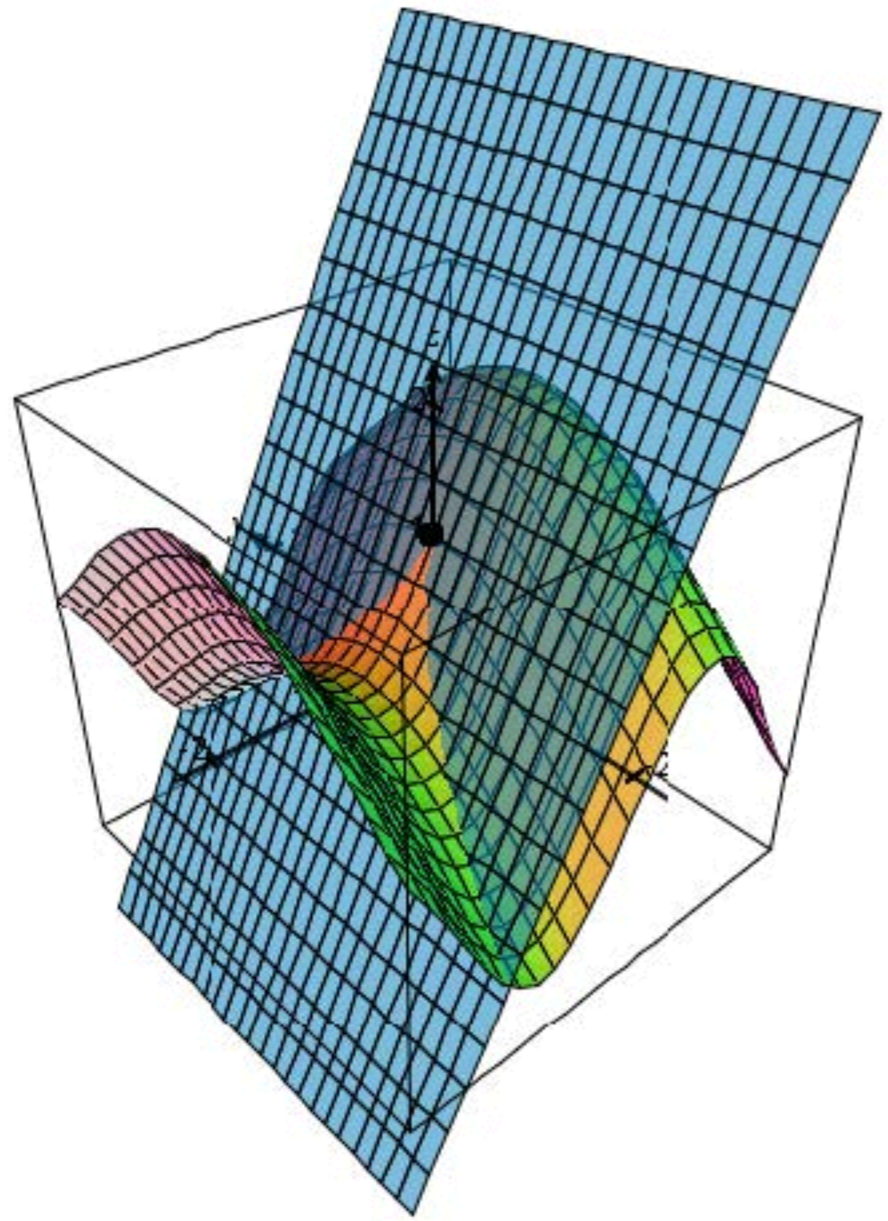


Extended Kalman Filter

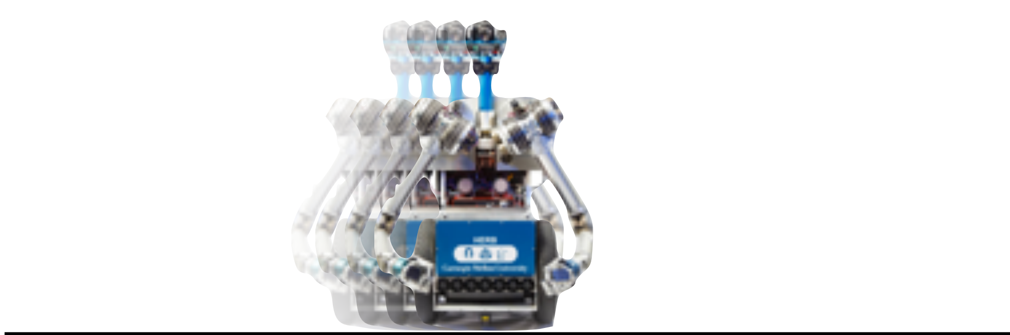
16-385 Computer Vision (Kris Kitani)

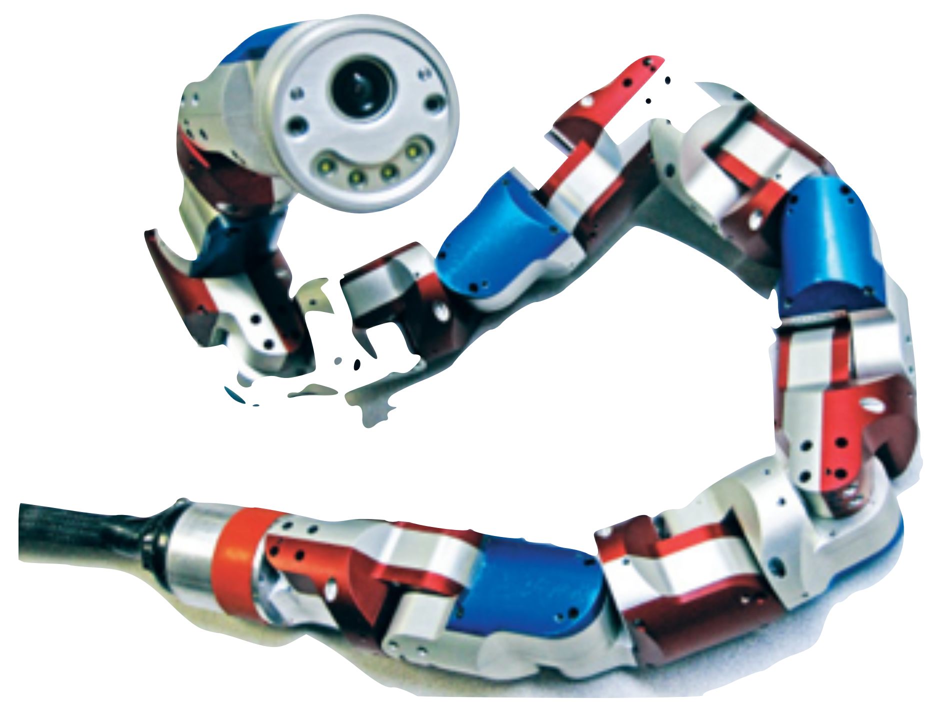
Carnegie Mellon University

Motion model of the Kalman filter is linear

$$x_t = Ax_{t-1} + Bu_t + \epsilon_t$$

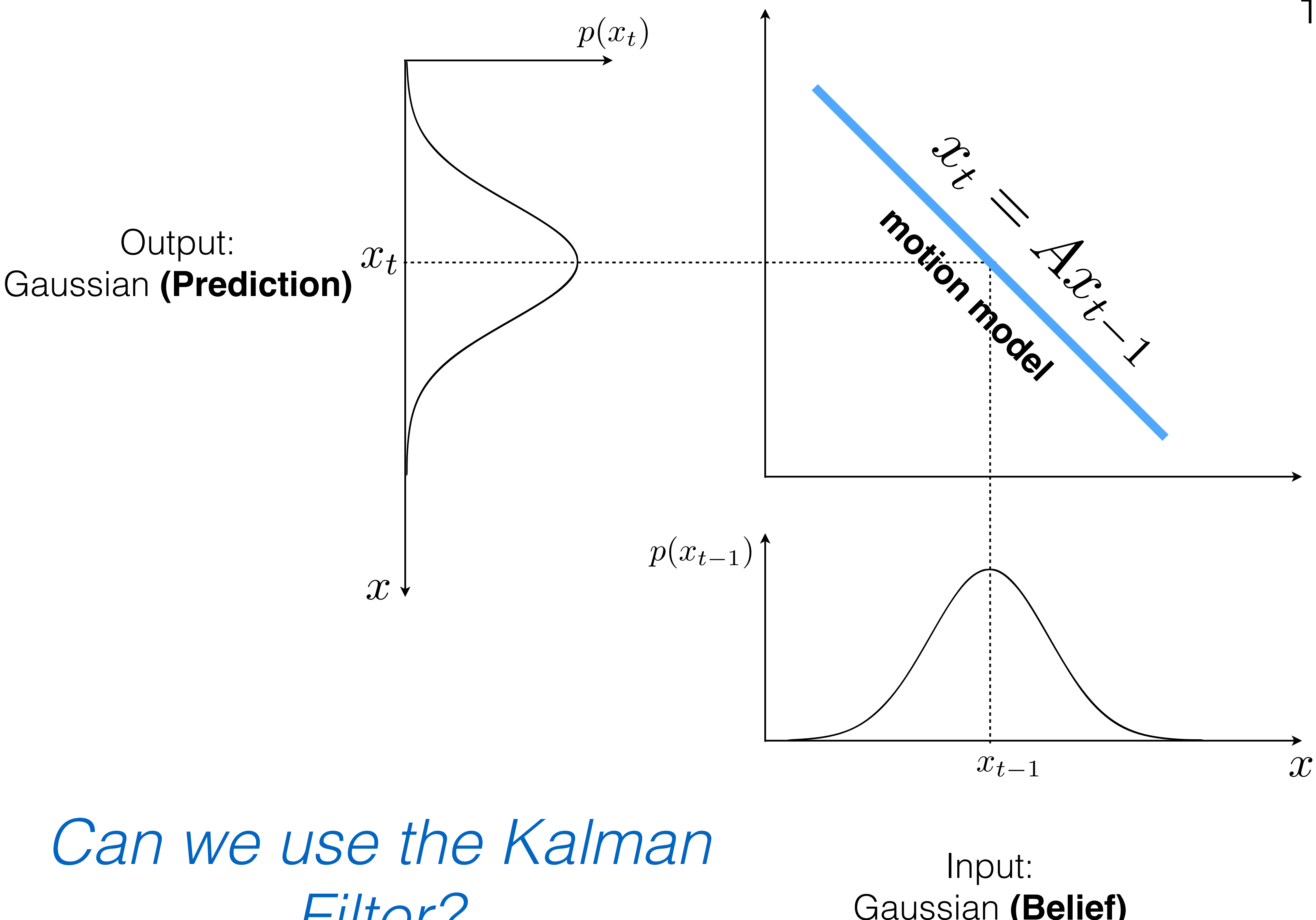
but motion is not always linear





Visualizing **linear** models

1D motion model example



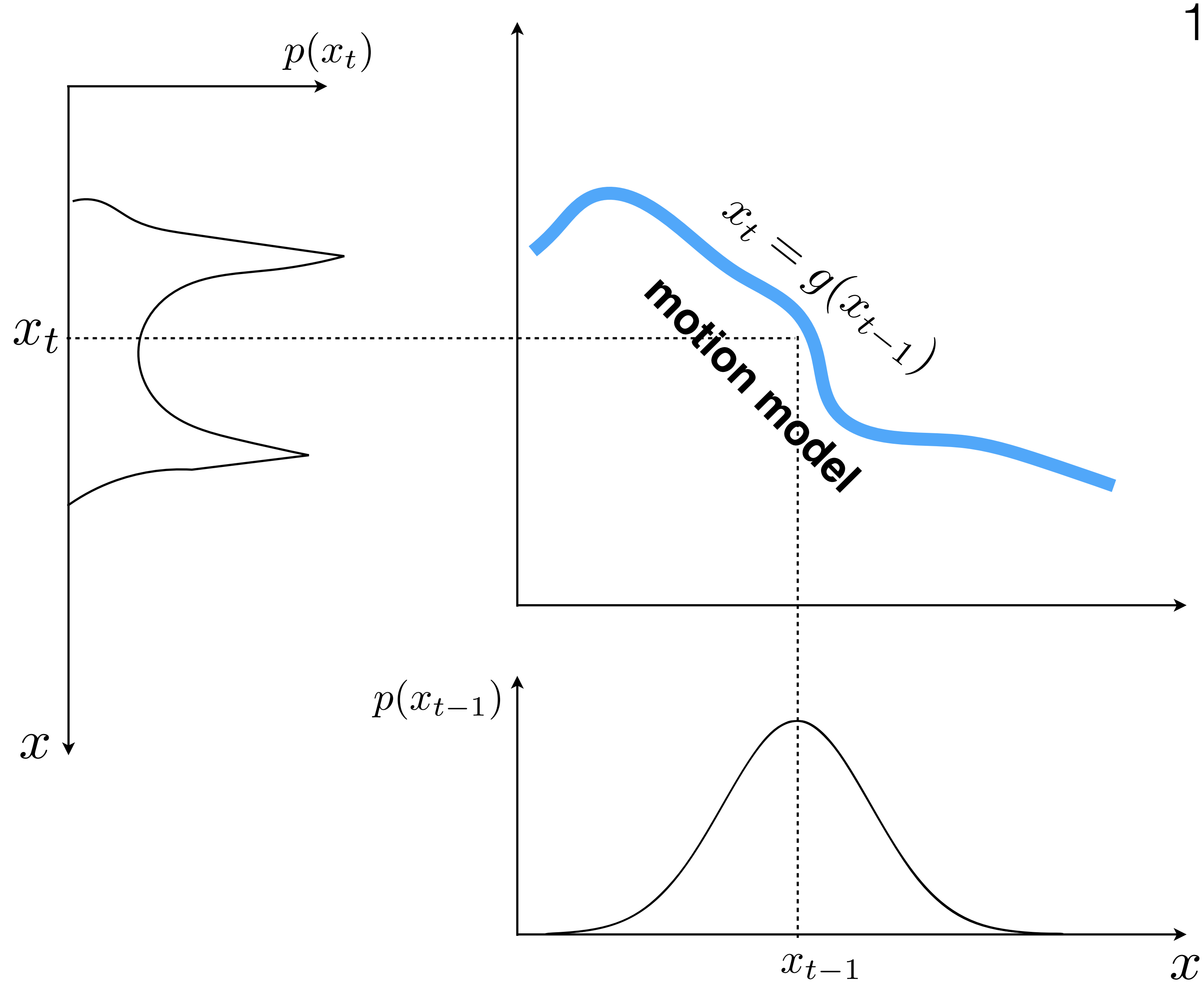
Can we use the Kalman Filter?

(motion model and observation model are linear)

Visualizing **non-linear** models

1D motion model example

Output: **NOT** Gaussian



Input: Gaussian **(Belief)**

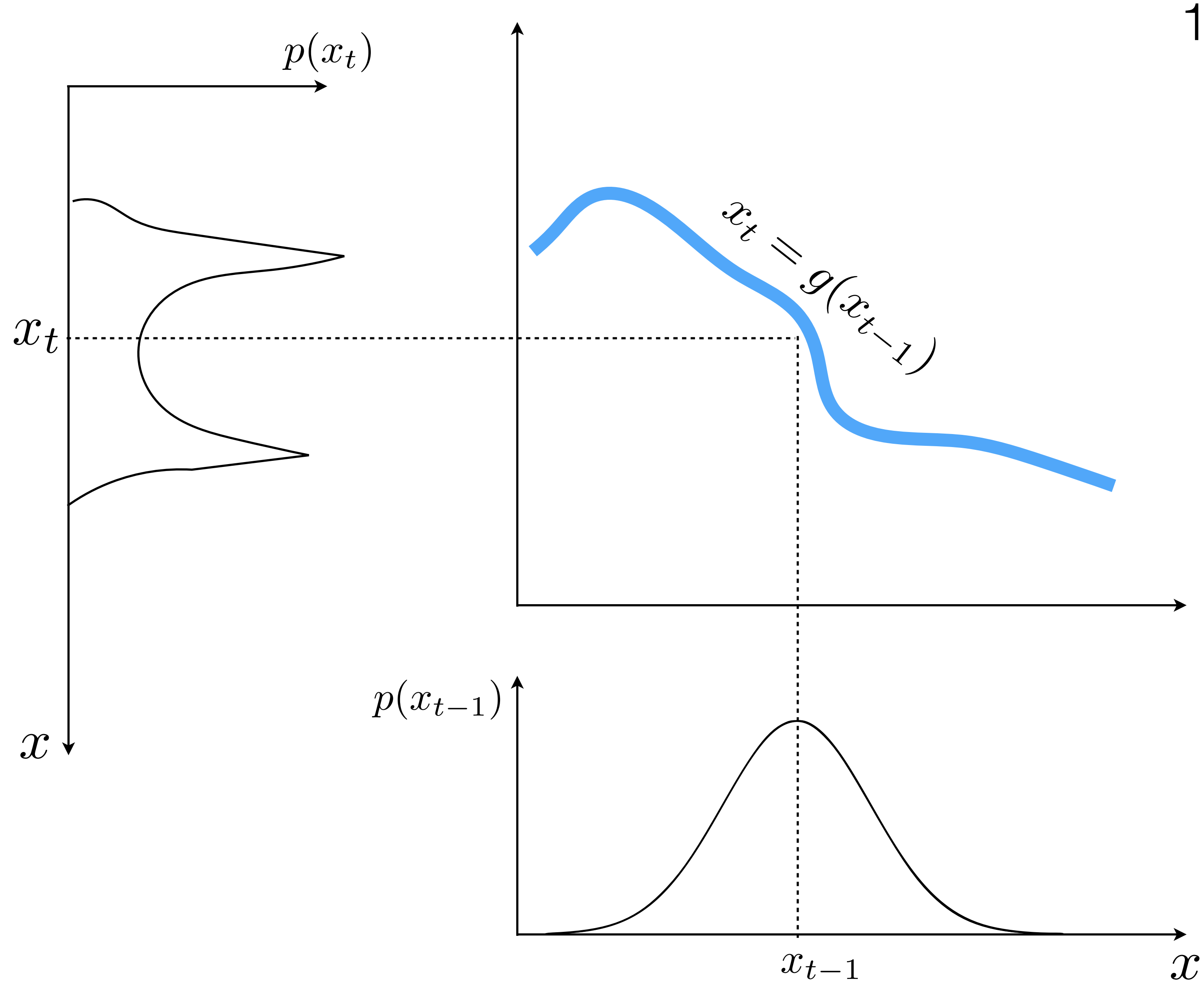
Can we use the Kalman Filter?

(motion model is not linear)

How do you deal with non-linear models?

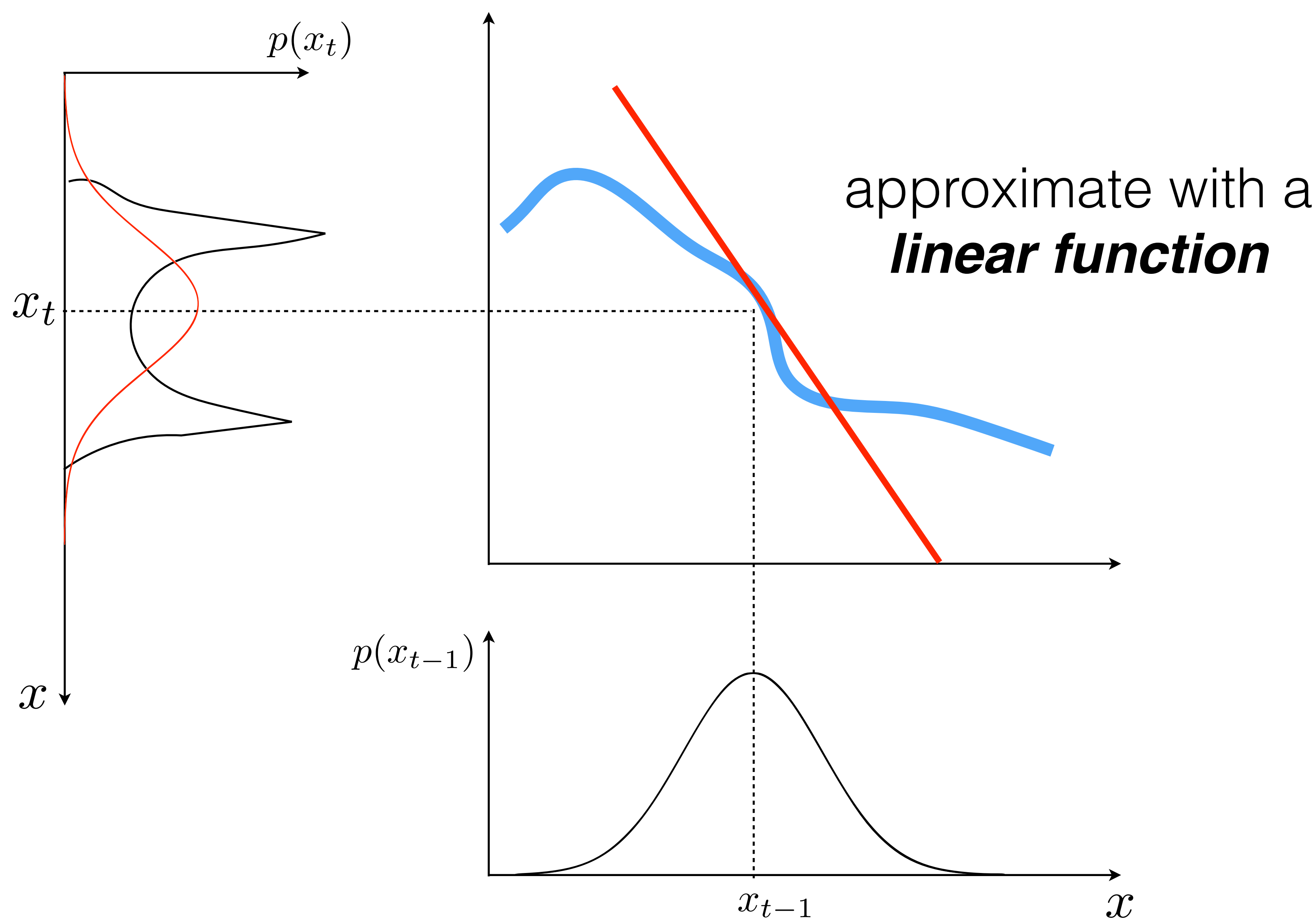
1D motion model example

Output:
NOT
Gaussian



Input:
Gaussian

How do you deal with non-linear models?



Output:
Gaussian

Input:
Gaussian

When does this trick work?

Extended Kalman Filter

- Does not assume linear Gaussian models
- Assumes Gaussian noise
- Uses local linear approximations of model to keep the efficiency of the KF framework

Kalman Filter	Extended Kalman Filter
linear motion model	non-linear motion model
$x_t = Ax_{t-1} + Bu_t + \epsilon_t$	$x_t = g(x_{t-1}, u_t) + \epsilon_t$
linear sensor model	non-linear sensor model
$z_t = C_t x_t + \delta_t$	$z_t = H(x_t) + \delta_t$

Motion model linearization

$$g(x_{t-1}, u_t) \approx g(\mu_{t-1}, u_t) + \frac{\partial g(\mu_{t-1}, u_t)}{\partial x_{t-1}}(x_{t-1} - \mu_{t-1})$$

Taylor series expansion

Motion model linearization

$$
\begin{aligned}
g(x_{t-1}, u_t) &\approx g(\mu_{t-1}, u_t) + \frac{\partial g(\mu_{t-1}, u_t)}{\partial x_{t-1}}(x_{t-1} - \mu_{t-1}) \\
&\approx g(\mu_{t-1}, u_t) + \qquad G_t \qquad (x_{t-1} - \mu_{t-1})
\end{aligned}
$$

What's this called?

Motion model linearization

$$g(x_{t-1}, u_t) \approx g(\mu_{t-1}, u_t) + \frac{\partial g(\mu_{t-1}, u_t)}{\partial x_{t-1}} (x_{t-1} - \mu_{t-1})$$

$$\approx g(\mu_{t-1}, u_t) + G_t (x_{t-1} - \mu_{t-1})$$

What's this called?

Jacobian Matrix

'the rate of change in x'
'slope of the function'

Motion model linearization

$$g(x_{t-1}, u_t) \approx g(\mu_{t-1}, u_t) + \frac{\partial g(\mu_{t-1}, u_t)}{\partial x_{t-1}}(x_{t-1} - \mu_{t-1})$$

$$\approx g(\mu_{t-1}, u_t) + \qquad G_t \qquad (x_{t-1} - \mu_{t-1})$$

Jacobian Matrix

'the rate of change in x'
'slope of the function'

Sensor model linearization

$$h(x_t) \approx h(\bar{\mu}_t) + \frac{\partial h(\bar{\mu}_t)}{\partial x_t}(x_{t-1} - \bar{\mu}_t)$$

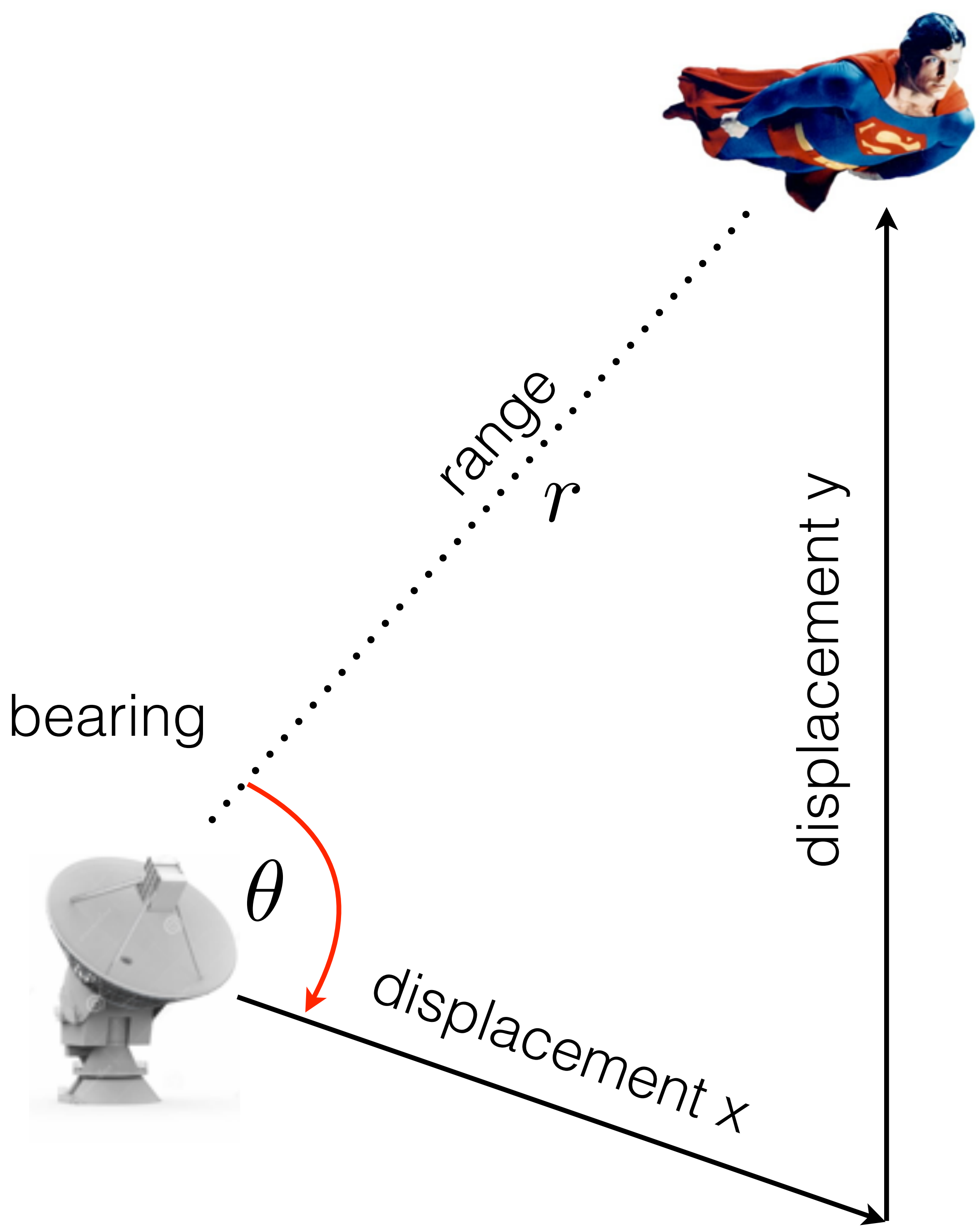
$$\approx h(\bar{\mu}_t) + \quad H_t \quad (x_t - \bar{\mu}_t)$$

New EKF Algorithm

(pretty much the same)

Kalman Filter	Extended KF
$\bar{\mu}_t = A_t \mu_{t-1} + B u_t$	$\bar{\mu}_t = g(\mu_{t-1}, u_t)$
$\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^\top + R$	$\bar{\Sigma}_t = G_t \bar{\Sigma}_{t-1} G_t^\top + R$
$K_t = \bar{\Sigma}_t C_t^\top (C_t \bar{\Sigma}_t C_t^\top + Q_t)^{-1}$	$K_t = \bar{\Sigma}_t H_t^\top \left(H_t \bar{\Sigma}_t H^\top + Q \right)^{-1}$
$\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$	$\mu_t = \bar{\mu}_t + K_t (z_t - h(\bar{\mu}_t))$
$\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$	$\Sigma_t = (I - K_t H_t) \bar{\Sigma}_t$

2D example



state: position-velocity

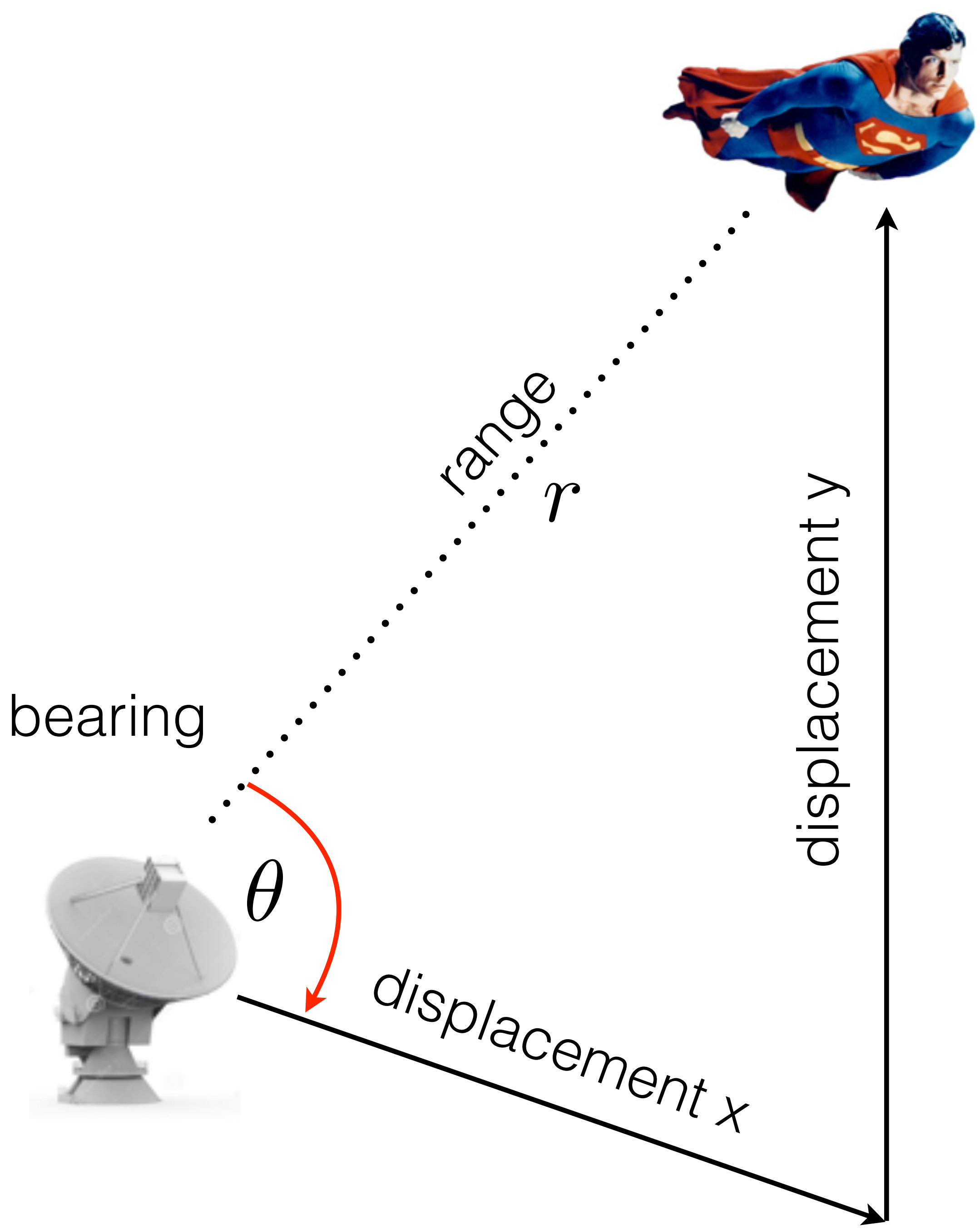
$$\boldsymbol{x} = \begin{array}{c} \left[\begin{array}{c} x \\ \dot{x} \\ y \\ \dot{y} \end{array} \right] \end{array} \begin{array}{l} \text{position} \\ \text{velocity} \\ \text{position} \\ \text{velocity} \end{array}$$

constant velocity motion model

$$A = \left[\begin{array}{cccc} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{array} \right]$$

with additive Gaussian noise

Motion model is linear but …



measurement: range-bearing

$$\boldsymbol{z} = \begin{bmatrix} r \\ \theta \end{bmatrix}$$

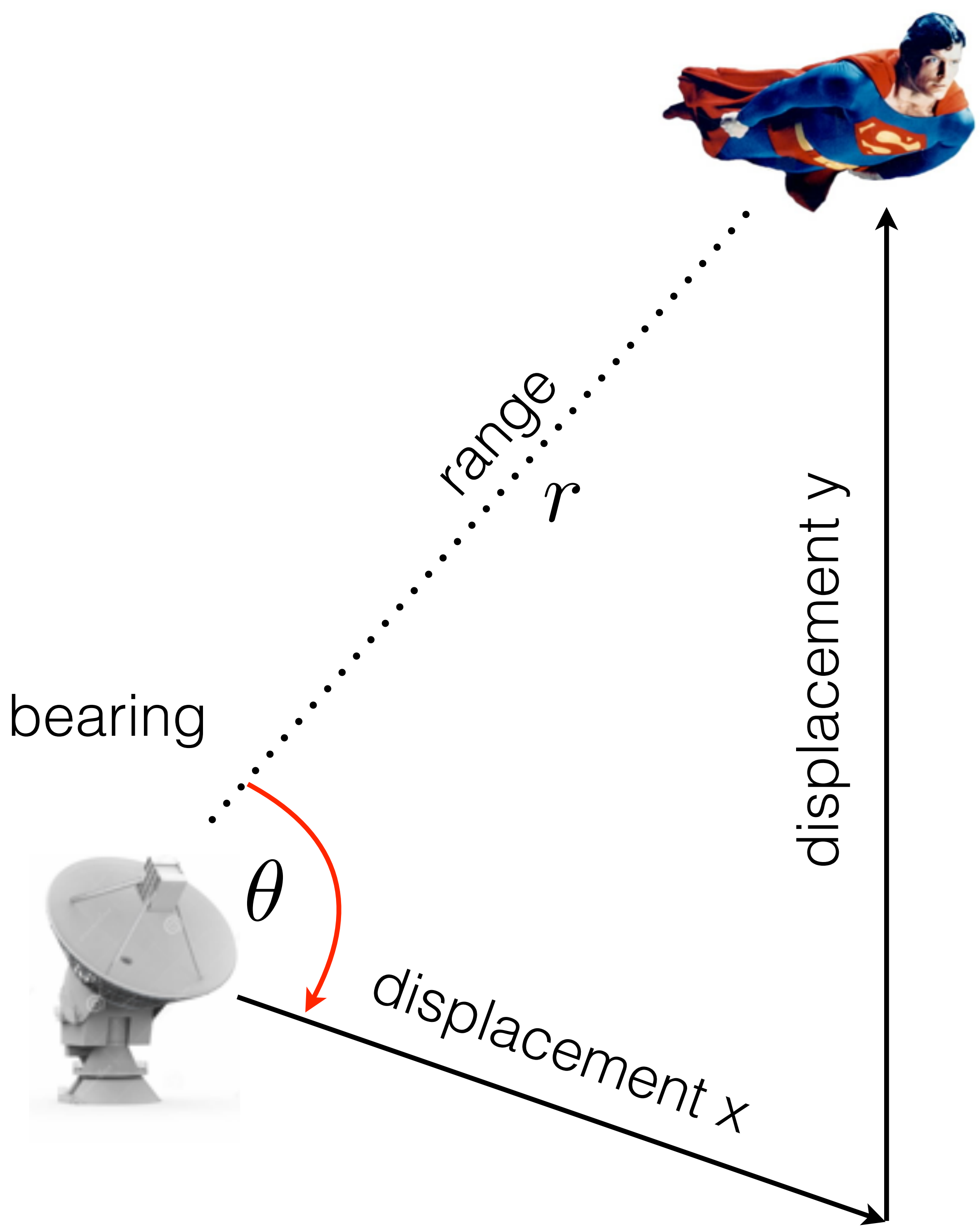
$$= \begin{bmatrix} \sqrt{x^2 + y^2} \\ \tan^{-1}(y/x) \end{bmatrix}$$

measurement model

Is the measurement model linear?

$$\boldsymbol{z} = h(r, \theta)$$

with additive Gaussian noise



measurement: range-bearing

$$\boldsymbol{z} = \left[\begin{array}{c} r \\ \theta \end{array} \right]$$

$$= \left[\begin{array}{c} \sqrt{x^2 + y^2} \\ \tan^{-1}(y/x) \end{array} \right]$$

measurement model

Is the measurement model linear?

$$\boldsymbol{z} = h(r, \theta)$$

with additive Gaussian noise

non-linear!

What should we do?

linearize the observation/measurement model!

$$\boldsymbol{z} = \begin{bmatrix} r \\ \theta \end{bmatrix}$$

$$= \begin{bmatrix} \sqrt{x^2 + y^2} \\ \tan^{-1}(y/x) \end{bmatrix}$$

$$H = \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}} = ?$$

What is the Jacobian?

$$H = \begin{bmatrix} \frac{\partial r}{\partial x} & \frac{\partial r}{\partial \dot{x}} & \frac{\partial r}{\partial y} & \frac{\partial r}{\partial \dot{y}} \\ \\ \frac{\partial \theta}{\partial x} & \frac{\partial \theta}{\partial \dot{x}} & \frac{\partial \theta}{\partial y} & \frac{\partial \theta}{\partial \dot{y}} \end{bmatrix} =$$

$$\boldsymbol{z} = \begin{bmatrix} r \\ \theta \end{bmatrix} = \begin{bmatrix} \sqrt{x^2 + y^2} \\ \tan^{-1}(y/x) \end{bmatrix}$$

$$H = \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}} = ?$$

What is the Jacobian?

Jacobian used in the Taylor series expansion looks like …

$$H = \begin{bmatrix} \frac{\partial r}{\partial x} & \frac{\partial r}{\partial \dot{x}} & \frac{\partial r}{\partial y} & \frac{\partial r}{\partial \dot{y}} \\ \frac{\partial \theta}{\partial x} & \frac{\partial \theta}{\partial \dot{x}} & \frac{\partial \theta}{\partial y} & \frac{\partial \theta}{\partial \dot{y}} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ -sin(\theta)/r & 0 & \cos(\theta)/r & 0 \end{bmatrix}$$

Parameters:

```
Q = diag([0 .1 0 .1]);
R = diag([50^2 0.005^2]);
F = [ 1 dt 0 0;
      0 1 0 0;
      0 0 1 dt;
      0 0 0 1];
```

```
[x P] = EKF(x,P,z,dt)

  r   = sqrt (x(1)^2+x(3)^2);
  b   = atan2(x(3),x(1));
  y   = [r; b];

  H = [ cos(b)    0  sin(b)    0;
       -sin(b)/r 0  cos(b)/r  0];

  x = F*x;
  P = F*P*F' + Q;

  K = P*H'/(H*P*H' + R);

  x   = x + K*(z - y);
  P   = (eye(size(K,1))-K*H)*P;
```

extra computation for the EKF measurement model Jacobian

Problems with EKFs

Taylor series expansion = poor approximation of non-linear functions
success of linearization depends on limited uncertainty and amount of local non-linearity

Computing partial derivatives is a pain

Drifts when linearization is a bad approximation

Cannot handle multi-modal (multi-hypothesis) distributions