Robotics

Estimation and Learning
with Dan Lee

Week 2.
Kalman Filter

2.1 Kalman Filter Model
2.2 Maximum-A-Posterior Estimation
2.3 Nonlinear Variations

Penn Engineering

Week 2.
Kalman Filter

2.1. Kalman Filter: Motivation

Intuition behind KF

- Dynamics

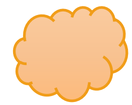
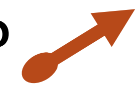
Intuition behind KF

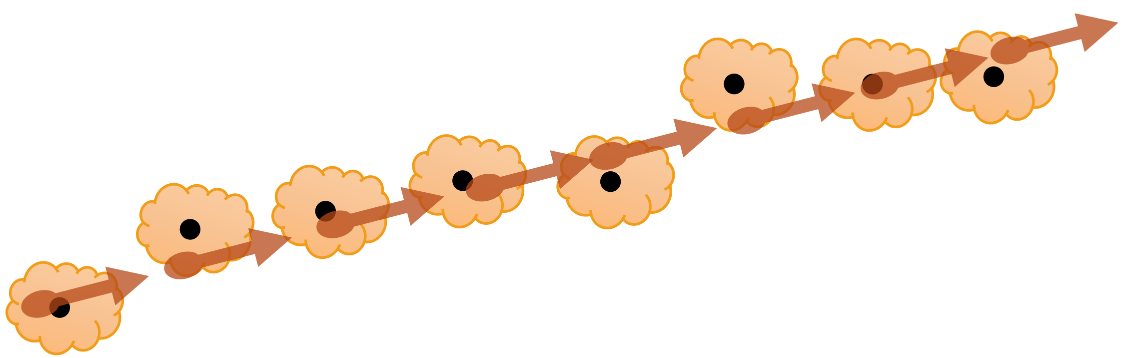
- Dynamics

Application

- Track a moving target
  - Soccer ball

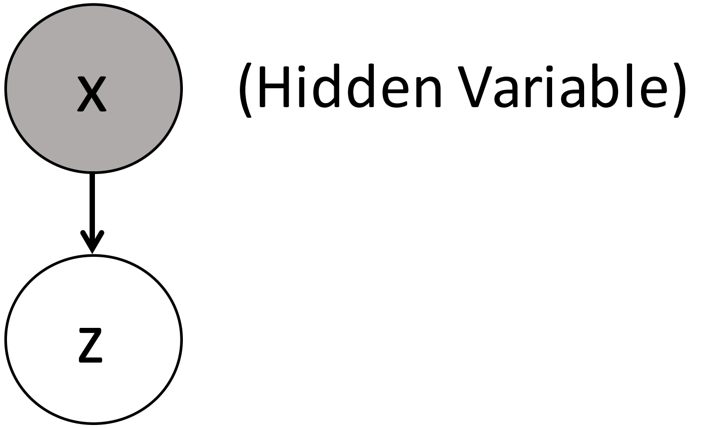
Intuition behind KF

- Multiple measurements: ●, ●, ●, ...
- Each measurement is *noisy*: 
- What is the *true* state of the object? 



State and Measurement

- State (x): any quantity of interest
- Measurement (z): what we observe



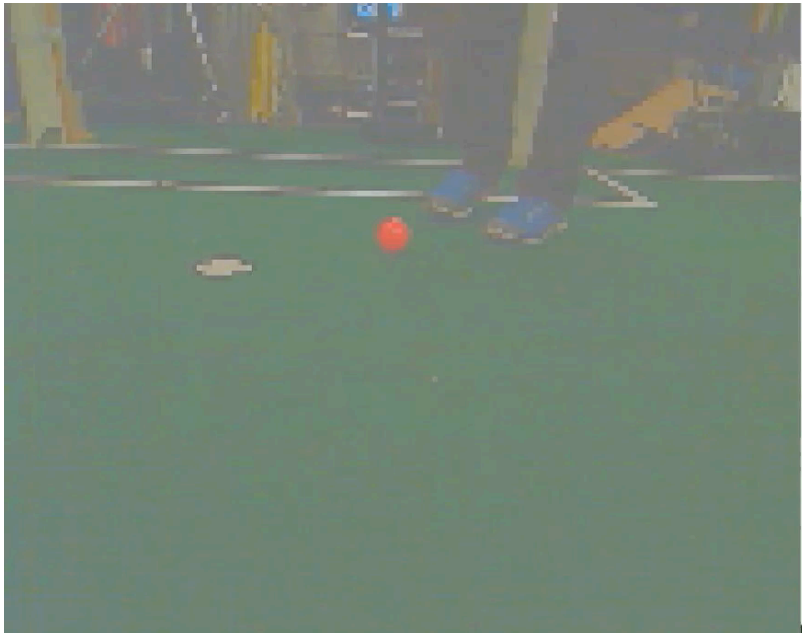
State

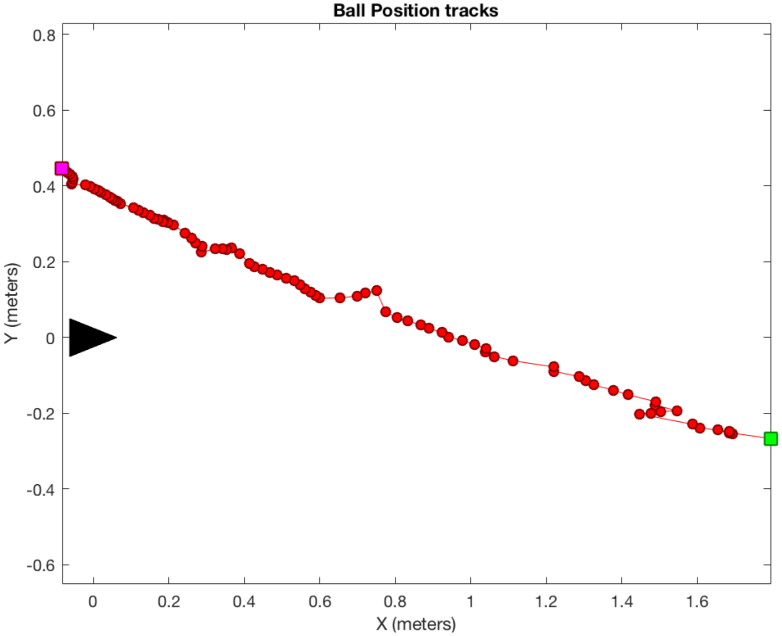
- Example: “What characterizes the state of a ball?”
  - Position, Velocity, Acceleration
  - Rotation
  - Color
  - Size
  - Weight
  - Temperature
  - Elasticity
  - …

Measurement

- Example: What do we observe or measure?
  - Distance
  - Angle
  - Inertia change
  - Color
  - …

Measurement





Robotics

Estimation and Learning
with Dan Lee

Week 2.
Kalman Filter

2.2 System and Measurement Models

Penn Engineering

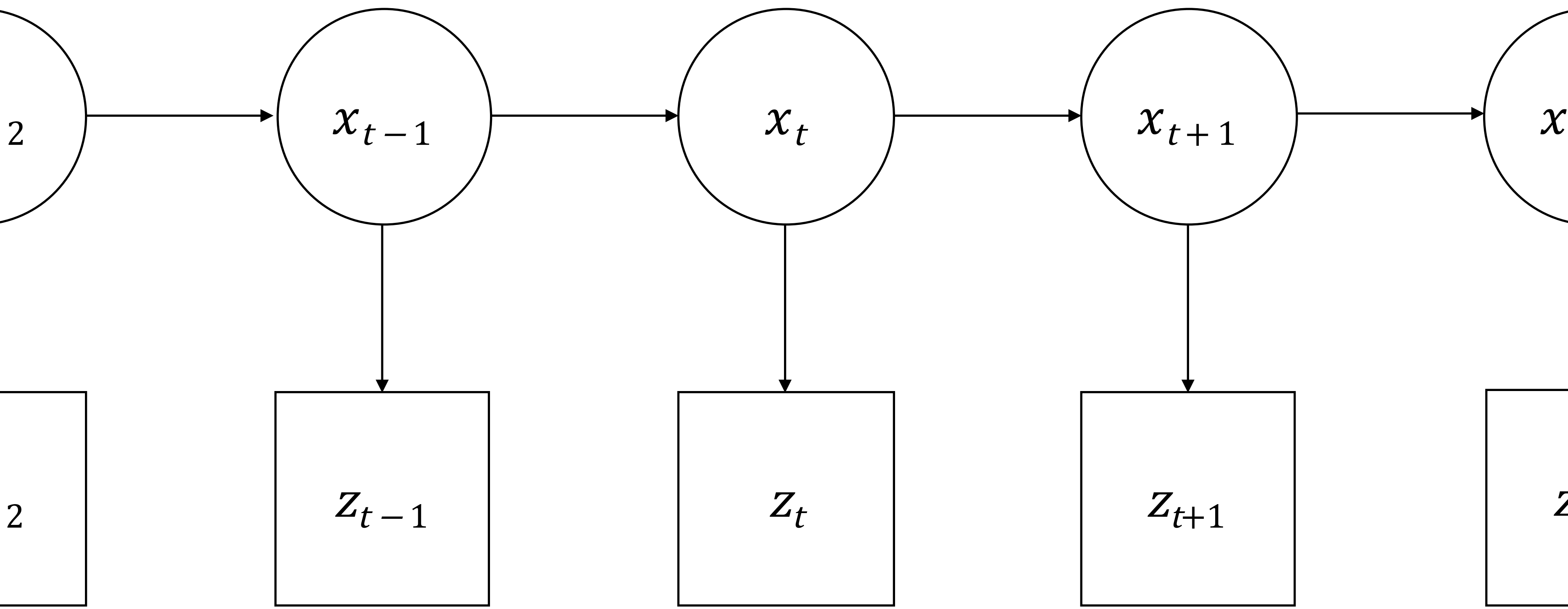
Bayesian Kalman Filtering

- Modeling motion and noise
- Mathematical underpinnings of Kalman filters
- Position tracking example

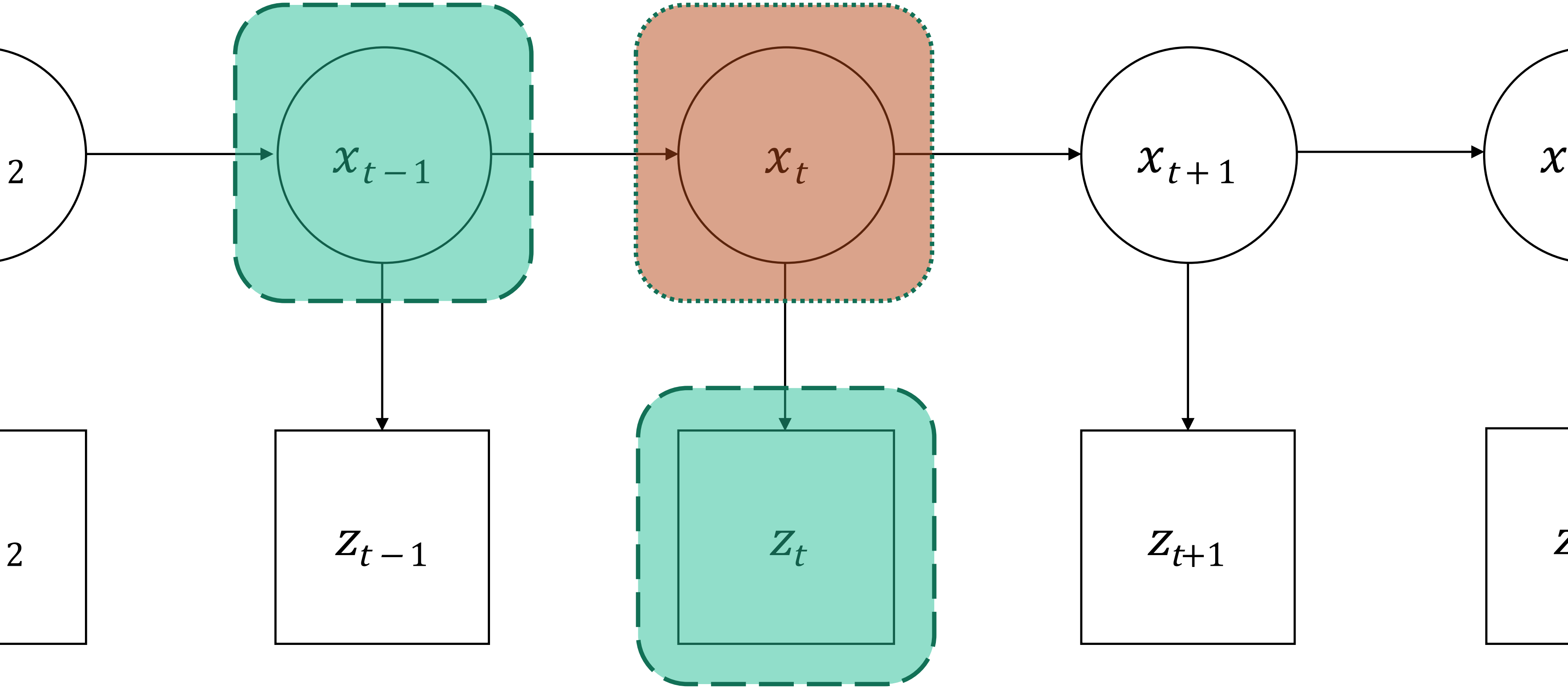
Linear Modeling

- Discrete Linear dynamical system of motion
  - $x_{t+1} = A\, x_t + B\, u_t$ $\qquad\qquad$ $z_t = C\, x_t$
- Simple state vector, x, is position and velocity
  - $x_{t+1} := [\, v \quad {}^{dv}/_{dt} \,]$
- Description of Dynamics
  - $A = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix}$

KF: Bayesian filtering



KF: Bayesian filtering



- Find x_t
- Know x_{t-1}
- Know z_t

Bayesian modeling

- Prediction using state dynamics model

$$p(x_{t+1}|x_t)$$

- Inference from noisy measurements

$$p(z_t|x_t)$$

- Model $\mathrm{x_t}$ with a Gaussian (mean and covariance)

$$p(x_t) = \mathcal{N}(x_t, P_t)$$

Bayesian filtering

- Apply linear dynamics

$$p(x_{t+1}|x_t) = Ap(x_t)$$
$$p(z_t|x_t) = Cp(x_t)$$

- Add noise for motion and observations

$$p(x_{t+1}|x_t) = Ap(x_t) + \upsilon_m$$
$$p(z_t|x_t) = Cp(x_t) + \upsilon_o$$

- Introduce Gaussian model of x_t

$$p(x_{t+1}|x_t) = A\mathcal{N}(x_t, P_t) + \mathcal{N}(0, \Sigma_m)$$
$$p(z_t|x_t) = C\mathcal{N}(x_t, P_t) + \mathcal{N}(0, \Sigma_o)$$

Bayesian filtering

- Consolidate expression using special properties

$$p(x_{t+1}|x_t) = A\mathcal{N}(x_t, P_t) + \mathcal{N}(0, \Sigma_m)$$

$$p(z_t|x_t) = C\mathcal{N}(x_t, P_t) + \mathcal{N}(0, \Sigma_o)$$

- Apply *linear* transform to Gaussian distributions

$$p(x_{t+1}|x_t) = \mathcal{N}(Ax_t, AP_tA^T) + \mathcal{N}(0, \Sigma_m)$$

$$p(z_t|x_t) = \mathcal{N}(Cx_t, CP_tC^T) + \mathcal{N}(0, \Sigma_o)$$

- Apply summation

$$p(x_{t+1}|x_t) = \mathcal{N}(Ax_t, AP_tA^T + \Sigma_m)$$

$$p(z_t|x_t) = \mathcal{N}(Cx_t, CP_tC^T + \Sigma_o)$$

Bayesian filtering

- Consolidate expression using special properties

$$p(x_{t+1}|x_t) = A\mathcal{N}(x_t, P_t) + \mathcal{N}(0, \Sigma_m)$$

$$p(z_t|x_t) = C\mathcal{N}(x_t, P_t) + \mathcal{N}(0, \Sigma_o)$$

- Apply *linear* transform to Gaussian distributions

$$p(x_{t+1}|x_t) = \mathcal{N}(Ax_t, AP_tA^T) + \mathcal{N}(0, \Sigma_m)$$

$$p(z_t|x_t) = \mathcal{N}(Cx_t, CP_tC^T) + \mathcal{N}(0, \Sigma_o)$$

- Apply summation

$$p(x_{t+1}|x_t) = \mathcal{N}(Ax_t, AP_tA^T + \Sigma_m)$$

$$p(z_t|x_t) = \mathcal{N}(Cx_t, CP_tC^T + \Sigma_o)$$

Robotics

Estimation and Learning with Dan Lee

Week 2. Kalman Filter

2.3 Maximum-A-Posterior Estimation

Penn Engineering

Bayesian Kalman Filtering

- Apply the MAP to Bayes' Rule
- Solve the maximization
- Establish Kalman Filter update method

Bayesian filtering

- Bayes' Rule

$$p(\alpha|\beta) = \frac{P(\beta|\alpha)P(\alpha)}{P(\beta)}$$

- Given from Kalman model:

$$p(x_t|x_{t-1}) = \mathcal{N}(Ax_{t-1}, AP_{t-1}A^T + \Sigma_m)$$

$$p(z_t|x_t) = \mathcal{N}(Cx_t, \Sigma_o)$$

Bayesian filtering

Note :

The equation:

$$p(z_t|x_t) = N(Cx_t, \sum_0)$$

should actually be :

$$p(z_t|x_t) = N(Cx_t, CP_tC^T + \sum_0)$$

as you will observe in the following slides.

- Apply Bayes' Rule $p(\alpha|\beta) = \dfrac{P(\beta|\alpha)P(\alpha)}{P(\beta)}$

$$p(x_t|x_{t-1}) = \mathcal{N}(Ax_{t-1}, AP_{t-1}A^T + \Sigma_m) \rightarrow \underbrace{\boxed{\alpha}}_{\text{Prior}}$$

$$p(z_t|x_t) = \mathcal{N}(Cx_t, \Sigma_o) \rightarrow \underbrace{\boxed{\beta|\alpha}}_{\text{Likelihood}}$$

$$\boxed{\underbrace{\boxed{p(x_t|z_t, x_{t-1})}}_{\text{Posterior}} = \frac{p(z_t|x_t, x_{t-1})p(x_t|x_{t-1})}{P(z_t)}}$$

Bayesian filtering

- Posterior distribution is another Gaussian
- MAP Estimates “optimal” x_t value
- Use MAP estimates to form a new mean and variance for the state

Bayesian filtering

- Calculate the Maximum A Posteriori Estimate

$$\hat{x}_t = \underset{x_t}{\operatorname{argmax}}\, p(x_t|z_t, x_{t-1})$$

$$\hat{x}_t = \underset{x_t}{\operatorname{argmax}} \frac{p(z_t|x_t)p(x_t|x_{t-1})}{P(z_t)}$$

$$\hat{x}_t = \underset{x_t}{\operatorname{argmax}}\, p(z_t|x_t)p(x_t|x_{t-1})$$

$$\hat{x}_t = \underset{x_t}{\operatorname{argmax}}\, \mathcal{N}(Cx_t, CP_tC^T + \Sigma_o)\mathcal{N}(Ax_{t-1}, AP_{t-1}A^T + \Sigma_m)$$

Bayesian filtering

- Calculate the Maximum A Posteriori Estimate

$$\hat{x}_t = \underset{x_t}{\operatorname{argmax}} \mathcal{N}(Cx_t, CP_tC^T + \Sigma_o)\mathcal{N}(Ax_t, AP_{t-1}A^T + \Sigma_m)$$

- Simplify with these substitutions

$$P = P_t = AP_{t-1}A^T + \Sigma_m$$
$$R = CP_tC^T + \Sigma_o$$

- Simplify the exponential form of $\mathcal{N}$ via logarithms

$$\hat{x}_t = \underset{x_t}{\operatorname{argmin}} \begin{array}{c} (z_t - Cx_t)R^{-1}(z_t - Cx_t) \\ +(x_t - Ax_{t-1})P^{-1}(x_t - Ax_{t-1}) \end{array}$$

Bayesian filtering

- Solve optimization by setting the derivative to zero

$$\hat{x}_t = \underset{x_t}{\operatorname{argmin}} \begin{array}{c} (z_t - Cx_t)R^{-1}(z_t - Cx_t) \\ +(x_t - Ax_{t-1})P^{-1}(x_t - Ax_{t-1}) \end{array}$$

$$0 = \frac{d}{dx_t}\begin{pmatrix} (z_t - Cx_t)R^{-1}(z_t - Cx_t) \\ +(x_t - Ax_{t-1})P^{-1}(x_t - Ax_{t-1}) \end{pmatrix}$$

All values inside the argmin function of the form,

$AR^{-1}A$ are actually of the form $AR^{-1}A^T$

where A = $(z_t - Cx_t)$, $(x_t - Ax_{t-1})$

Bayesian filtering

- Collect terms in the derivative

$$(C^T R^{-1} C + P^{-1}) x_t = z_t^T R^{-1} C + P^{-1} A x_{t-1}$$

$$x_t = (C^T R^{-1} C + P^{-1})^{-1} (z_t^T R^{-1} C + P^{-1} A x_{t-1})$$

- Apply the *Matrix Inversion Lemma*

$$(C^T R^{-1} C + P^{-1})^{-1} = P - P C^T (R + C P C^T)^{-1} C P$$

- Define *Kalman Gain*: $K = P C^T (R + C P C^T)^{-1}$

Bayesian filtering

- Expand the terms

$$x_t = (C^T R^{-1} C + P^{-1})^{-1} (C^T R^{-1} y_t + P^{-1} A x_{t-1})$$

$$x_t = (P - KCP)(C^T R^{-1} z_t + P^{-1} A x_{t-1})$$

$$x_t = A x_{t-1} + P C^T R^{-1} z_t - KCA x_{t-1} - KCPC^T R^{-1} y_t$$

$$x_t = A x_{t-1} - KCA x_{t-1} + (P C^T R^{-1} - KCPC^T R^{-1}) y_t$$

$$x_t = A x_{t-1} - KCA x_{t-1} + K y_t$$

$$\boxed{\hat{x}_t = A x_{t-1} + K(z_t - CA x_{t-1})}$$

- Convince yourself that $K = P C^T R^{-1} - KCPC^T R^{-1}$

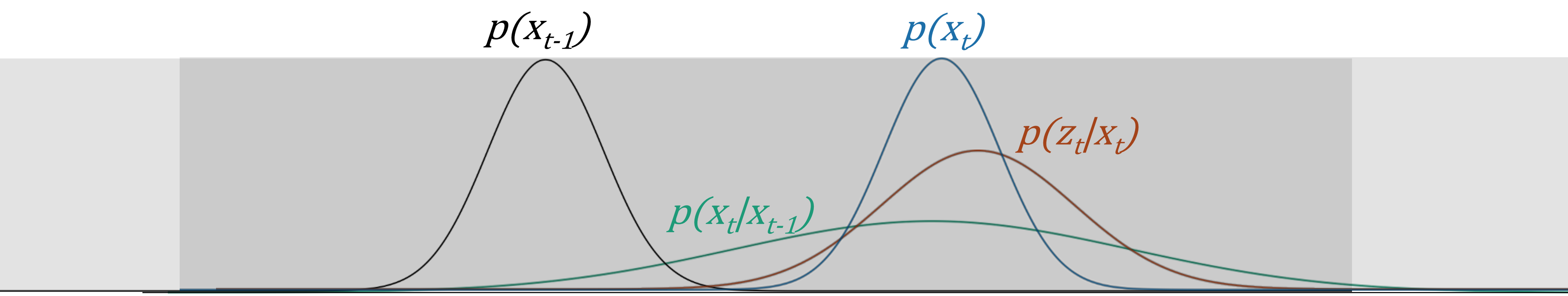
Bayesian filtering

- Must update the covariance of the state

$$\boxed{\hat{P}_t = P - KCP}$$

1D Visualization

- The position of x is moving forward
  - Uncertain motion model increases the spread
- We observe a noisy position estimate, z_t
- The corrected position has less spread than both the observation and motion adjusted state



Robotics

Estimation and Learning
with Dan Lee

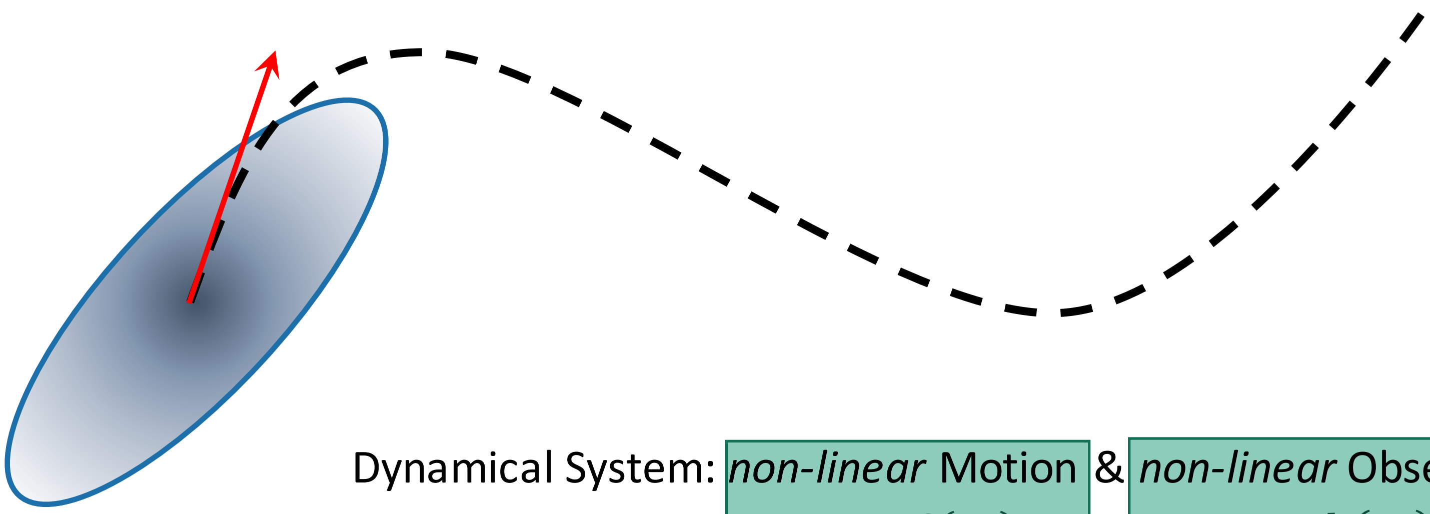
Week 2.
Kalman Filter

2.4 Non-linear Kalman filters

Penn Engineering

Extended Kalman Filter

- Linearize around the transition function
  - Jacobian represents the derivative in matrix form
- Calculate the uncertainty update via linearization



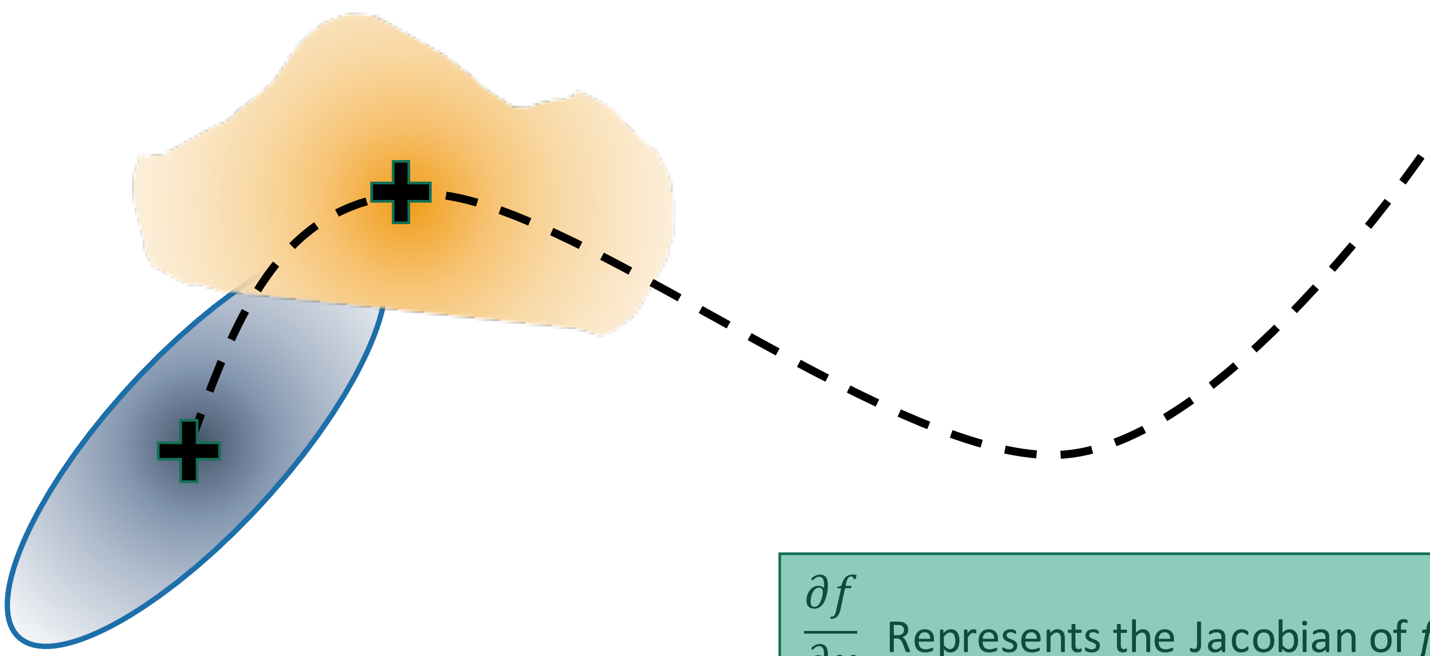
Dynamical System: *non-linear* Motion & *non-linear* Observation

$x_{t+1} = f(x_t)$ $z_{t+1} = h(x_t)$

Extended Kalman Filter

- Covariance prediction $p(x_{t+1}|x_t) = \mathcal{N}(Ax_t, AP_tA^T + \Sigma_m)$

$$p(x_{t+1}|x_t) = \mathcal{N}\left(f(x_t), \frac{\partial f}{\partial x} P_t \frac{\partial f}{\partial x}^T + \Sigma_m\right)$$



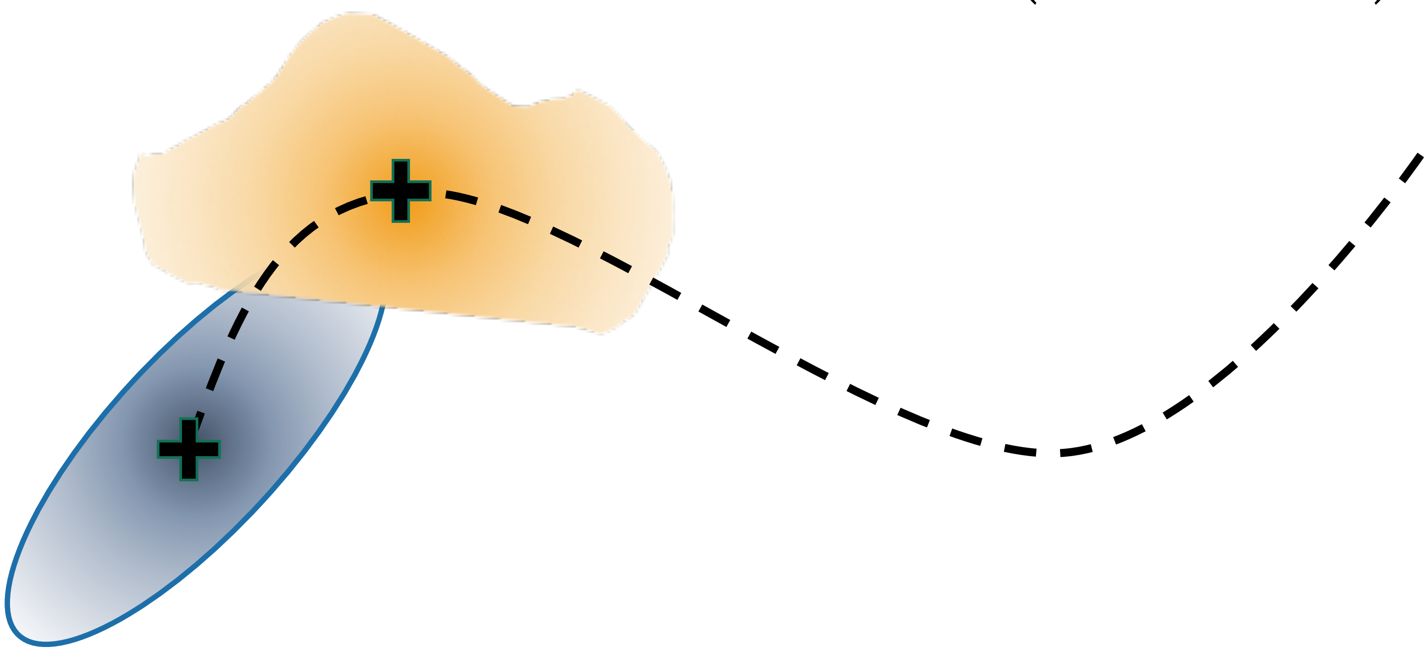
$\frac{\partial f}{\partial x}$ Represents the Jacobian of f

Extended Kalman Filter

- Kalman Gain

$$K = PC^T(\Sigma_o + CPC^T)^{-1}$$

$$K = P\frac{\partial h^T}{\partial x}\left(\Sigma_o + \frac{\partial h}{\partial x}P_t\frac{\partial h^T}{\partial x}\right)^{-1}$$

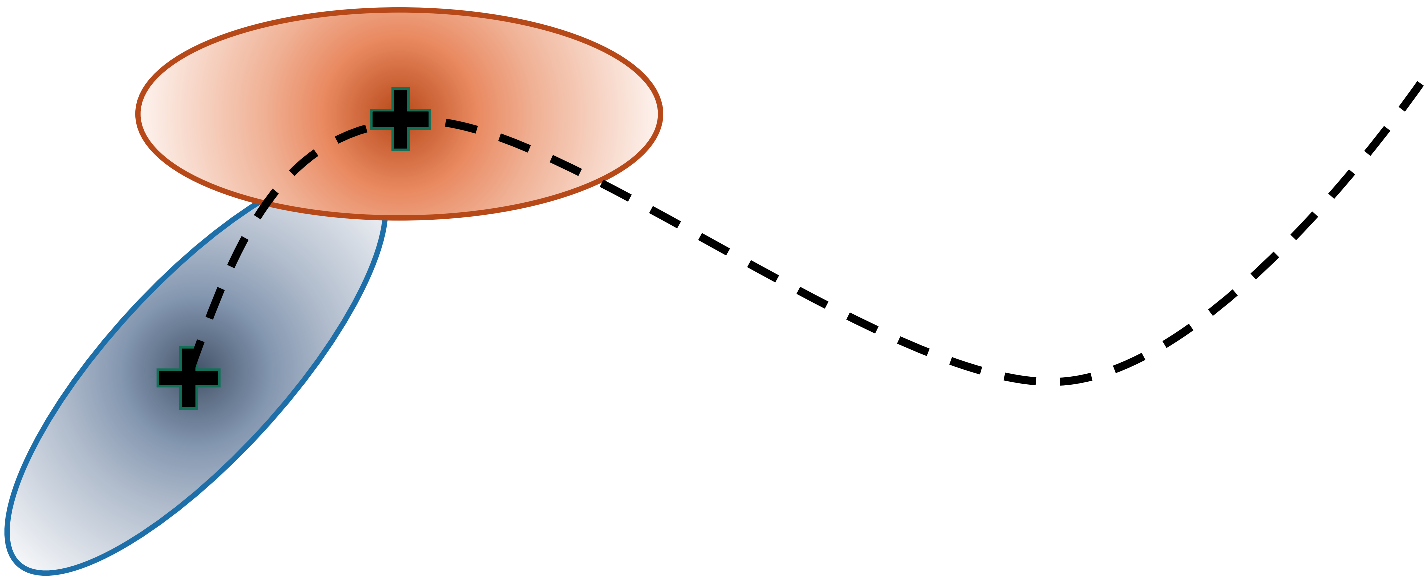


Extended Kalman Filter

- Overall update

$$\hat{x}_t = f(x_{t-1}) + K(z_t - h(f(x_t)))$$

$$\hat{P}_t = P - K\frac{\partial h}{\partial x}P$$

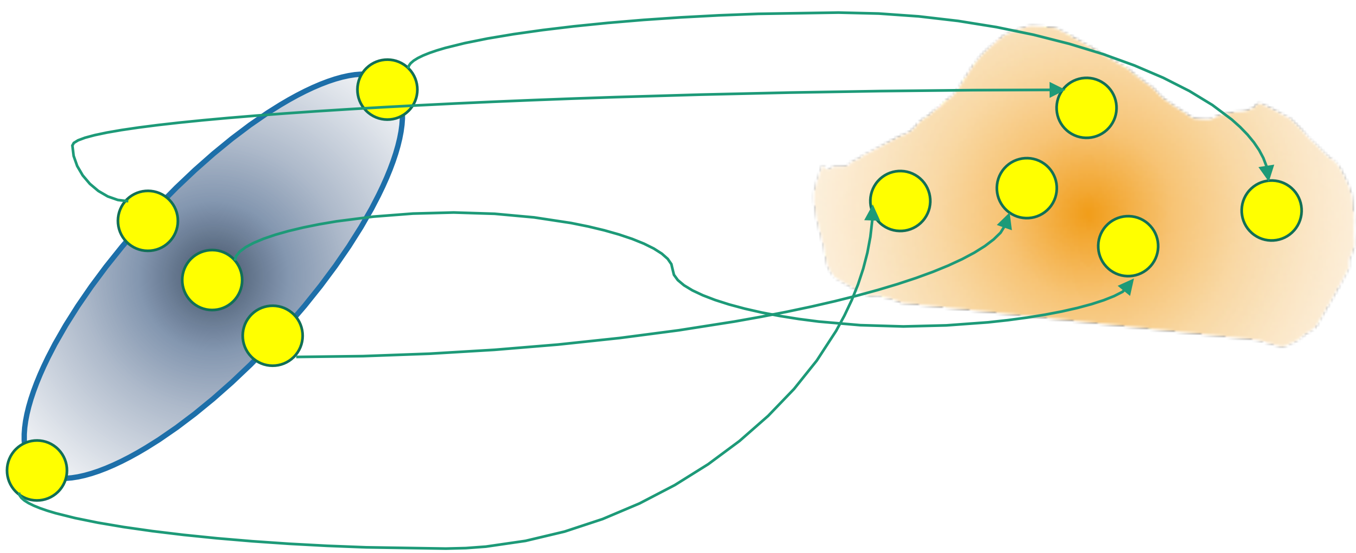


Unscented Kalman Filter

- Noise distribution characterized by a set of points
- Transform these points in non-linear fashion
- Re-estimate mean and covariance for recalculating Gaussian distribution to characterize uncertainty

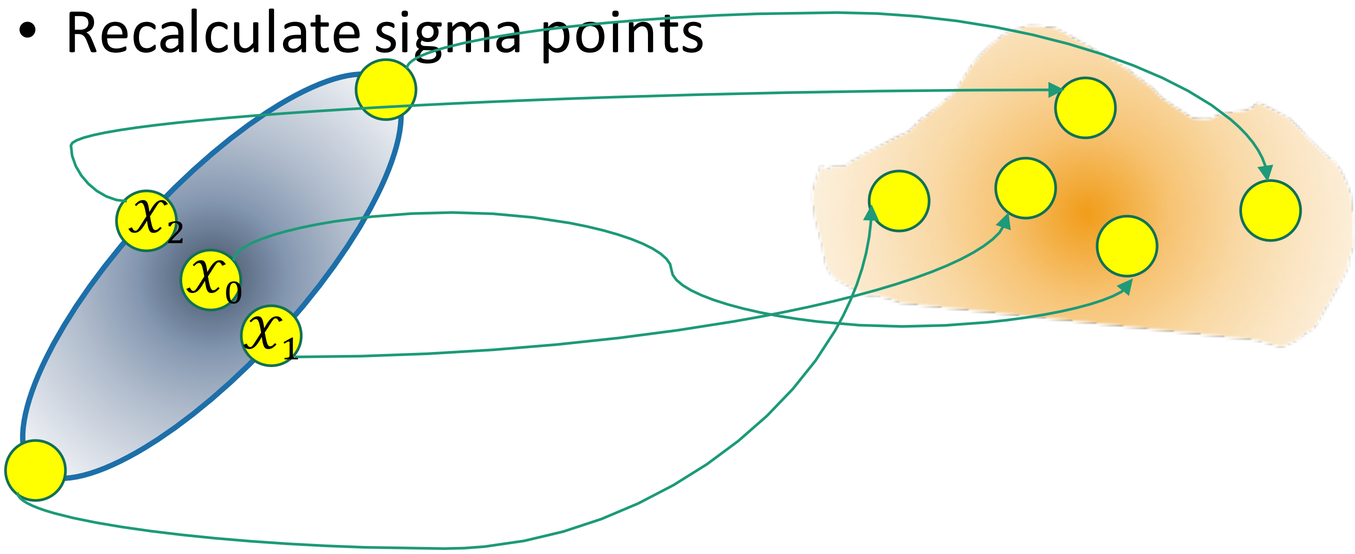
Unscented Kalman Filter

- Model the distribution based on a set of *sigma points*
    - Statistics of these points captures the distribution



Unscented Kalman Filter

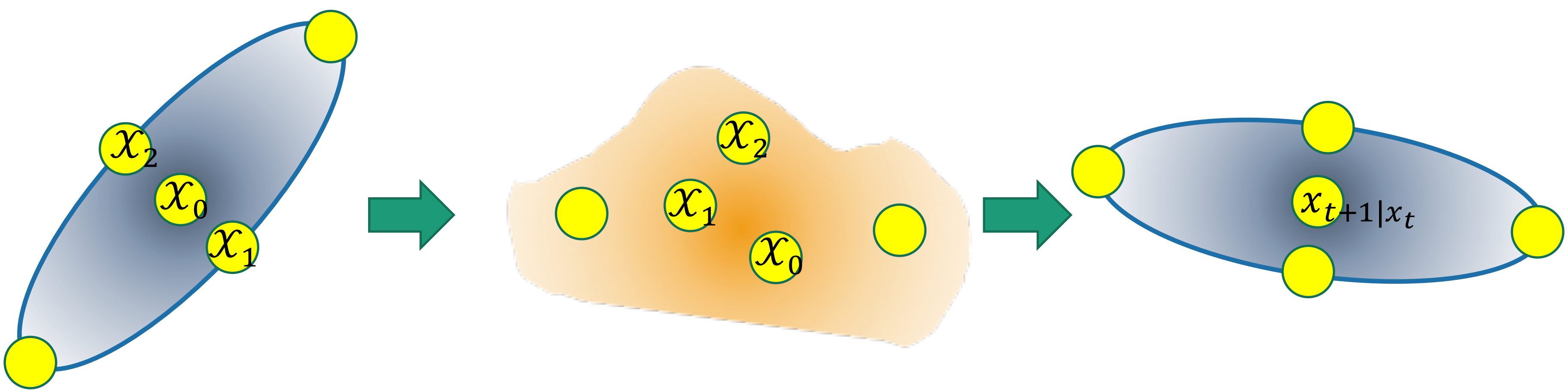
- Approximate new distribution by computing the statistics of the *sigma points* $\mathcal{X}_i$
- Approximate a Gaussian with first two moments
  - Recalculate sigma points



Unscented Kalman Filter

- State prediction
  - Mean of Sigma points run through dynamic system

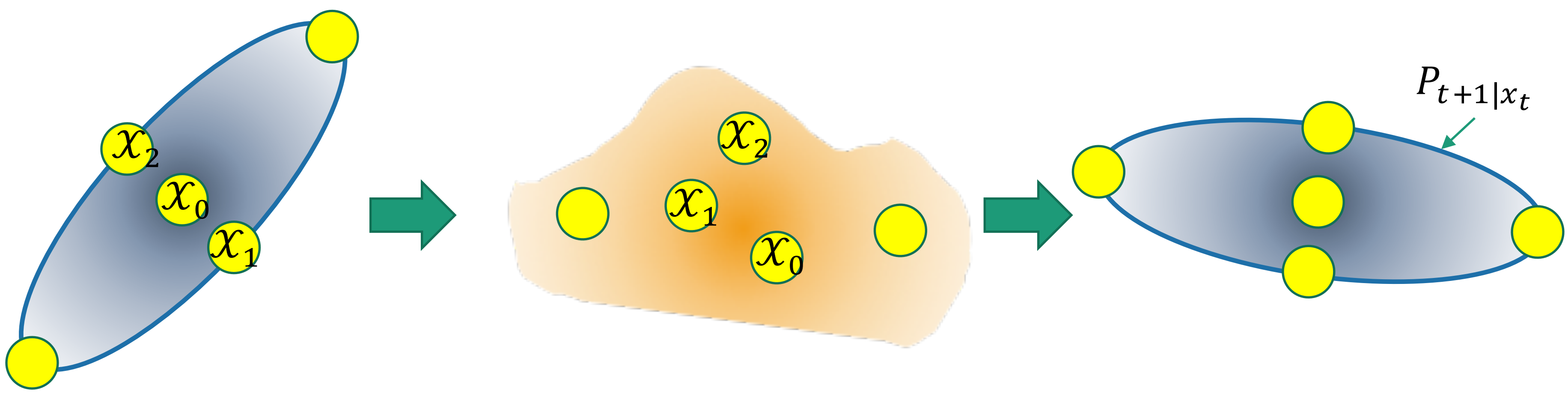
$$x_{t+1|x_t} = \frac{1}{N_{\mathcal{X}}} \sum_i f(\mathcal{X}_i)$$



Unscented Kalman Filter

- Uncertainty prediction
  - Covariance of sigma points run through dynamic system

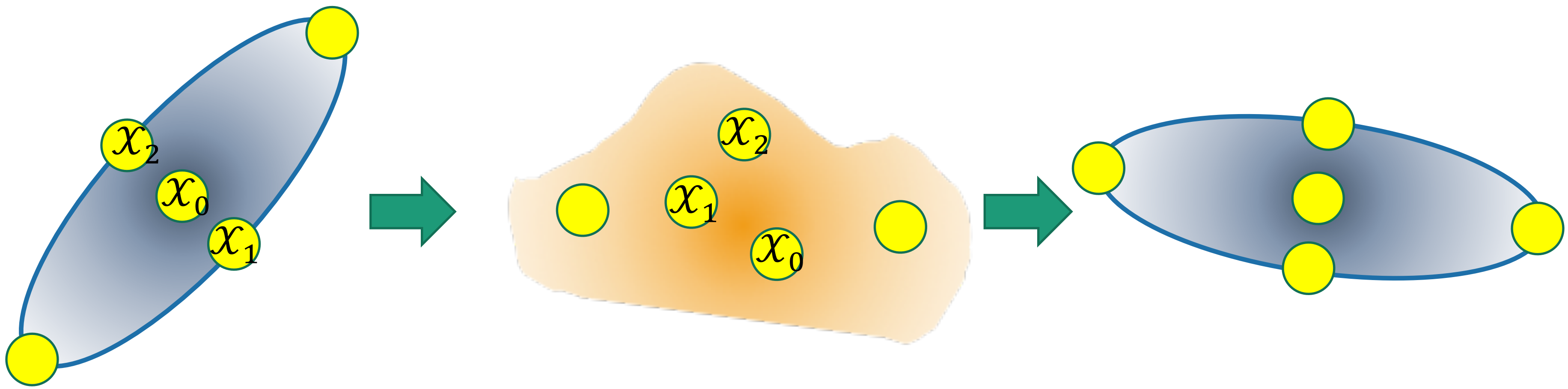
$$P_{t+1|x_t} = \frac{1}{N_\chi} \sum_i \left(f(\mathcal{X}_i) - x_{t+1|x_t}\right)\left(f(\mathcal{X}_i) - x_{t+1|x_t}\right)^T$$



Unscented Kalman Filter

- Expected Observation
  - Mean of Sigma points' expected obsrvation

$$z_{t+1|x_t} = \frac{1}{N_\chi} \sum_i h(f(\mathcal{X}_i))$$

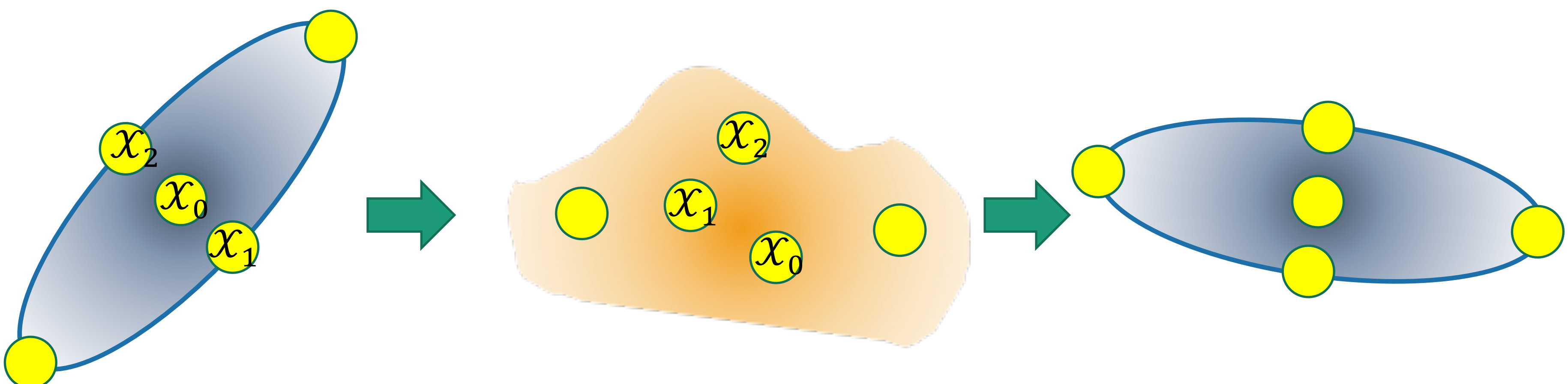


Unscented Kalman Filter

Recall Linear System
$K = PC^T(\Sigma_o + CPC^T)^{-1}$

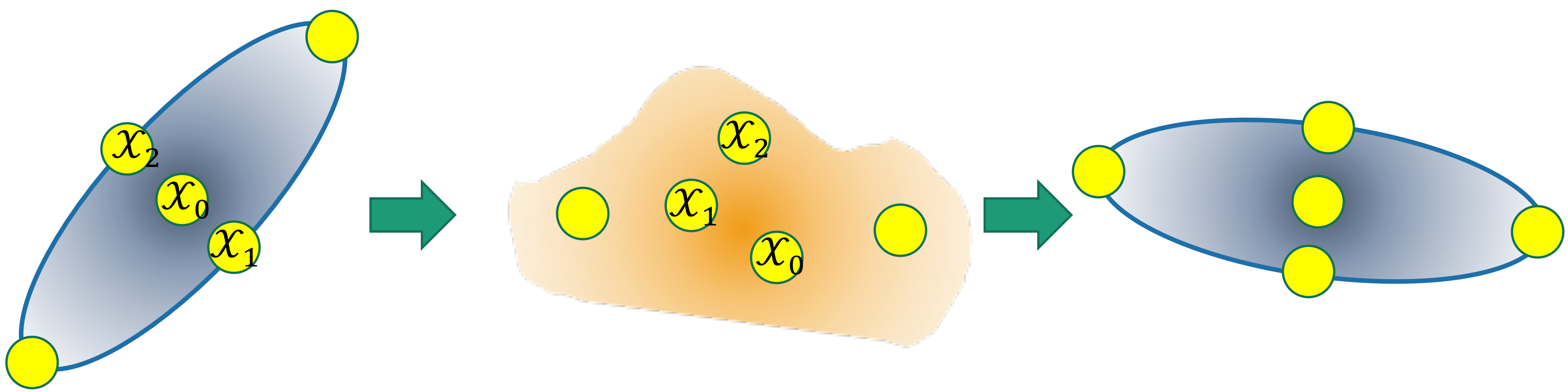
- Kalman Gain
  - Utilize sigma points, not observation model

$$K = \frac{1}{N_\chi}\sum_i \left(f(\mathcal{X}_i) - x_{t+1|x_t}\right)\left(h\left(f(\mathcal{X}_i)\right) - z_{t+1|x_t}\right)^T$$
$$\cdot\left(\frac{1}{N_\chi}\sum_i \left(h\left(f(\mathcal{X}_i)\right) - z_{t+1|x_t}\right)\left(h\left(f(\mathcal{X}_i)\right) - z_{t+1|x_t}\right)^T\right)^{-1}$$



Unscented Kalman Filter

- Update just as the linear filter
  - The covariance update will be slightly different
- See notes for good resources on further details



Particle Filter

- Take the unscented filter to the limit, and use many points to characterize the distribution
- Distribution is not limited to Gaussian
- Discuss in Week 4