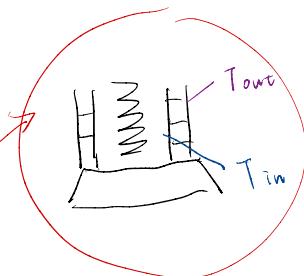
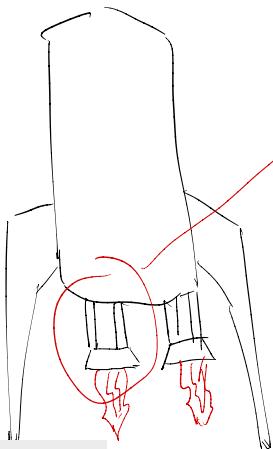


• Why we use Kalman Filters?

1月31:



常用的传感器

T_{out}: Available

T_{in}: Not available for measurement

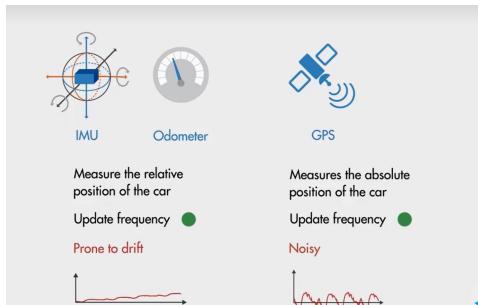
- The variables of interest can only be measured indirectly.

Kalman Filter

利用间接接测量值，
计算内部温度的最优估算值。

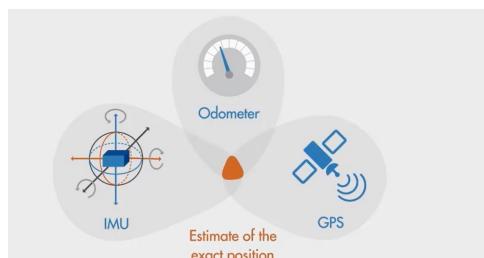
Optimal estimation
algorithm.

1月31:



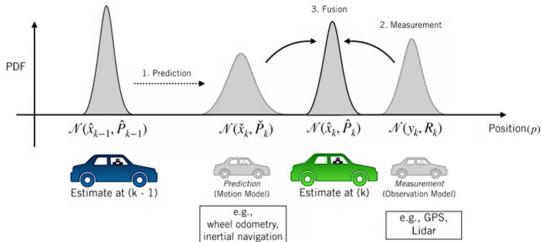
有 Noise

- Measurements are available from various sensors but might be subject to noise.



Linear Kalman Filter.

The Kalman Filter | Prediction and Correction



The Kalman Filter | Linear Dynamical System

- The Kalman Filter requires the following motion and measurement models:

$$\text{Motion model: } \mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \quad \begin{matrix} \text{input} \\ \text{noise} \end{matrix}$$

$$\text{Measurement model: } \mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \quad \begin{matrix} \text{noise} \end{matrix}$$

- With the following noise properties:

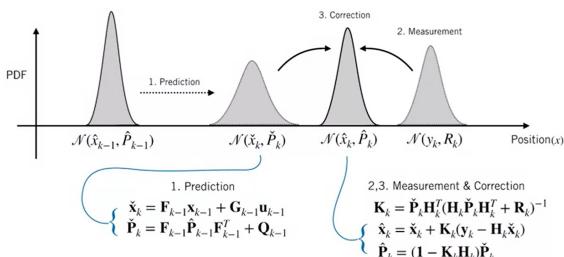
$$\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k) \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$$

Measurement Noise Process or Motion Noise

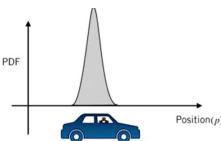
The Kalman Filter | Recursive Least Squares + Process Model

- The Kalman filter is a recursive least squares estimator that also includes a motion model

1 Prediction $\hat{\mathbf{x}}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1}$ $\hat{\mathbf{P}}_k = \mathbf{F}_{k-1}\check{\mathbf{P}}_{k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1}$	2a Optimal Gain $\mathbf{K}_k = \check{\mathbf{P}}_k \mathbf{H}_k^T (\mathbf{H}_k \check{\mathbf{P}}_k \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$
2b Correction $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k)$ $\hat{\mathbf{P}}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \check{\mathbf{P}}_k$ $(\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k)$ is often called "innovation"	Prediction $\hat{\mathbf{x}}_k$ (given motion model) at time k Corrected prediction $\hat{\mathbf{x}}_k$ (given measurement) at time k



Ex:



太抽象了!!!

我们已知 位置 和 其一阶导数速度

$$X = \begin{bmatrix} p \\ \frac{dp}{dt} = \dot{p} \end{bmatrix}$$

我们输入量则为 加速度

$$u = a = \frac{d^2 p}{dt^2}$$

- Motion / Process Model

$$\dot{X}_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} X_{k-1} + \begin{bmatrix} 0 \\ \Delta v \end{bmatrix} u_{k-1} + w_{k-1}$$

- Position Observation

$$y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} X_k + v_k$$

- Noise Densities

$$v_k \sim \mathcal{N}(0, 0.05) \quad w_k \sim \mathcal{N}(\mathbf{0}, (0.1)\mathbf{I}_{2 \times 2})$$

Prediction

$$\begin{aligned} \dot{x}_k &= F_{k-1}x_{k-1} + G_{k-1}u_{k-1} \\ \begin{bmatrix} \dot{p}_1 \\ \dot{p}_1 \end{bmatrix} &= \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 5 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} (-2) = \begin{bmatrix} 2.5 \\ 4 \end{bmatrix} \end{aligned}$$

- Data

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

$$\Delta t = 0.5s$$

$$u_0 = -2 \text{ [m/s}^2\text{]} \quad y_1 = 2.2 \text{ [m]}$$

$$\dot{P}_k = F_{k-1} \dot{P}_{k-1} F_{k-1}^T + Q_{k-1}$$

$$\dot{P}_1 = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0.01 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix}^T + \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} = \begin{bmatrix} 0.36 & 0.5 \\ 0.5 & 1.1 \end{bmatrix}$$

Correction

$$\begin{aligned} K_1 &= \dot{P}_1 H_1^T (H_1 \dot{P}_1 H_1^T + R_1)^{-1} \\ &= \begin{bmatrix} 0.36 & 0.5 \\ 0.5 & 1.1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \left(\begin{bmatrix} 1 & 0 \\ 0.5 & 1.1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 0.05 \right)^{-1} \\ &= \begin{bmatrix} 0.88 \\ 1.22 \end{bmatrix} \end{aligned}$$

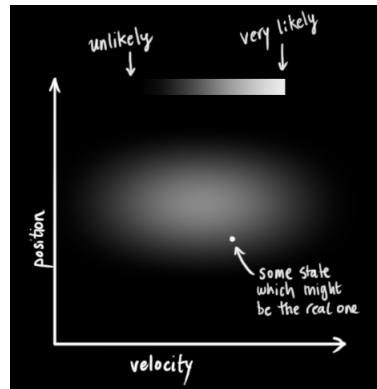
$$\hat{x}_1 = \hat{x}_0 + K_1(y_1 - H_1 \hat{x}_0)$$

$$\begin{bmatrix} \hat{p}_1 \\ \hat{p}_1 \end{bmatrix} = \begin{bmatrix} 2.5 \\ 4 \end{bmatrix} + \begin{bmatrix} 0.88 \\ 1.22 \end{bmatrix} (2.2 - \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 2.5 \\ 4 \end{bmatrix}) = \begin{bmatrix} 2.24 \\ 3.63 \end{bmatrix}$$

$$\begin{aligned} \dot{P}_1 &= (I - K_1 H_1) \dot{P}_1 \\ &= \begin{bmatrix} 0.04 & 0.06 \\ 0.06 & 0.49 \end{bmatrix} \end{aligned}$$

详细例3：

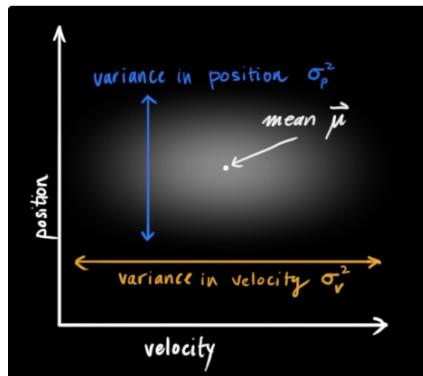
$$\vec{x} = \begin{bmatrix} p \\ v \end{bmatrix}$$



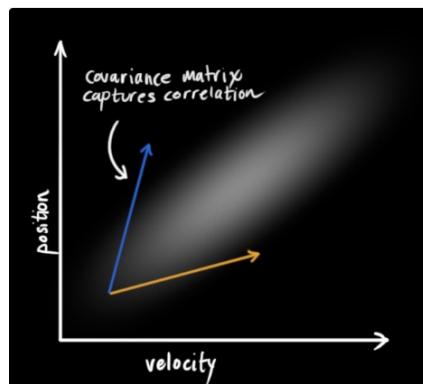
我们不知道
相前实际 P 和
 V 是多少

Kalman filter 假设两个变量 (P 和 V)

都是随机且高斯分布 (μ , σ^2)



从图中 P 和 V
不相关，这意味着
 V 的状态
无法告诉 P 的状态。



从图中 P 和 V
correlate (covariance)
这种关系用协方差
矩阵来捕获。

$\sum v_j$
(ith state variable and
jth ~ correlate.)

▷ Describing problem with matrices.

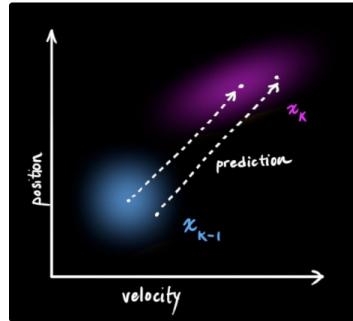
We see two pieces of information at k time

We call best estimate \hat{x}_k (the mean, elsewhere named μ)

and its covariance matrix P_k

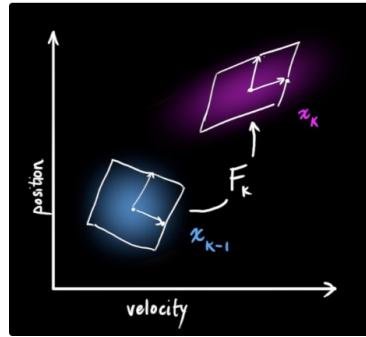
$$\hat{x}_k = \begin{bmatrix} \text{Position} \\ \text{velocity} \end{bmatrix} \quad P_k = \begin{bmatrix} \Sigma_{pp} & \Sigma_{pv} \\ \Sigma_{vp} & \Sigma_{vv} \end{bmatrix}$$

当前状态 ($k-1$)
预测状态 (k)



→ 高斯行驶点

用矩阵 F_k
表示预测运动



现在我们有一个预测
矩阵，但我们还不知道
如何更新 covariance
↑

$$P_k = P_{k-1} + \Delta t \cdot V_{k-1}$$

$$V_k = V_{k-1}$$

\Rightarrow

$$= F_k \cdot \hat{x}_{k-1}$$

$$\hat{x}_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \hat{x}_{k-1}$$

$$\text{Cov}(x) = \Sigma$$

$$\text{Cov}(Ax) = A \Sigma A^T$$

$$\Rightarrow \hat{x}_k = F_k \hat{x}_{k-1}$$

$$P_k = F_k P_{k-1} F_k^T$$

△ External influence.
外部世界会影响系统
at time: 加速度 a.

$$\left\{ \begin{array}{l} P_k = P_{k-1} + \Delta t V_{k-1} + \frac{1}{2} a \Delta t^2 \\ V_k = V_{k-1} + a \Delta t \end{array} \right.$$

In matrix form:

$$\hat{x}_k = F_k \hat{x}_{k-1} + \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} a.$$

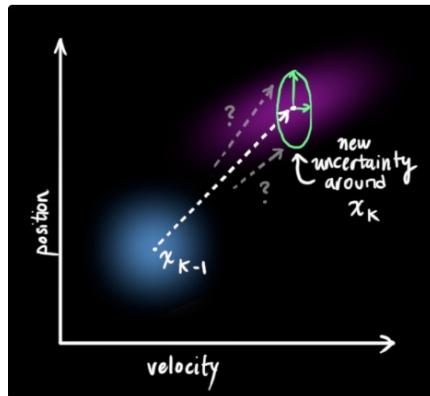
$$= F_k \hat{x}_{k-1} + B_k \vec{u}_k.$$

B_k : control matrix

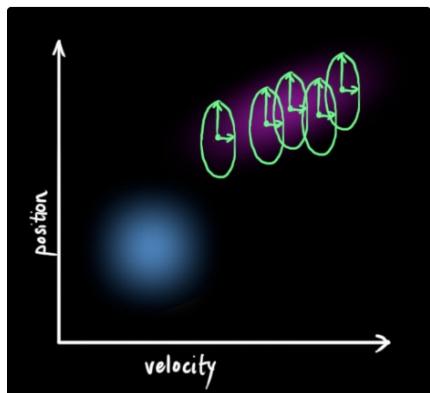
\vec{u}_k : control vector.

▷ External uncertainty.

从图中：无人机会受到风的影响。
对于机器人，轮子会打滑。

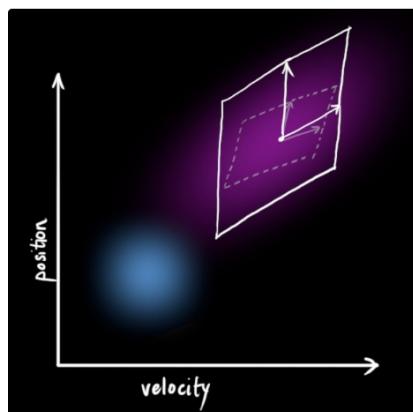


each point in \hat{x}_{k-1} move to somewhere inside 高斯点 with covariance \mathcal{Q}_k
另一种说法是我们将未跟踪的 influence 视为 noise with covariance \mathcal{Q}_k



different covariance
(but the same mean)

We get the expanded covariance by simple adding \mathcal{Q}_k .



$$\hat{x}_k = F_k \hat{x}_{k-1} + B_k \vec{u}_k$$

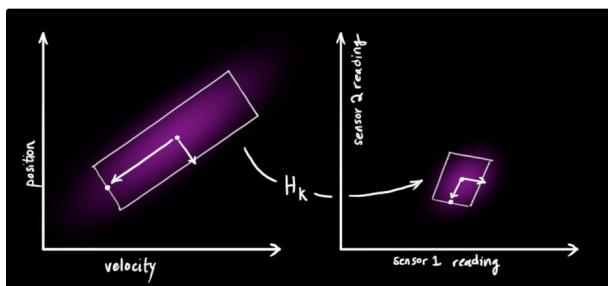
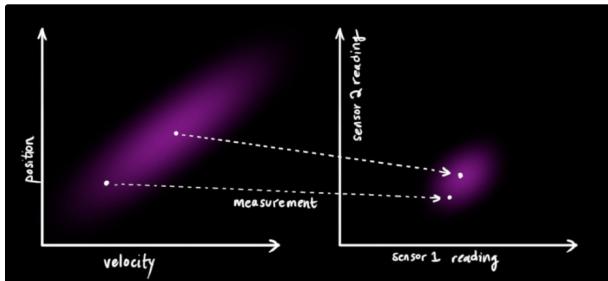
$$P_k = F_k P_{k-1} F_k^T + \mathcal{Q}_k.$$

换句话说，新的最佳估计是根据先前的最佳估计做出的预测，加上对已知外部影响的修正。

新的不确定性是根据旧的不确定性预测的，还有一些来自环境的额外不确定性。

Refining the estimate with measurements.

我们可能有几个传感器可以为我们提供有关系统状态的信息。目前，他们测量什么并不重要；重要的是。也许一个读取位置，另一个读取速度。每个传感器都会间接告诉我们有关状态的信息，换句话说，传感器在一个状态上运行并产生一组读数。



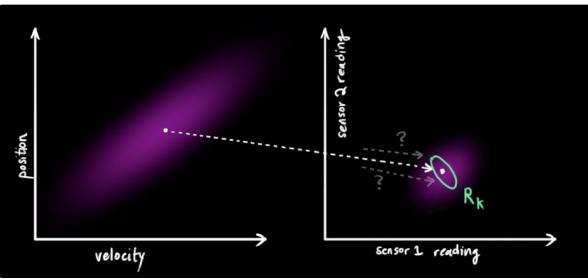
读数的单位和比例
可能与我们跟踪的状态
单元比例不同

∴ 我们使用矩阵对
传感器进行建模

H_k

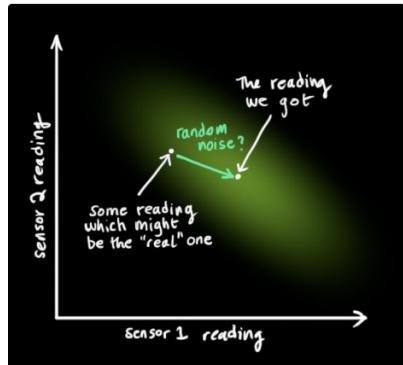
$$\stackrel{\rightarrow}{\mathcal{M}}_{\text{expected}} = H_k \hat{x}_k$$

$$\sum_{\text{expected}} = H_k P_k H_k^T$$



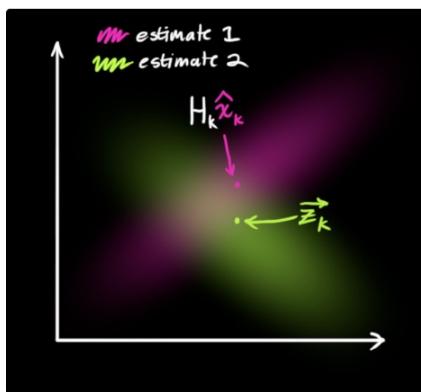
由于存在不确定性

- some state 和其它 state 更有可能产生我们所看到的读数



将这种不确定性 (传感器噪声) 的 covariance 称为 R_k .

该平均分布的 mean 等于 the reading we observed, which called \bar{z}_k .



现在我们有西丁高斯点

一个是转换后的预测平均值

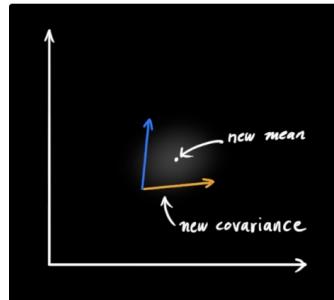
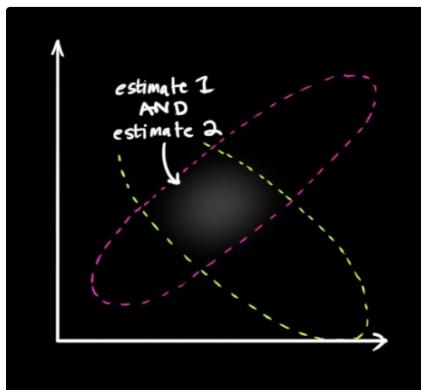
一个是围绕我们得到的实际信号
误差

误差

预测平均值

传感器误差.

两个概率都为真，相乘。



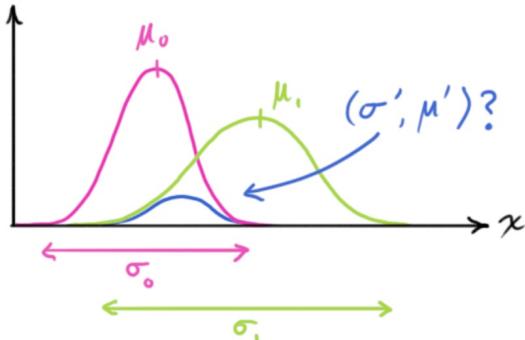
得到重叠

(新的高斯点)

Combining Gaussians

Let's find that formula. It's easiest to look at this first in **one dimension**. A 1D Gaussian bell curve with variance σ^2 and mean μ is defined as:

$$\mathcal{N}(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (9)$$



两个高斯曲线
相乘
得到蓝色。

$$\mathcal{N}(x, \mu_0, \sigma_0) \cdot \mathcal{N}(x, \mu_1, \sigma_1) \stackrel{?}{=} \mathcal{N}(x, \mu', \sigma') \quad (10)$$

$$\mu' = \mu_0 + \frac{\sigma_0^2 (\mu_1 - \mu_0)}{\sigma_0^2 + \sigma_1^2}$$

$$\sigma'^2 = \sigma_0^2 - \frac{\sigma_0^4}{\sigma_0^2 + \sigma_1^2}$$

但矩阵版本：

$$\text{let } K = \frac{\sigma_0^2}{\sigma_0^2 + \sigma_1^2} \quad K = \Sigma_0 (\Sigma_0 + \Sigma_1)^{-1}$$

$$\bar{\mu}' = \mu_0 + K (\mu_1 - \mu_0)$$

$$\sigma'^2 = \sigma_0^2 - K \sigma_0^2$$

$$\bar{\mu}' = \vec{\mu}_0 + K (\vec{\mu}_1 - \vec{\mu}_0)$$

$$\Sigma' = \Sigma_0 - K \Sigma_0$$

(K 为 Kalman gain)

↪ Putting together

two distribution:

- The predicted measurement with $(\mu_0, \Sigma_0) = (H_k \hat{X}_k, H_k P_k H_k^T)$
- observed measurement with $(\mu_1, \Sigma_1) = (\vec{z}_k, R_k)$

$$H_k \hat{X}'_k = H_k \hat{X}_k + K(\vec{z}_k - H_k \hat{X}_k)$$

$$H_k P'_k H_k^T = H_k P_k H_k^T - K H_k P_k H_k^T$$

$$\therefore K = \Sigma_0 (\Sigma_0 + \Sigma_1)^{-1}$$

$$\therefore K' = H_k P_k H_k^T (H_k P_k H_k^T + R_k)^{-1}$$

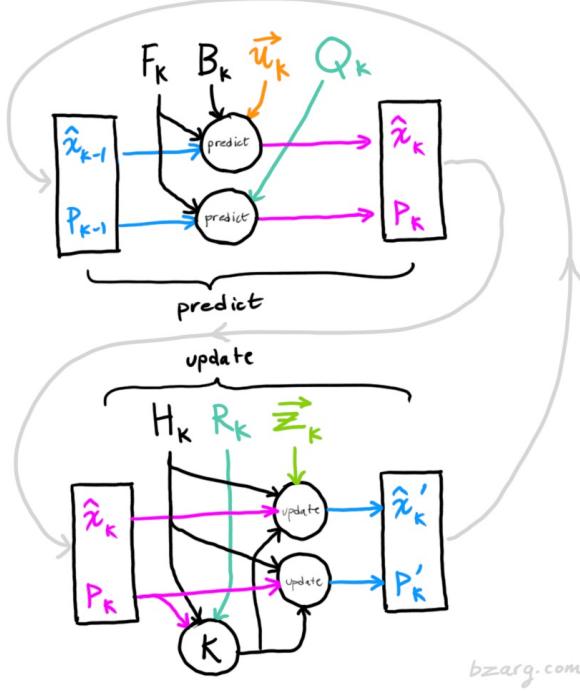
得 $\hat{X}'_k = \hat{X}_k + K'(\vec{z}_k - H_k \hat{X}_k)$

$$\hat{P}'_k = P_k - K' H_k P_k$$

$$\left\{ \begin{array}{l} K' = P_k H_k^T (H_k P_k H_k^T + R_k)^{-1} \\ \end{array} \right\}$$

\hat{X}'_k is new best estimate

Kalman Filter Information Flow



我们需要实现的是以上公式中 3 个

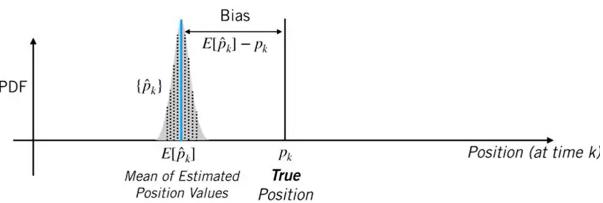
$$\textcircled{1} \quad \left\{ \begin{array}{l} \hat{x}_k = F_k \hat{x}_{k-1} + B_k \vec{u}_k \\ \hat{P}_k = F_k P_{k-1} F_k^T + Q_k. \end{array} \right.$$

$$\textcircled{2} \quad \left\{ \begin{array}{l} \hat{x}'_k = \hat{x}_k + K' (\vec{z}_k - H_k \hat{x}_k) \\ P'_k = P_k - K' H_k P_k. \end{array} \right.$$

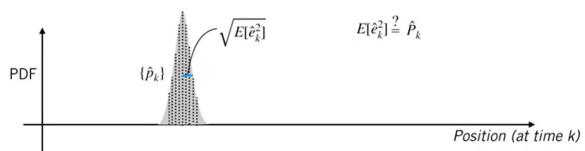
$$\textcircled{3} \quad K' = P_k H_k^T (H_k P_k H_k^T + R_k)^{-1}$$

Kalman Filter and The Bias BLUES

Bias in State Estimation



Consistency in State Estimation



- Consider the *error dynamics*:

$$\begin{aligned} \check{e}_k &= \hat{x}_k - x_k \\ \text{Predicted state error} & \quad \hat{e}_k = \hat{x}_k - x_k \\ & \quad \text{Corrected estimate error} \end{aligned}$$

- Using the Kalman Filter equations, we can derive:

$$\begin{aligned} \check{e}_k &= F_{k-1}\check{e}_{k-1} - w_k \\ \hat{e}_k &= (I - K_k H_k)\check{e}_k + K_k v_k \end{aligned}$$

- For the Kalman filter, for all k ,

$$\begin{aligned} E[\check{e}_k] &= E[F_{k-1}\check{e}_{k-1} - w_k] \\ &= F_{k-1}E[\check{e}_{k-1}] - E[w_k] \\ &= 0 \end{aligned} \quad \begin{aligned} E[\hat{e}_k] &= E[(I - K_k H_k)\check{e}_k + K_k v_k] \\ &= (I - K_k H_k)E[\check{e}_k] + K_k E[v_k] \\ &= 0 \end{aligned}$$

Unbiased predictions!

So long as $E[\hat{e}_0] = \mathbf{0}$ $E[v] = \mathbf{0}$ $E[w] = \mathbf{0}$
+ white, uncorrelated noise

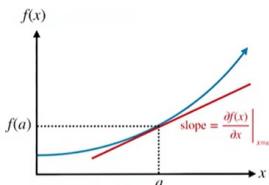
Note: this does *not* mean that the error on a *given* trial will be zero, but that, with enough trials, our expected error is zero!

(Non linear) Extended Kalman Filter.

c) E K F b

要点：对非线性系统进行线性化处理

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:

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

First-order terms

Higher-order terms

泰勒展開

只看第一回

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

Linearized motion model

$$x_k = f_{k-1}(x_{k-1}, u_{k-1}, w_{k-1}) \approx f_{k-1}(\hat{x}_{k-1}, u_{k-1}, 0) + \frac{\partial f_{k-1}}{\partial x_{k-1}} \Big|_{\hat{x}_{k-1}, u_{k-1}, 0} (x_{k-1} - \hat{x}_{k-1}) + \frac{\partial f_{k-1}}{\partial w_{k-1}} \Big|_{\hat{x}_{k-1}, u_{k-1}, 0} w_{k-1}$$

Linearized measurement model

$$y_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{v}_k) \approx \mathbf{h}_k(\check{\mathbf{x}}_k, \mathbf{0}) + \frac{\partial \mathbf{h}_k}{\partial \mathbf{x}_k} \Big|_{\check{\mathbf{x}}_k, \mathbf{0}} (\mathbf{x}_k - \check{\mathbf{x}}_k) + \frac{\partial \mathbf{h}_k}{\partial \mathbf{v}_k} \Big|_{\check{\mathbf{x}}_k, \mathbf{0}} \mathbf{v}_k$$

1313

$$f(x, y) = (x^4 + 3y^2 x, 5y^2 - 2xy + 1)$$

$$\frac{\partial f_1}{\partial x} = 4x^3 + 3y^2 \quad \frac{\partial f_1}{\partial y} = 6yx$$

$$\frac{\partial f_2}{\partial x} = -2y \quad \frac{\partial f_2}{\partial y} = 10^{-2}x$$

EKF I 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:

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

$$J_f(x,y) = \begin{pmatrix} \frac{\partial f_1}{\partial x}, & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x}, & \frac{\partial f_2}{\partial y} \end{pmatrix}$$

$$= \begin{pmatrix} 4x^3 + 3y^2 & 6y^4 \\ -2y & 10 - 2x \end{pmatrix}$$

13) 3:

$$J(x, y, z) = \left(x e^{2y} \cos(-z), (y-2)^3 \cdot \sin(\frac{z}{2}), e^{2y} \cdot \ln(\frac{x}{3}) \right)$$

$$\begin{pmatrix} \frac{\partial f_1}{\partial x}, & \frac{\partial f_1}{\partial y}, & \frac{\partial f_1}{\partial z}, \\ \frac{\partial f_2}{\partial x}, & \frac{\partial f_2}{\partial y}, & \frac{\partial f_2}{\partial z}, \\ \frac{\partial f_3}{\partial x}, & \frac{\partial f_3}{\partial y}, & \frac{\partial f_3}{\partial z} \end{pmatrix}$$

$$\frac{\partial f_1}{\partial x} = e^{2y} \cos(-z)$$

$$\frac{\partial f_1}{\partial y} = 2e^{2y} x \cos(-z)$$

$$\frac{\partial f_1}{\partial z} = \sin(-z) e^{2y} x$$

,

,

,

EKF | Putting it all together

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

Linearized motion model

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

Prediction

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

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

Linearized measurement model

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

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

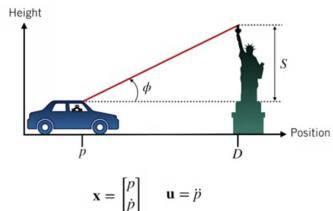
$\hat{\mathbf{x}}_k$ Prediction
(given motion model)
at time k

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

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

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

EKF | Short example



Motion/Process model

$$\begin{aligned} \mathbf{x}_k &= \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}) \\ &= \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \hat{\mathbf{x}}_{k-1} + \begin{bmatrix} 0 \\ \Delta t \end{bmatrix} \mathbf{u}_{k-1} + \mathbf{w}_{k-1} \end{aligned}$$

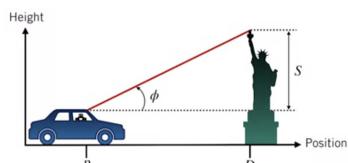
Landmark measurement model

$$\begin{aligned} \mathbf{y}_k &= \phi_k = h(p_k, v_k) \\ &= \tan^{-1} \left(\frac{S}{D - p_k} \right) + v_k \end{aligned}$$

Noise densities

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

EKF | Short example



Motion model Jacobians

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

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

Measurement model Jacobians

$$\mathbf{H}_k = \frac{\partial h}{\partial \hat{\mathbf{x}}_k} \Big|_{\hat{\mathbf{x}}_k, \mathbf{0}} = \begin{bmatrix} \frac{S}{(D - p_k)^2 + S^2} & 0 \end{bmatrix}$$

$$M_k = \frac{\partial h}{\partial v_k} \Big|_{\hat{\mathbf{x}}_k, \mathbf{0}} = 1$$

Prediction

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

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

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

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

Data

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

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

$$u_0 = -2 \text{ [m/s}^2\text{]} \quad y_1 = \pi/6 \text{ [rad]}$$

$$S = 20 \text{ [m]} \quad D = 40 \text{ [m]}$$

Kalman Gain

Correction

$$\begin{aligned} \mathbf{K}_1 &= \check{\mathbf{P}}_1 \mathbf{H}_1^T (\mathbf{H}_1 \check{\mathbf{P}}_1 \mathbf{H}_1^T + \mathbf{M}_1 \mathbf{R}_1 \mathbf{M}_1^T)^{-1} \\ &= \begin{bmatrix} 0.36 & 0.5 \\ 0.5 & 1.1 \end{bmatrix} \begin{bmatrix} 0.011 & 0 \\ 0 & 1 \end{bmatrix} \left(\begin{bmatrix} 0.011 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0.36 & 0.5 \\ 0.5 & 1.1 \end{bmatrix} \begin{bmatrix} 0.011 & 0 \\ 0 & 1 \end{bmatrix} + 1(0.01)(1) \right)^{-1} \\ &= \begin{bmatrix} 0.40 \\ 0.55 \end{bmatrix} \end{aligned}$$

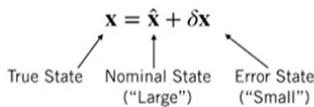
$$\hat{\mathbf{x}}_1 = \check{\mathbf{x}}_1 + \mathbf{K}_1 (\mathbf{y}_1 - \mathbf{h}_1(\check{\mathbf{x}}_1, \mathbf{0}))$$

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

$$\begin{aligned} \hat{\mathbf{P}}_1 &= (1 - \mathbf{K}_1 \mathbf{H}_1) \check{\mathbf{P}}_1 \\ &= \begin{bmatrix} 0.36 & 0.5 \\ 0.5 & 1.1 \end{bmatrix} \end{aligned}$$

並傳

$$\text{Error State} = EKF.$$



1. Update nominal state with motion model

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

↑
This could be
 $\check{\mathbf{x}}_{k-1}$ or $\hat{\mathbf{x}}_{k-1}$

2. Propagate uncertainty

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

↑
This could be
 $\check{\mathbf{P}}_{k-1}$ or $\hat{\mathbf{P}}_{k-1}$

3. If a measurement is available:

1. Compute Kalman Gain

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

2. Compute error state

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

3. Correct nominal state

$$\hat{\mathbf{x}}_k = \check{\mathbf{x}}_k + \delta\hat{\mathbf{x}}_k$$

4. Correct state covariance

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

Project 1

Motion Model

The vehicle motion model receives linear and angular velocity odometry readings as inputs, and outputs the state (i.e., the 2D pose) of the vehicle:

$$\mathbf{x}_k = \mathbf{x}_{k-1} + T \begin{bmatrix} \cos \theta_{k-1} & 0 \\ \sin \theta_{k-1} & 0 \\ 0 & 1 \end{bmatrix} \left(\begin{bmatrix} v_k \\ \omega_k \end{bmatrix} + \mathbf{w}_k \right), \quad \mathbf{w}_k = \mathcal{N}(\mathbf{0}, \mathbf{Q})$$

- $\mathbf{x}_k = [x \ y \ \theta]^T$ is the current 2D pose of the vehicle
- v_k and ω_k are the linear and angular velocity odometry readings, which we use as inputs to the model

The process noise \mathbf{w}_k has a (zero mean) normal distribution with a constant covariance \mathbf{Q} .

Measurement Model



The measurement model relates the current pose of the vehicle to the LIDAR range and bearing measurements $\mathbf{y}_k^l = [r \ \phi]^T$.

$$\mathbf{y}_k^l = \begin{bmatrix} \sqrt{(x_l - x_k - d \cos \theta_k)^2 + (y_l - y_k - d \sin \theta_k)^2} \\ \text{atan2}(y_l - y_k - d \sin \theta_k, x_l - x_k - d \cos \theta_k) - \theta_k \end{bmatrix} + \mathbf{n}_k^l, \quad \mathbf{n}_k^l = \mathcal{N}(\mathbf{0}, \mathbf{R})$$

- x_l and y_l are the ground truth coordinates of the landmark l
- x_k and y_k and θ_k represent the current pose of the vehicle
- d is the known distance between robot center and laser rangefinder (LIDAR)

The landmark measurement noise \mathbf{n}_k^l has a (zero mean) normal distribution with a constant covariance \mathbf{R} .

$$\mathbf{y} = \begin{bmatrix} r \\ \phi \end{bmatrix}$$

- Compute the measurement model Jacobians at $\tilde{\mathbf{x}}_k$

$$\mathbf{y}_k^l = \mathbf{h}(\mathbf{x}_k, \mathbf{n}_k^l)$$

$$\mathbf{H}_k = \frac{\partial \mathbf{h}}{\partial \mathbf{x}_k} \Big|_{\tilde{\mathbf{x}}_k, 0}, \quad \mathbf{M}_k = \frac{\partial \mathbf{h}}{\partial \mathbf{n}_k} \Big|_{\tilde{\mathbf{x}}_k, 0}$$

- Compute the Kalman Gain

$$\mathbf{H}_k = \begin{bmatrix} \frac{\partial r}{\partial x_k} & \frac{\partial r}{\partial y_k} & \frac{\partial r}{\partial \theta_k} \\ \frac{\partial \phi}{\partial x_k} & \frac{\partial \phi}{\partial y_k} & \frac{\partial \phi}{\partial \theta_k} \end{bmatrix}$$

$$\frac{\partial r}{\partial x_k} = \frac{x_k + d \cos(\theta_k) - x_l}{r}$$