MODULE 2 LESSON 3

GOING NONLINEAR: THE EXTENDED KALMAN FILTER

The Extended Kalman Filter (EKF)

By the end of this video, you will be able to

- Describe how the EKF uses first-order linearization to turn a nonlinear problem into a linear one
- Understand the role of Jacobian matrices in the EKF and how to compute them
- Apply the EKF to a simple nonlinear tracking problem

Recap | The Linear Kalman Filter

Linear motion / process model

$$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1}$$

current state — previous state — inputs — process noise

$$\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$$

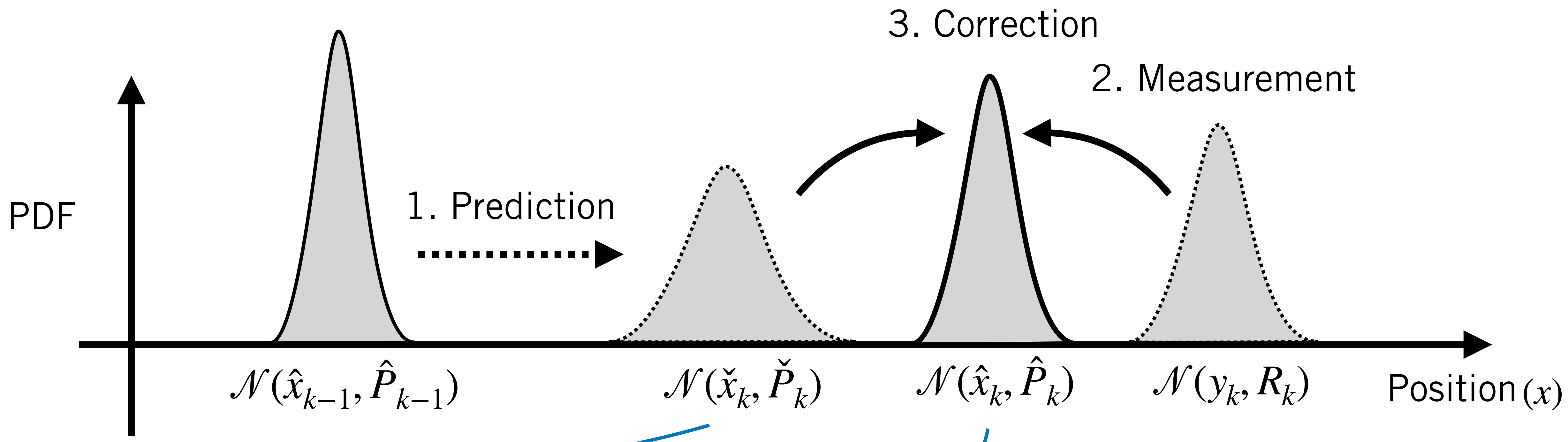
Linear measurement model

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k$$

measurement — state — measurement noise

$$\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$$

Linear discrete-time Kalman Filter:



1. Prediction

$$\begin{cases} \check{\mathbf{x}}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} \\ \check{\mathbf{P}}_k = \mathbf{F}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \end{cases}$$

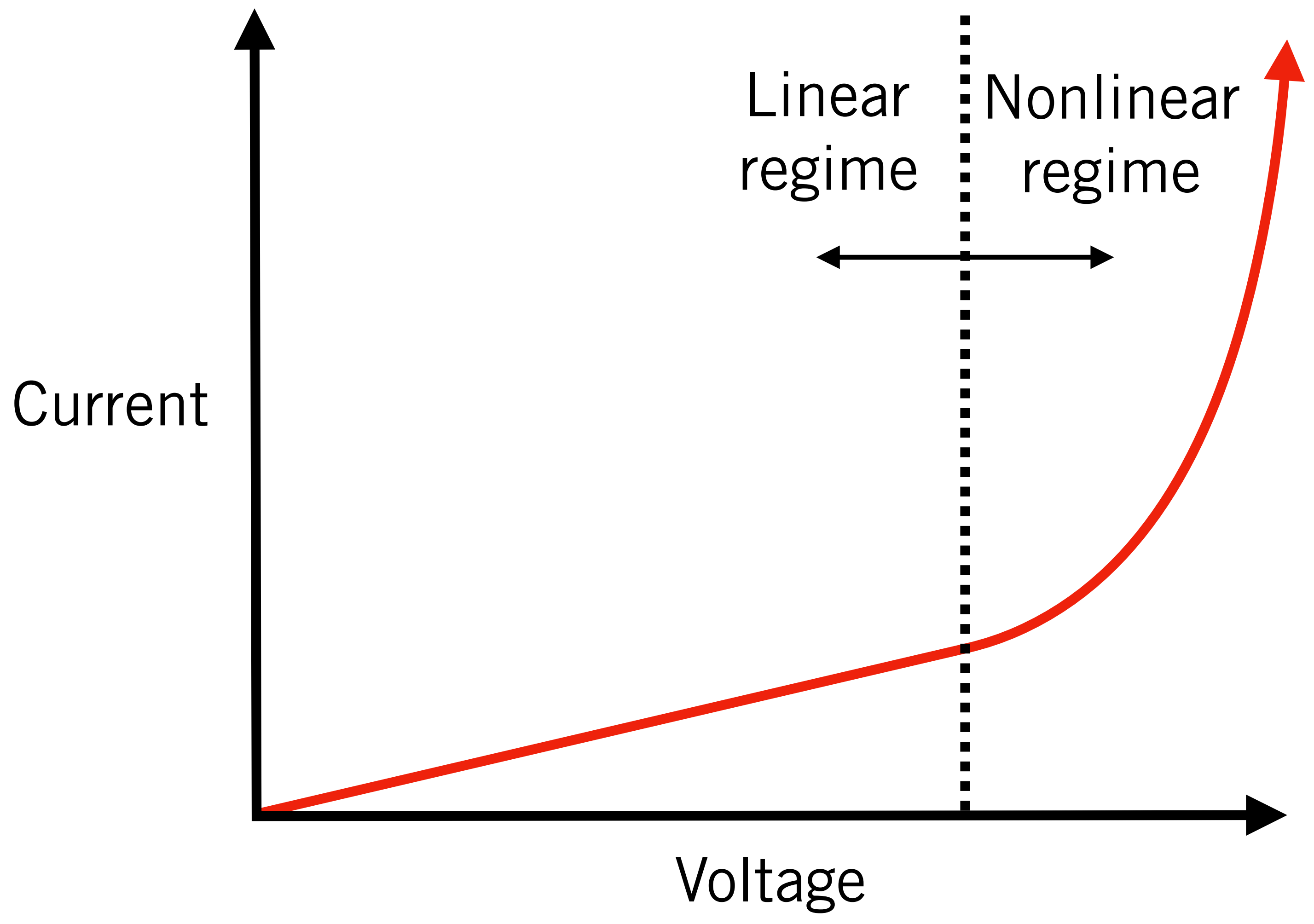
2/3. Measurement & Correction

$$\begin{cases} \mathbf{K}_k = \check{\mathbf{P}}_k\mathbf{H}_k^T(\mathbf{H}_k\check{\mathbf{P}}_k\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \\ \hat{\mathbf{x}}_k = \check{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}_k\check{\mathbf{x}}_k) \\ \hat{\mathbf{P}}_k = (\mathbf{1} - \mathbf{K}_k\mathbf{H}_k)\check{\mathbf{P}}_k \end{cases}$$

Nonlinear Kalman Filtering

Linear systems do not exist in reality!

Ohm's Law: $I = V/R$



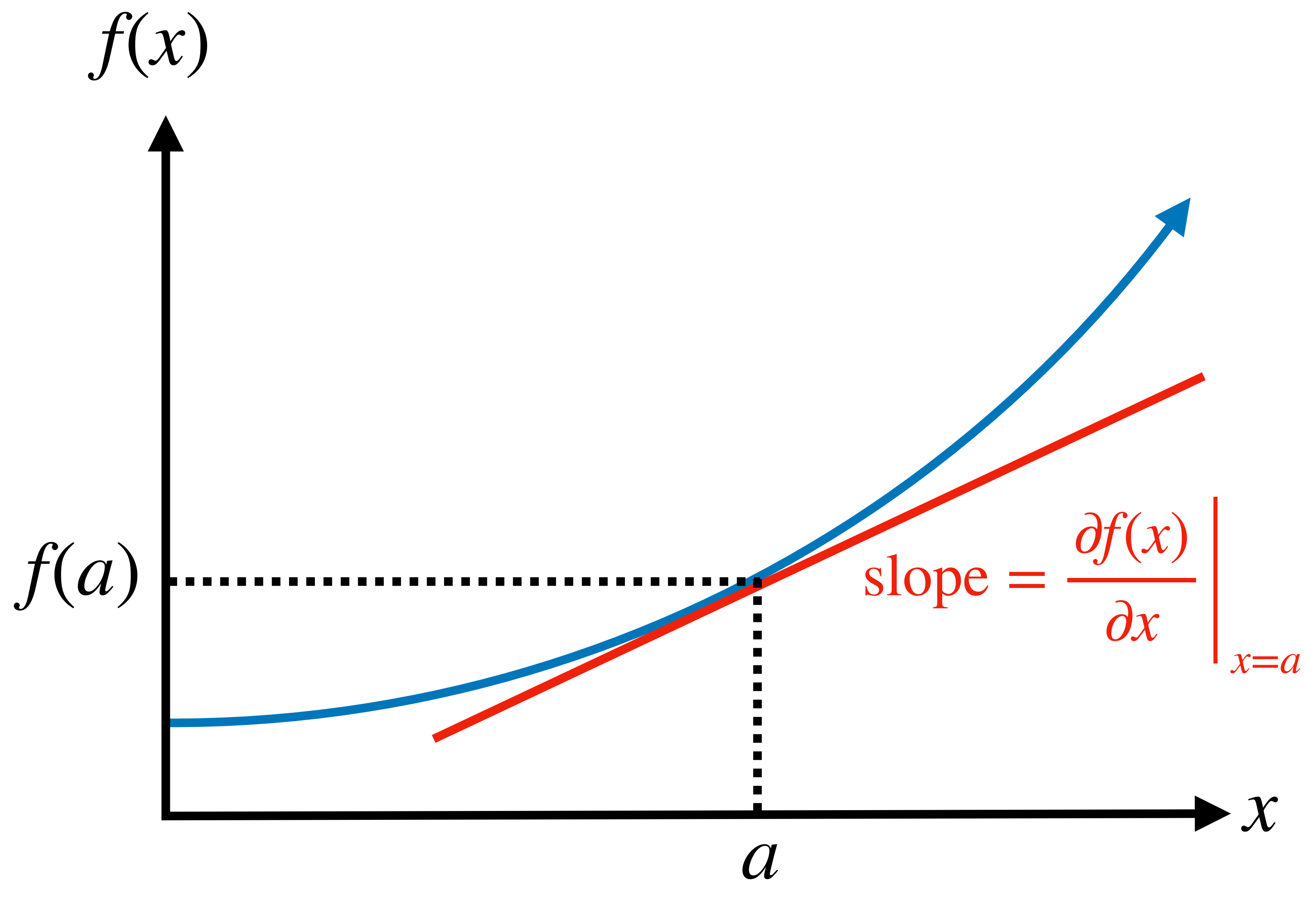
How can we adapt the Kalman Filter to *nonlinear* discrete-time systems?

$$\underset{\substack{\uparrow \\ \text{current} \\ \text{state}}}{\mathbf{x}_k} = \mathbf{f}_{k-1}(\underset{\substack{\uparrow \\ \text{previous} \\ \text{state}}}{\mathbf{x}_{k-1}}, \underset{\substack{\uparrow \\ \text{inputs}}}{\mathbf{u}_{k-1}}, \underset{\substack{\uparrow \\ \text{process} \\ \text{noise}}}{\mathbf{w}_{k-1}})$$

$$\underset{\substack{\uparrow \\ \text{measurement}}}{\mathbf{y}_k} = \mathbf{h}_k(\underset{\substack{\uparrow \\ \text{state}}}{\mathbf{x}_k}, \underset{\substack{\uparrow \\ \text{measurement} \\ \text{noise}}}{\mathbf{v}_k})$$

EKF | Linearizing a Nonlinear System

Choose an operating point *a* and approximate the nonlinear function by a tangent line at that point



Mathematically, we compute this linear approximation using a first-order Taylor expansion:

$$\boxed{f(x) \approx f(a) + \left.\frac{\partial f(x)}{\partial x}\right|_{x=a}(x-a)} + \frac{1}{2!}\left.\frac{\partial^2 f(x)}{\partial x^2}\right|_{x=a}(x-a)^2 + \frac{1}{3!}\left.\frac{\partial^2 f(x)}{\partial x^2}\right|_{x=a}(x-a)^3 + \dots$$

First-order terms

Higher-order terms

EKF | Linearizing a Nonlinear System

For the EKF, we choose the operating point to be our most recent state estimate, our known input, and zero noise:

<u>Linearized motion model</u>

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}) \approx \mathbf{f}_{k-1}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0}) + \underbrace{\left.\frac{\partial \mathbf{f}_{k-1}}{\partial \mathbf{x}_{k-1}}\right|_{\hat{\mathbf{x}}_{k-1},\mathbf{u}_{k-1},\mathbf{0}}}_{\mathbf{F}_{k-1}} (\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}) + \underbrace{\left.\frac{\partial \mathbf{f}_{k-1}}{\partial \mathbf{w}_{k-1}}\right|_{\hat{\mathbf{x}}_{k-1},\mathbf{u}_{k-1},\mathbf{0}}}_{\mathbf{L}_{k-1}} \mathbf{w}_{k-1}$$

<u>Linearized measurement model</u>

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{v}_k) \approx \mathbf{h}_k(\check{\mathbf{x}}_k, \mathbf{0}) + \underbrace{\left.\frac{\partial \mathbf{h}_k}{\partial \mathbf{x}_k}\right|_{\check{\mathbf{x}}_k,\mathbf{0}}}_{\mathbf{H}_k} (\mathbf{x}_k - \check{\mathbf{x}}_k) + \underbrace{\left.\frac{\partial \mathbf{h}_k}{\partial \mathbf{v}_k}\right|_{\check{\mathbf{x}}_k,\mathbf{0}}}_{\mathbf{M}_k} \mathbf{v}_k$$

We now have a linear system in state-space! The matrices $\mathbf{F}_{k-1}$, $\mathbf{L}_{k-1}$, $\mathbf{H}_k$, and $\mathbf{M}_k$ are called the *Jacobian matrices* of the system

EKF | Computing Jacobian Matrices

In vector calculus, a *Jacobian matrix* is the matrix of all first-order partial derivatives of a vector-valued function

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial x_1} & \cdots & \frac{\partial \mathbf{f}}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Intuitively, the Jacobian matrix tells you how fast each output of the function is changing along each input dimension

For example:

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} x_1 + x_2 \\ x_1^2 \end{bmatrix} \quad \Rightarrow \quad \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2x_1 & 0 \end{bmatrix}$$

EKF | Putting It All Together

With our linearized models and Jacobians, we can now use the Kalman Filter equations!

Linearized motion model

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0}) + \mathbf{F}_{k-1}\left(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}\right) + \mathbf{L}_{k-1}\mathbf{w}_{k-1}$$

Linearized measurement model

$$\mathbf{y}_k = \mathbf{h}_k(\check{\mathbf{x}}_k, \mathbf{0}) + \mathbf{H}_k\left(\mathbf{x}_k - \check{\mathbf{x}}_k\right) + \mathbf{M}_k\mathbf{v}_k$$

Prediction

$$\check{\mathbf{x}}_k = \mathbf{f}_{k-1}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0})$$

$$\check{\mathbf{P}}_k = \mathbf{F}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{F}_{k-1}^T + \mathbf{L}_{k-1}\mathbf{Q}_{k-1}\mathbf{L}_{k-1}^T$$

Optimal gain

$$\mathbf{K}_k = \check{\mathbf{P}}_k\mathbf{H}_k^T(\mathbf{H}_k\check{\mathbf{P}}_k\mathbf{H}_k^T + \mathbf{M}_k\mathbf{R}_k\mathbf{M}_k^T)^{-1}$$

Correction

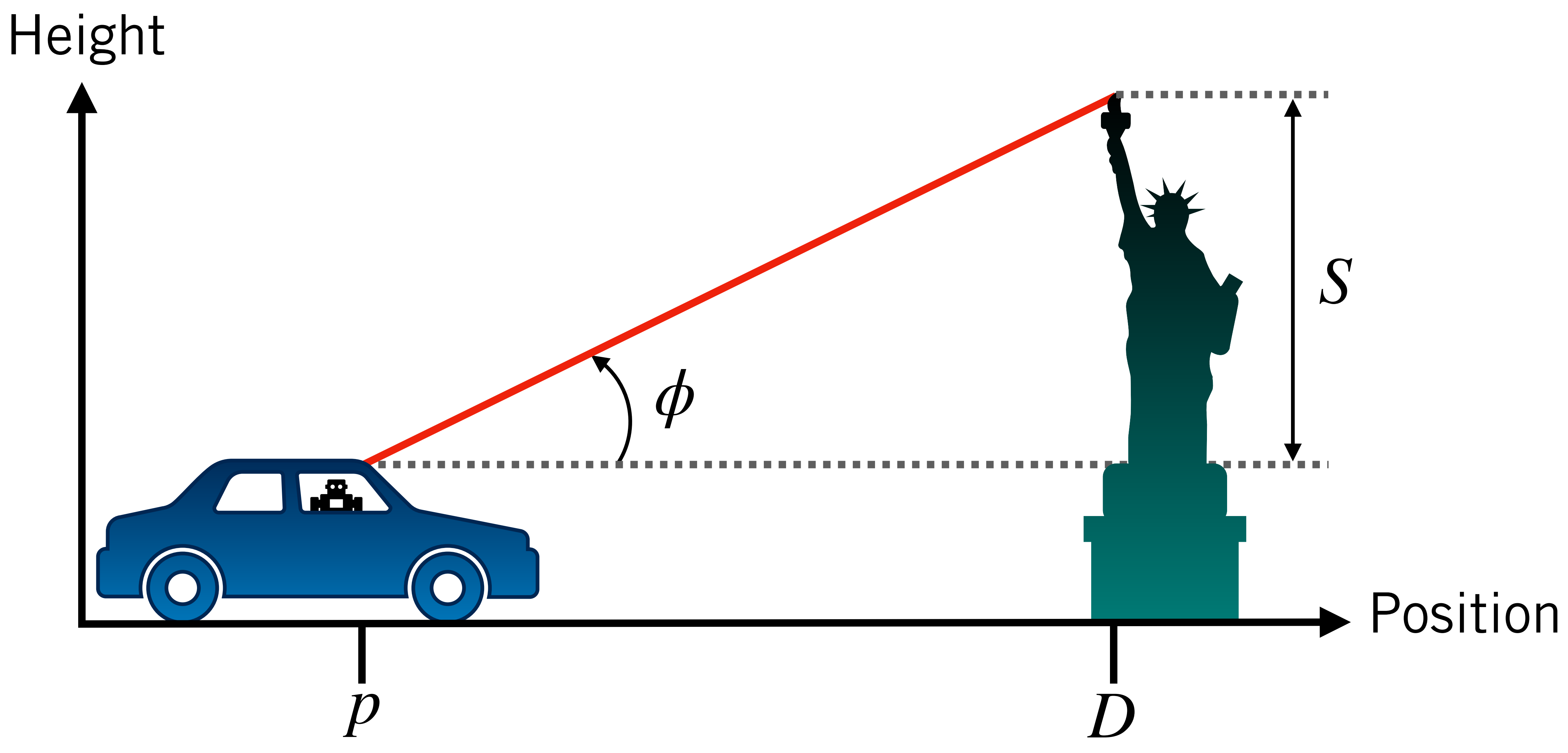
$$\hat{\mathbf{x}}_k = \check{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{y}_k - \mathbf{h}_k(\check{\mathbf{x}}_k, \mathbf{0}))$$

$$\hat{\mathbf{P}}_k = (\mathbf{1} - \mathbf{K}_k\mathbf{H}_k)\check{\mathbf{P}}_k$$

$\check{\mathbf{x}}_k$ Prediction (given motion model) at time *k*

$\hat{\mathbf{x}}_k$ Corrected prediction (given measurement) at time *k*

EKF | Short Example



$$\mathbf{x} = \begin{bmatrix} p \\ \dot{p} \end{bmatrix} \qquad \mathbf{u} = \ddot{p}$$

S and *D* are known in advance

Motion/Process model

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1})$$

$$= \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \begin{bmatrix} 0 \\ \Delta t \end{bmatrix} \mathbf{u}_{k-1} + \mathbf{w}_{k-1}$$

Landmark measurement model

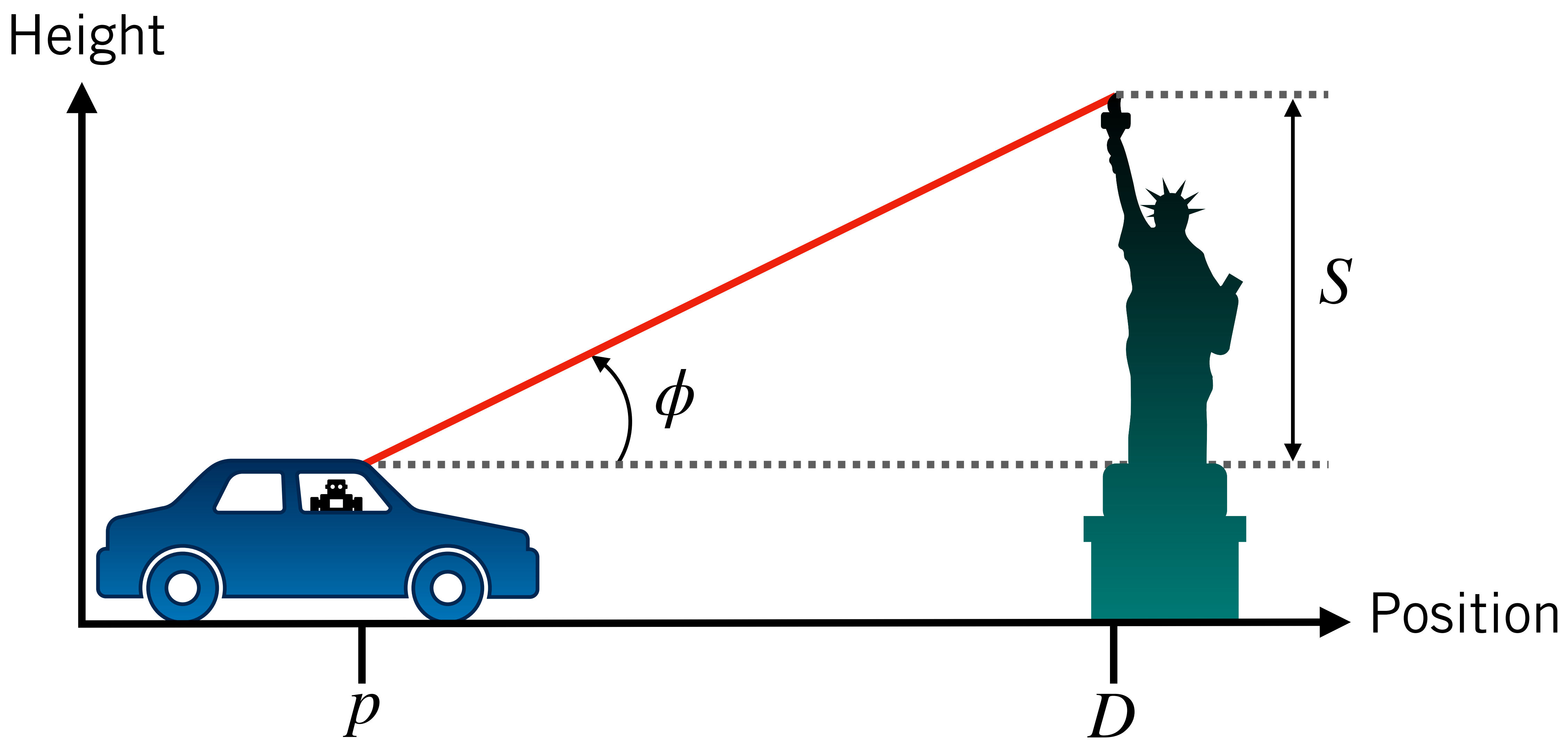
$$y_k = \phi_k = h(p_k, v_k)$$

$$= \tan^{-1}\left(\frac{S}{D - p_k}\right) + v_k$$

Noise densities

$$v_k \sim \mathcal{N}(0,\ 0.01) \qquad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, (0.1)\mathbf{1}_{2\times 2})$$

EKF | Short Example



Motion model Jacobians

$$\mathbf{F}_{k-1} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}_{k-1}}\right|_{\hat{\mathbf{x}}_{k-1},\mathbf{u}_{k-1},\mathbf{0}} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$

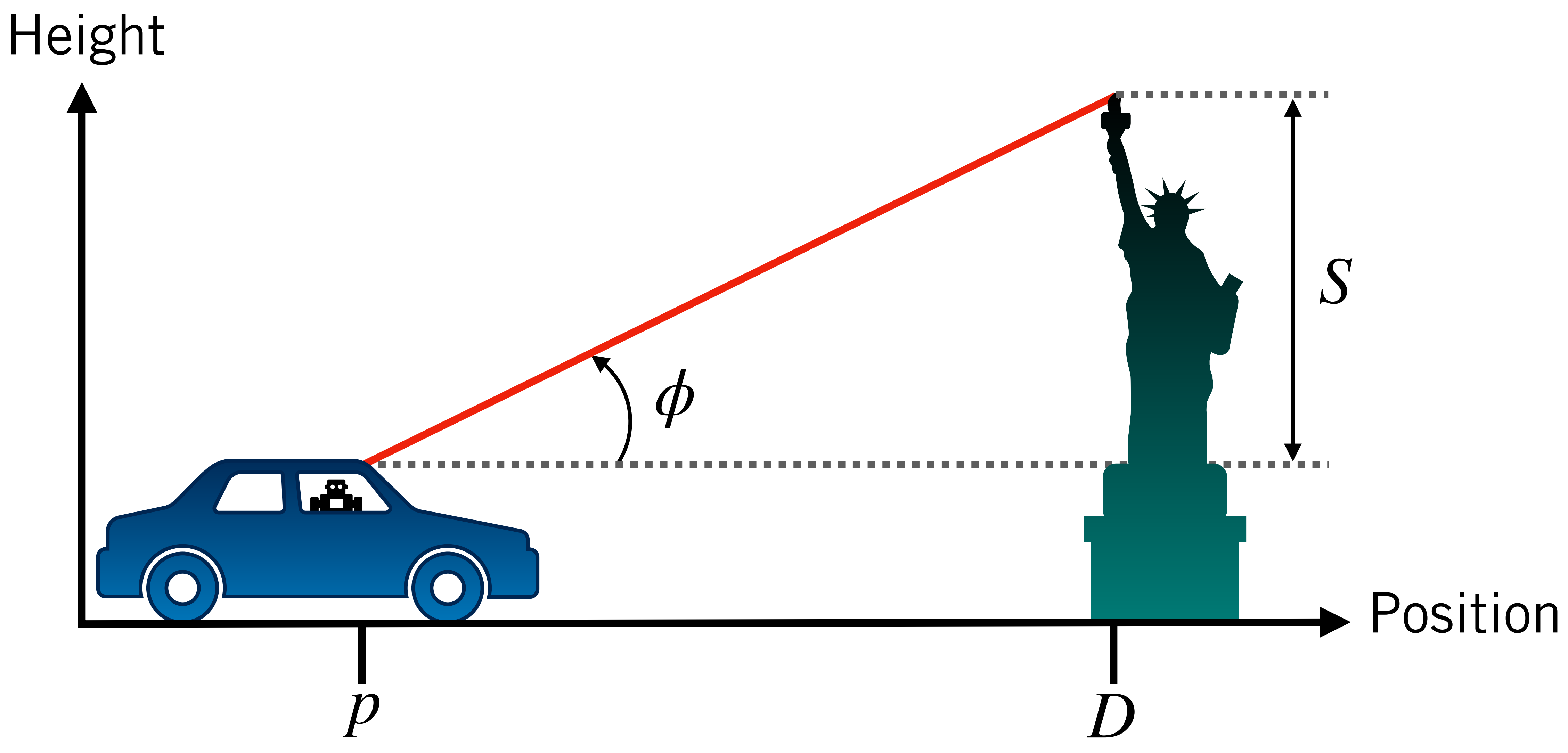
$$\mathbf{L}_{k-1} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{w}_{k-1}}\right|_{\hat{\mathbf{x}}_{k-1},\mathbf{u}_{k-1},\mathbf{0}} = \mathbf{1}_{2\times 2}$$

Measurement model Jacobians

$$\mathbf{H}_{k} = \left.\frac{\partial h}{\partial \mathbf{x}_{k}}\right|_{\check{\mathbf{x}}_{k},\mathbf{0}} = \begin{bmatrix} \frac{S}{(D-\check{p}_k)^2 + S^2} & 0 \end{bmatrix}$$

$$M_{k} = \left.\frac{\partial h}{\partial v_{k}}\right|_{\check{\mathbf{x}}_{k},\mathbf{0}} = 1$$

EKF | Short Example



<u>Data</u>

$$\hat{\mathbf{x}}_0 \sim \mathcal{N}\left(\begin{bmatrix}0\\5\end{bmatrix}, \begin{bmatrix}0.01 & 0\\0 & 1\end{bmatrix}\right)$$

$$\Delta t = 0.5 \text{ s}$$

$u_0 = -2\ [m/s^2]$ $\qquad$ $y_1 = \pi/6$ [rad]

$S = 20\ [m]$ $\qquad$ $D = 40\ [m]$

Using the Extended Kalman Filter equations, what is our updated position?

$$\hat{p}_1$$

EKF | Short Example Solution

Prediction

$$\check{\mathbf{x}}_1 = \mathbf{f}_0(\hat{\mathbf{x}}_0, \mathbf{u}_0, \mathbf{0})$$

$$\begin{bmatrix} \check{p}_1 \\ \check{\dot{p}}_1 \end{bmatrix} = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 5 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} (-2) = \begin{bmatrix} 2.5 \\ 4 \end{bmatrix}$$

This is the same result as in the linear Kalman Filter example because the motion model is already linear!

$$\check{\mathbf{P}}_1 = \mathbf{F}_0 \hat{\mathbf{P}}_0 \mathbf{F}_0^T + \mathbf{L}_0 \mathbf{Q}_0 \mathbf{L}_0^T$$

$$\check{\mathbf{P}}_1 = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0.01 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0.5 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.36 & 0.5 \\ 0.5 & 1.1 \end{bmatrix}$$

EKF | Short Example Solution

Correction

$$\mathbf{K}_1 = \check{\mathbf{P}}_1 \mathbf{H}_1^T (\mathbf{H}_1 \check{\mathbf{P}}_1 \mathbf{H}_1^T + \mathbf{M}_1 \mathbf{R}_1 \mathbf{M}_1^T)^{-1}$$

$$= \begin{bmatrix} 0.36 & 0.5 \\ 0.5 & 1.1 \end{bmatrix} \begin{bmatrix} 0.011 \\ 0 \end{bmatrix} \left(\begin{bmatrix} 0.011 & 0 \end{bmatrix} \begin{bmatrix} 0.36 & 0.5 \\ 0.5 & 1.1 \end{bmatrix} \begin{bmatrix} 0.011 \\ 0 \end{bmatrix} + 1(0.01)(1) \right)^{-1}$$

$$= \begin{bmatrix} 0.40 \\ 0.55 \end{bmatrix}$$

$$\hat{\mathbf{x}}_1 = \check{\mathbf{x}}_1 + \mathbf{K}_1(\mathbf{y}_1 - \mathbf{h}_1(\check{\mathbf{x}}_1, \mathbf{0}))$$

$$\begin{bmatrix} \hat{p}_1 \\ \hat{\dot{p}}_1 \end{bmatrix} = \begin{bmatrix} 2.5 \\ 4 \end{bmatrix} + \begin{bmatrix} 0.40 \\ 0.55 \end{bmatrix} (0.52 - 0.49) = \begin{bmatrix} 2.51 \\ 4.02 \end{bmatrix}$$

Bonus!

$$\hat{\mathbf{P}}_1 = (\mathbf{1} - \mathbf{K}_1 \mathbf{H}_1) \check{\mathbf{P}}_1$$

$$= \begin{bmatrix} 0.36 & 0.50 \\ 0.50 & 1.1 \end{bmatrix}$$

Summary | Extended Kalman Filter (EKF)

- The EKF uses *linearization* to adapt the Kalman filter to nonlinear systems
- Linearization works by computing a local linear approximation to a nonlinear function about a chosen operating point
- Linearization relies on computing *Jacobian matrices*, which contain all the first-order partial derivatives of a function