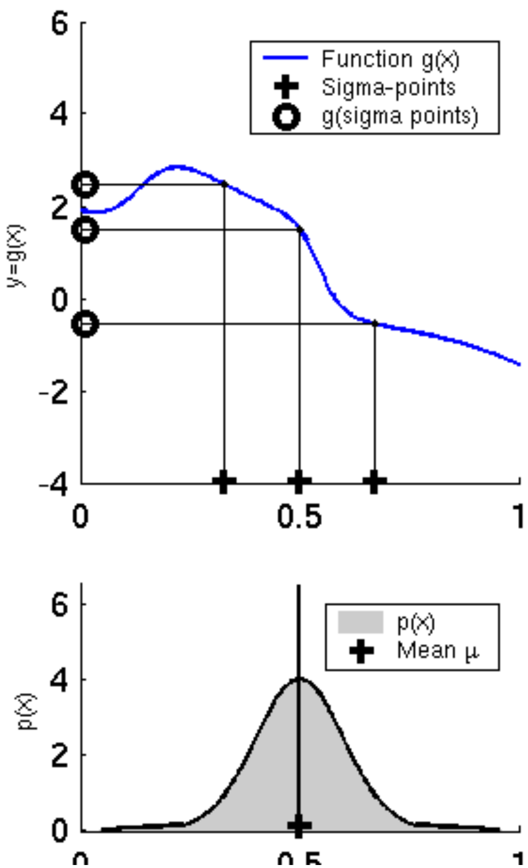
Linearization via Unscented Transform



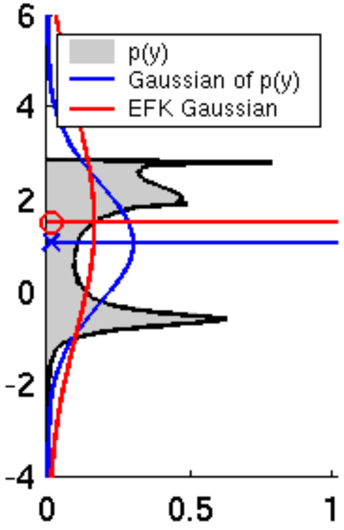
EKF



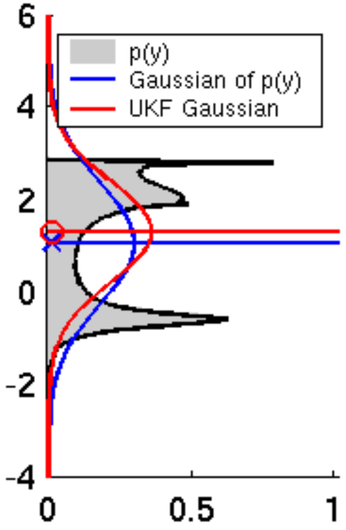
UKF



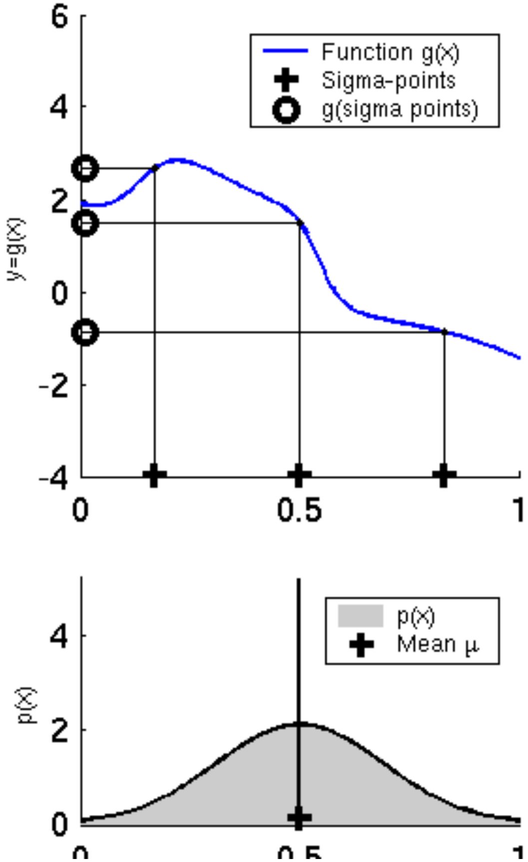
UKF Sigma-Point Estimate (2)



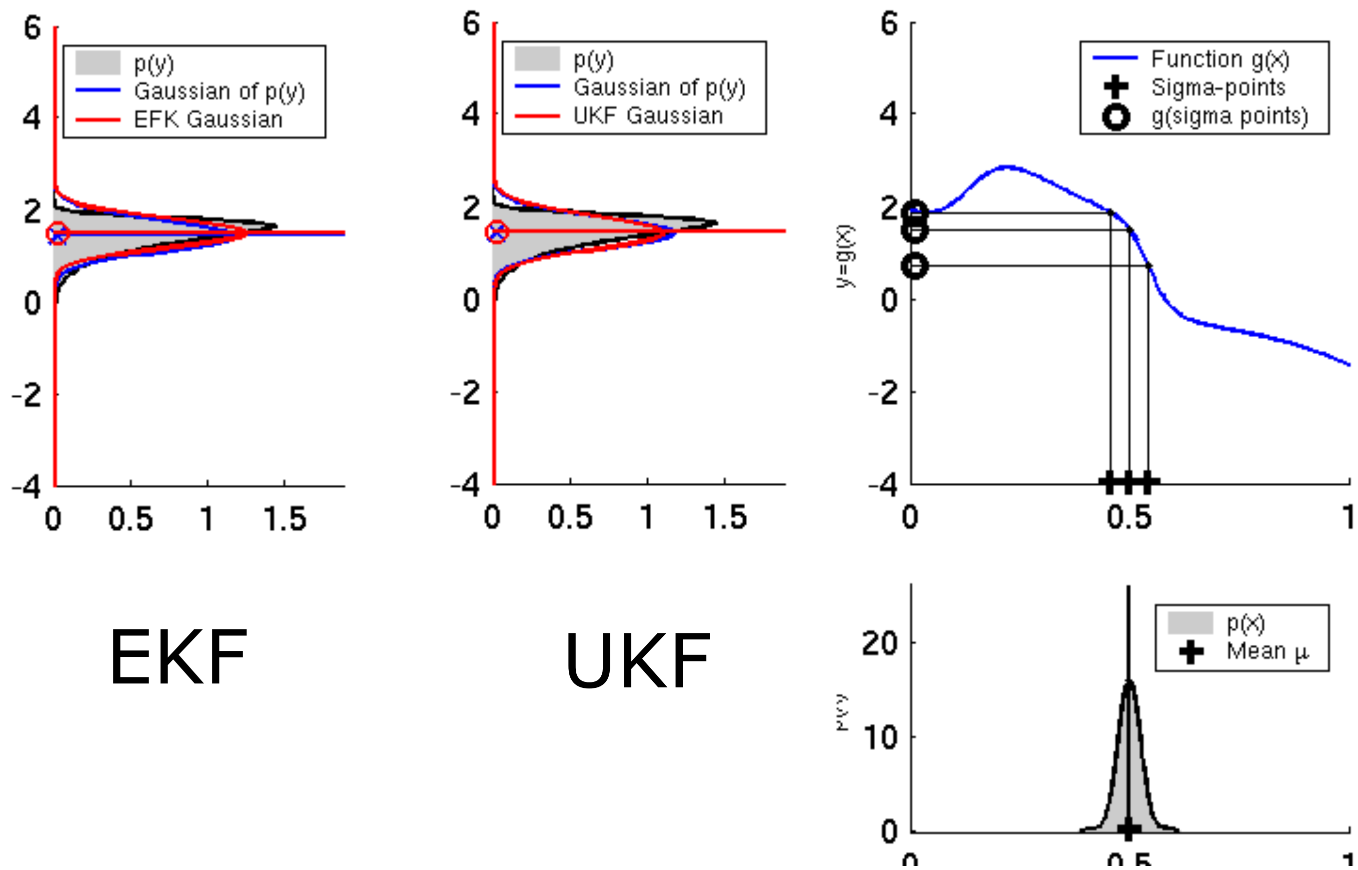
EKF



UKF



UKF Sigma-Point Estimate (3)

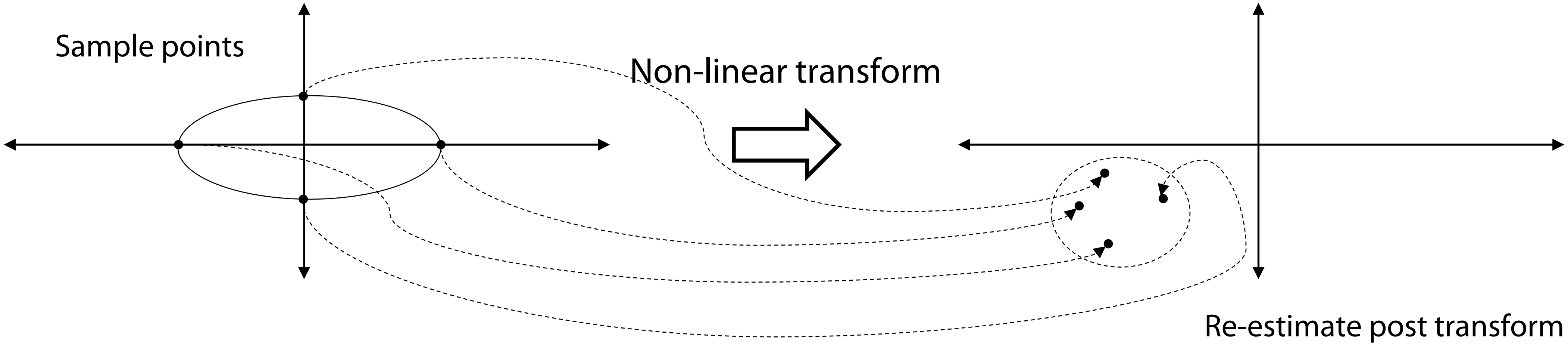


EKF

UKF

How can we achieve closer approximations?

- Idea: Rather than linearizing first and then propagate, propagate through non-linear transform and re-estimate Gaussian



- Question 1: What points should we send through non linearity?
- Question 2: How should we restimate the means and covariances?
- Question 3: Why can this be better than the EKF?

Sigma Points

- Question 1: What points should we send through non linearity?
  - Choose minimal points (2N + 1 to send through non-linearity to match 1,2 moments of a Gaussian

Sigma points

$$\chi^0 = \mu$$

$$\chi^i = \mu \pm \left(\sqrt{(n+\lambda)\Sigma}\right)_i$$

Weights

$$w_m^0 = \frac{\lambda}{n+\lambda} \qquad w_c^0 = \frac{\lambda}{n+\lambda}$$

$$w_m^i = w_c^i = \frac{1}{2(n+\lambda)} \qquad \text{for } i = 1,\ldots,2n$$

- What is a matrix square root?

$$L = \sqrt{\Sigma} \quad \text{if} \quad LL^T = \Sigma$$

- Why these points → they ensure that the moments match. Not a unique choice!

Unscented Transform

- Question 2: How should we re-estimate the means and covariances?

Sigma points

$$\chi^0 = \mu$$

$$\chi^i = \mu \pm \left(\sqrt{(n+\lambda)\Sigma}\right)_i$$

Weights

$$w_m^0 = \frac{\lambda}{n+\lambda} \qquad w_c^0 = \frac{\lambda}{n+\lambda}$$

$$w_m^i = w_c^i = \frac{1}{2(n+\lambda)} \qquad \text{for } i = 1,\ldots,2n$$

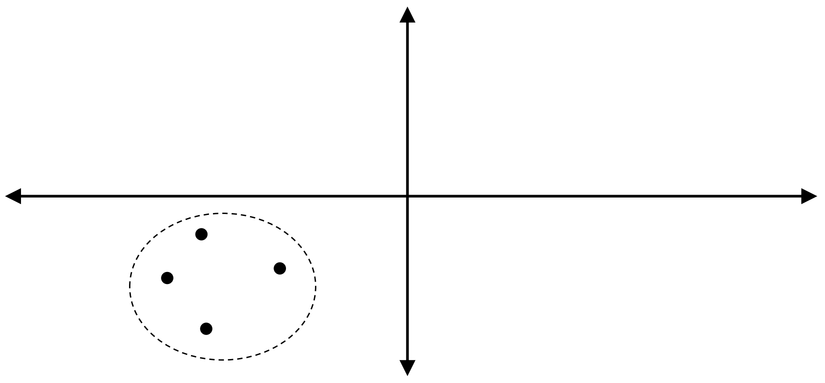
Pass sigma points through nonlinear function

$$\psi^i = g(\chi^i)$$

Recover mean and covariance

$$\mu' = \sum_{i=0}^{2n} w_m^i \psi^i$$

$$\Sigma' = \sum_{i=0}^{2n} w_c^i (\psi^i - \mu)(\psi^i - \mu)^T$$



Re-estimate post transform

Why do these make sense?

Sigma points

$$\chi^0 = \mu$$

$$\chi^i = \mu \pm \left(\sqrt{(n+\lambda)\Sigma}\right)_i$$

Weights

$$w_m^0 = \frac{\lambda}{n+\lambda} \qquad w_c^0 = \frac{\lambda}{n+\lambda}$$

$$w_m^i = w_c^i = \frac{1}{2(n+\lambda)} \qquad \text{for } i = 1,\ldots,2n$$

Filtering with the Unscented Transform

- Given the tool of the unscented transform, let us revisit the nonlinear filter

$$p(x_0)$$

Dynamics

Estimate $\overline{Bel}(x_t)$

$$\overline{Bel}(x_t) = \int p(x_t | x_{t-1}, u_{t-1}) Bel(x_{t-1}) dx_{t-1}$$

→ Directly use unscented transform for dynamics

Measurement

Estimate $Bel(x_t)$

$$Bel(x_t) = \eta p(z_t | x_t) \overline{Bel}(x_t) = \frac{p(z_t | x_t) \overline{Bel}(x_t)}{\sum_{x'} p(z_t | x_t = x') \overline{Bel}(x_t = x')}$$

→ Estimate empirical covariance matrix with unscented transform and do Kalman filter

Unscented KF Dynamics Step

- Sample Sigma points given current belief, send them through non-linear dynamics
- Re-estimate the post-update belief using the unscented transform

$$\chi^0 = \mu_{t|0:t}$$

$$\chi^i = \mu_{t|0:t} \pm (\sqrt{(n+\lambda)\Sigma_{t|0:t}})_i$$

$$w_m^0 = \frac{\lambda}{n+\lambda} \qquad w_c^0 = \frac{\lambda}{n+\lambda}$$

$$w_m^i = w_c^i = \frac{1}{2(n+\lambda)} \qquad \text{for } i = 1,\ldots,2n$$

Pass sigma points through nonlinear function $\psi^i = g(\chi^i, u_t)$

$$\mu_{t+1|0:t} = \sum_{i=0}^{2n} w_m^i \psi^i$$

$$\Sigma_{t+1|0:t} = \sum_{i=0}^{2n} w_c^i (\psi^i - \mu_{t+1|0:t})(\psi^i - \mu_{t+1|0:t})^T + Q$$

Unscented KF Measurement Step

- More tricky because now C/H is not known! How to compute Kalman gain?

$$K_{t+1} = \Sigma_{t+1|0:t} C^T (C \Sigma_{t+1|0:t} C^T + R_{t+1})^{-1}$$

Cross covariance under forward transform

Covariance under forward transform

Remember from earlier

Diagonal Covariance

$$\begin{aligned}
\Sigma_{t+1|0:t} &= \mathbb{E}\left[(X_{t+1|0:t} - \mu_{t+1|0:t})(X_{t+1|0:t} - \mu_{t+1|0:t})^T\right] \\
&= \mathbb{E}\left[(AX_{t|0:t} + Bu_t + \epsilon_t - A\mu_{t|0:t} - Bu_t)(AX_{t|0:t} + Bu_t + \epsilon_t - A\mu_{t|0:t} - Bu_t)^T\right] \\
&= A\mathbb{E}\left[(X_{t|0:t} - \mu_{t|0:t})(X_{t|0:t} - \mu_{t|0:t})^T\right] A^T + Q_t \\
&= A\Sigma_{t|0:t} A^T + Q_t
\end{aligned}$$

Cross Covariance

$$\Sigma_{t,t+1|0:t} = \mathbb{E}\left[(X_{t|0:t} - \mu_{t|0:t})(X_{t+1|0:t} - \mu_{t+1|0:t})^T\right]$$

$$\Sigma_{t,t+1|0:t} = \Sigma_{t|0:t} A^T$$

Unscented KF Measurement Step

- More tricky because now C/H is not known! How to compute Kalman gain?

$$K_{t+1} = \Sigma_{t+1|0:t} C^T (C \Sigma_{t+1|0:t} C^T + R_{t+1})^{-1}$$

Cross covariance under forward transform

Covariance under forward transform

Send sigma points through non-linear measurement model $\bar{\psi}^i = h(x_t)$

$$\bar{z} = \sum_{i=0}^{2n} w_m^i \bar{\psi}^i \qquad S = \sum_{i=0}^{2n} w_c^i (\bar{\psi}^i - \bar{z})(\bar{\psi}^i - \bar{z})^T \qquad T = \sum_{i=0}^{2n} w_c^i (\psi^i - \mu_{t+1|0:t})(\bar{\psi}^i - \bar{z})^T$$

$$K_{t+1} = T S^{-1}$$

Then use standard KF measurement update

Cross covariance

Covariance

UKF Pseudocode

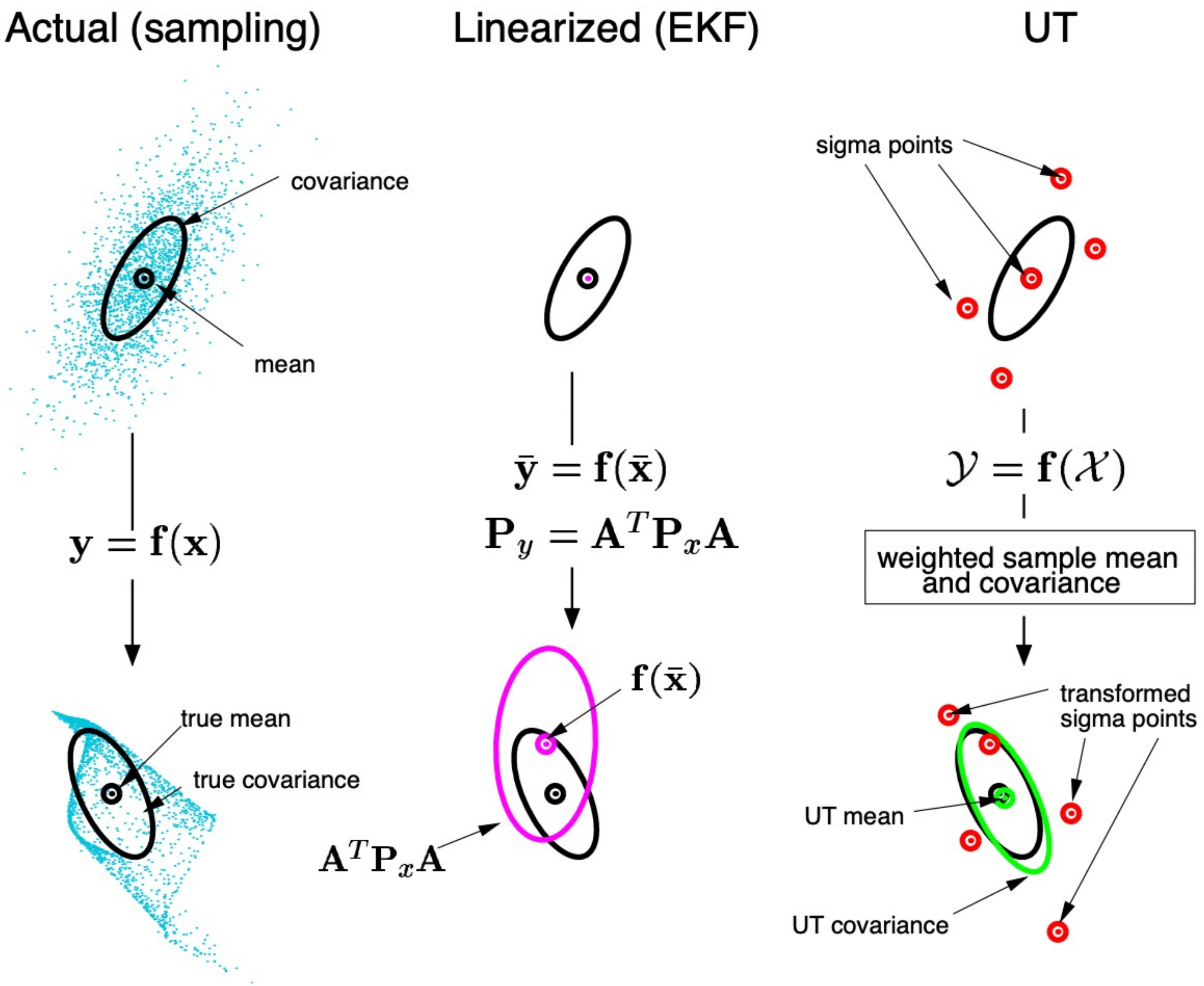
def Unscented_Kalman_filter($\mu_{t|0:t}$, $\Sigma_{t|0:t}$, u_t, z_{t+1}):

1. Dynamics
   1. Sample Sigma Points from $\mathcal{N}(\mu_{t|0:t}, \Sigma_{t|0:t})$
   2. Send them through $g(x_t, u_t)$
   3. Compute $\mu_{t+1|0:t}, \Sigma_{t+1|0:t}$ via UT
2. Measurement:
   1. Sample Sigma Points from $\mathcal{N}(\mu_{t+1|0:t}, \Sigma_{t+1|0:t})$
   2. Send them through $h(x_t)$
   3. Compute T, S as cross covariance and covariance
   4. Compute $K_{t+1} = TS^{-1}$
   5. Use standard KF updates
3. Return mean, cov ,

$$x_{t+1} = g(x_t, u_t) + \epsilon_t$$
$$z_t = h(x_t) + \delta_t$$
$$\epsilon_t \sim \mathcal{N}(0, Q)$$
$$\delta_t \sim \mathcal{N}(0, R)$$

Reminder of the model

How well does this do?



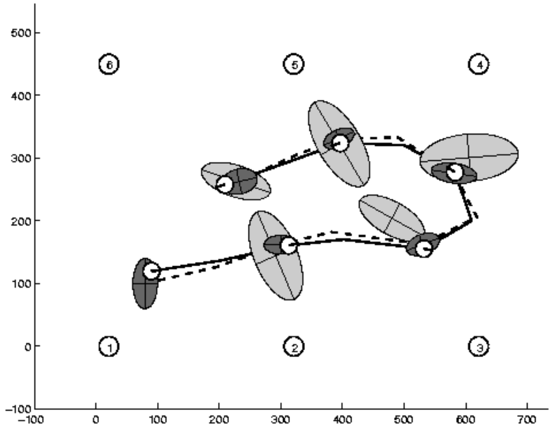
When/why is the UKF better than the EKF?

- EKF:
  - First linearize then propagate
  - Misses higher order terms

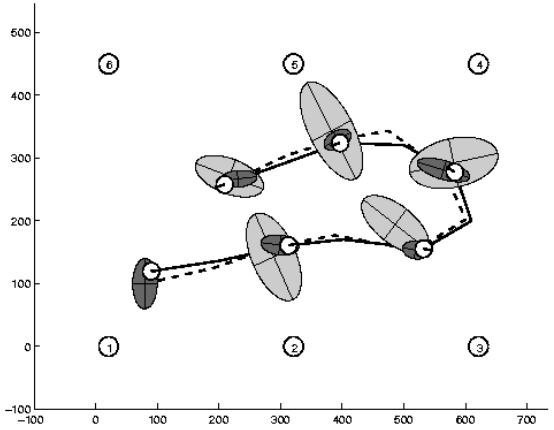
- UKF:
  - First propagate then linearize
  - Approximates the higher order terms as well

Approximately the same if sigma points are close to linearization point

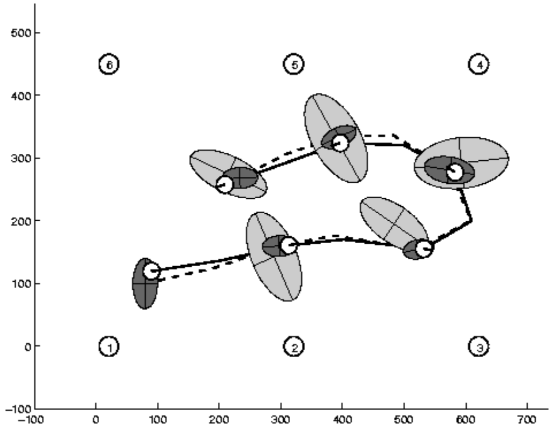
Estimation Sequence



EKF

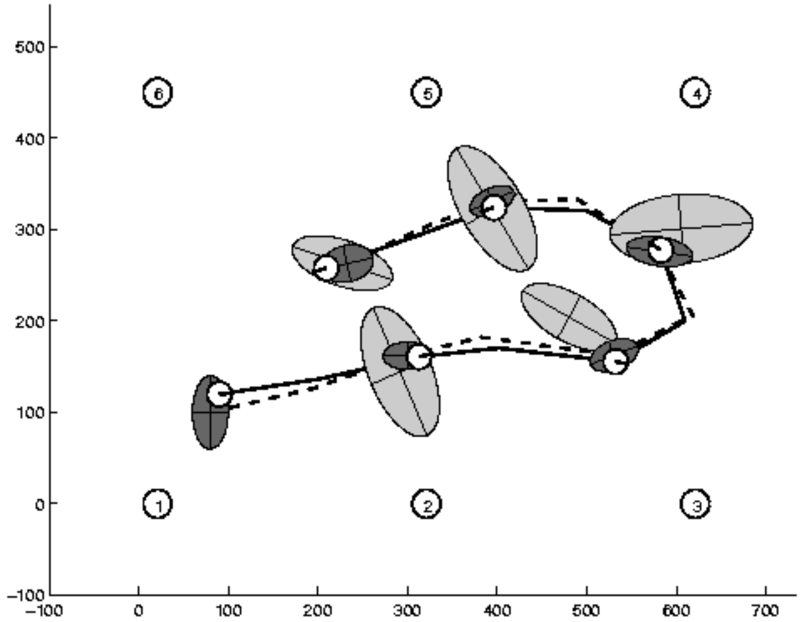


PF

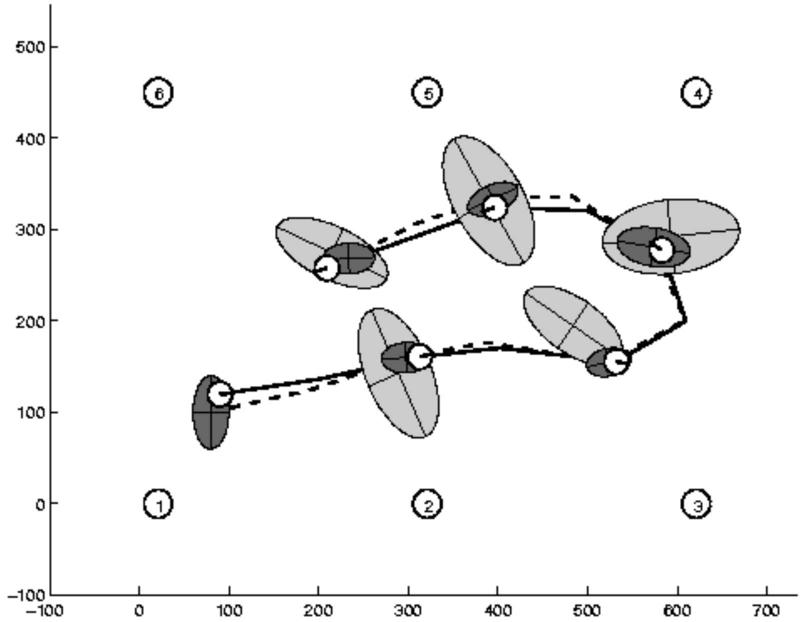


UKF

Estimation Sequence

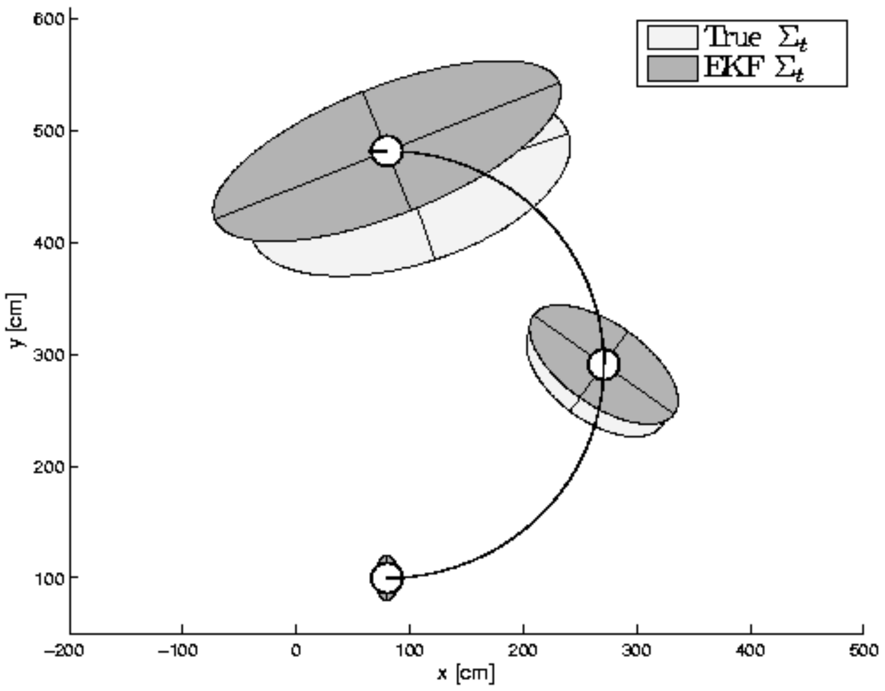


EKF

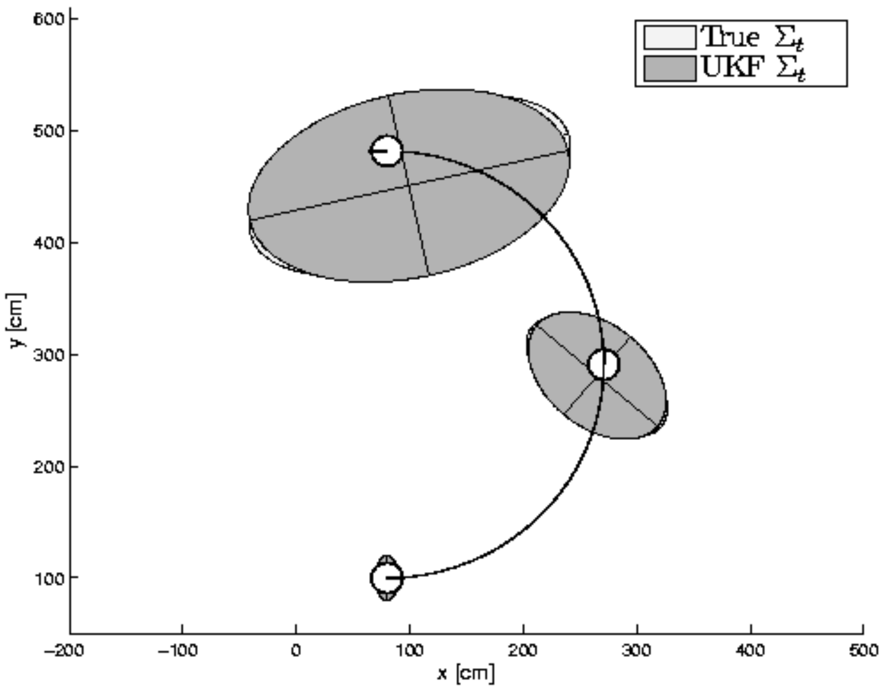


UKF

Prediction Quality

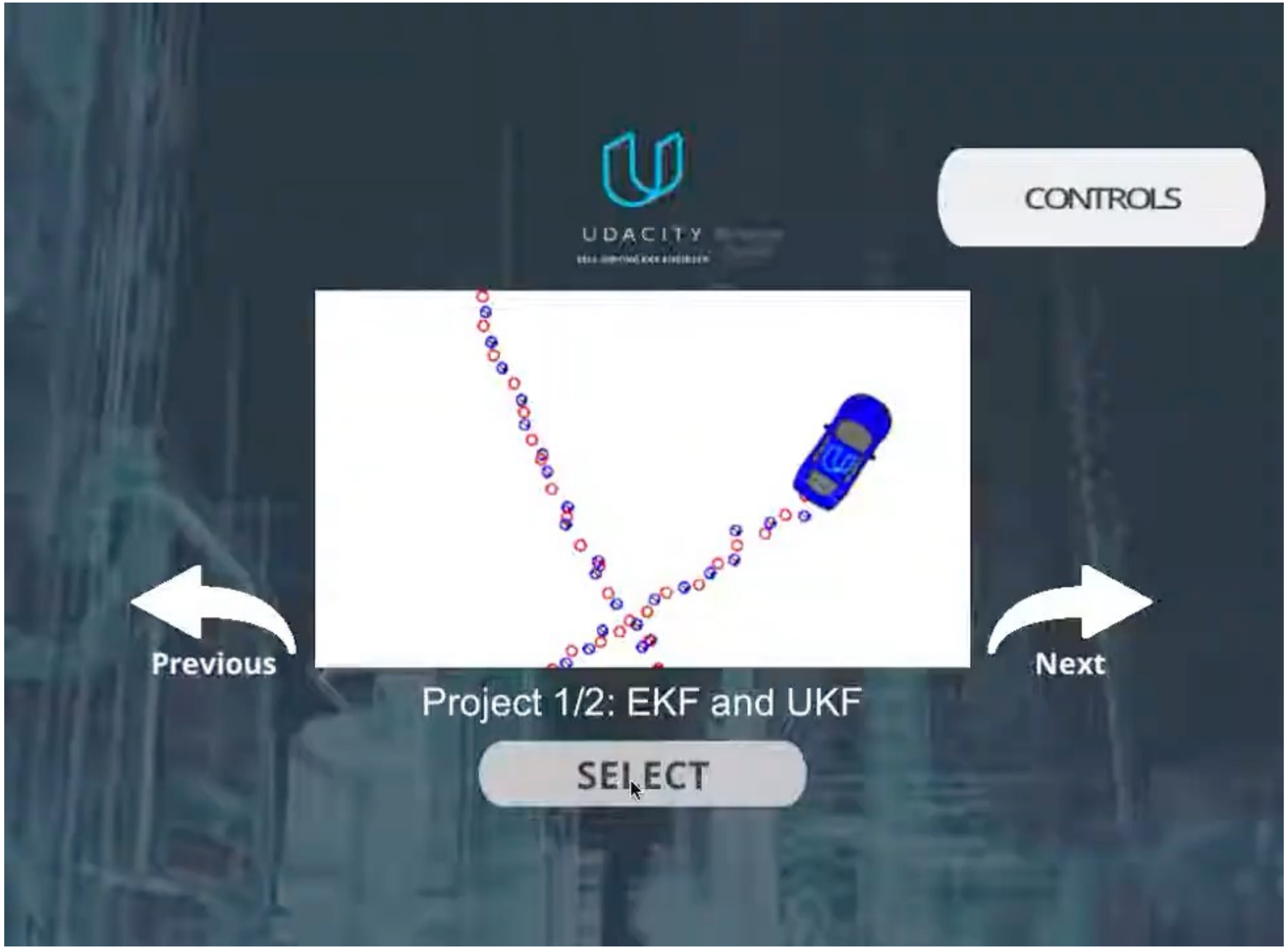


EKF



UKF

UKF in Action



UKF Summary

- Highly efficient: Same complexity as EKF, with a constant factor slower in typical practical applications
- Better linearization than EKF: Accurate in first two terms of Taylor expansion (EKF only first term)
- Derivative-free: No Jacobians needed
- Still not optimal!

Lecture Outline

Kalman Filtering

↓

Extended Kalman Filter

↓

Unscented Kalman Filter