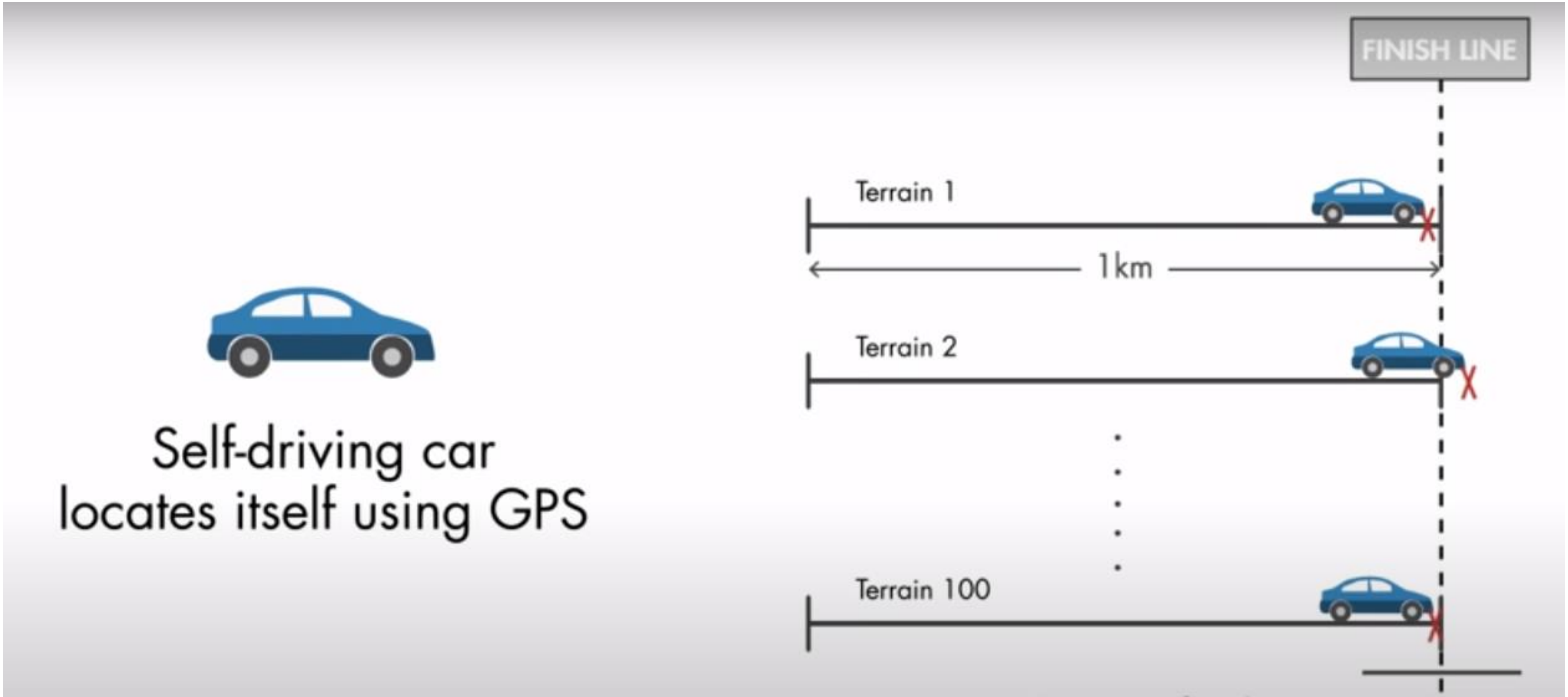
Lecture on Kalman Filter

ESP 3201

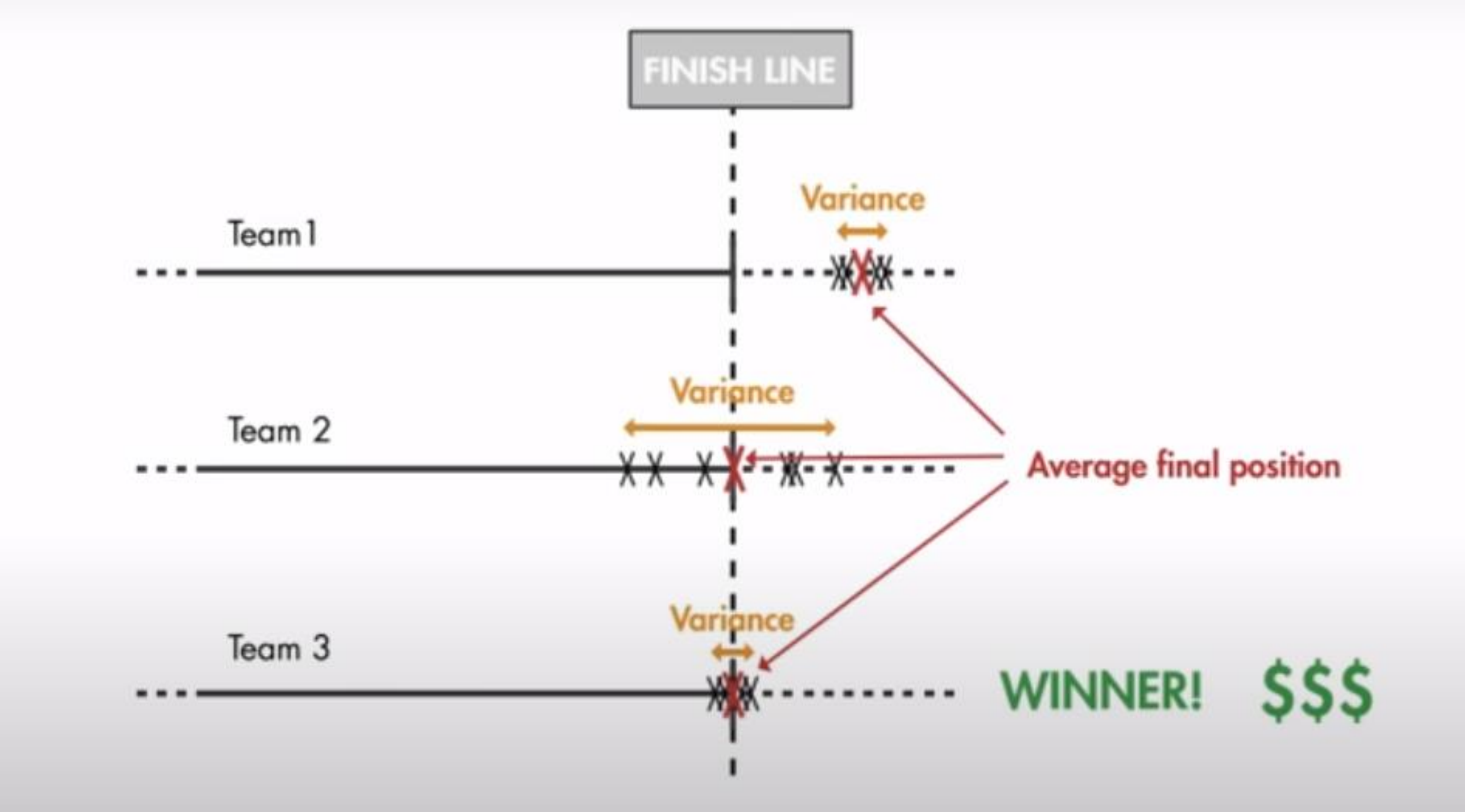
Dr Ng Gee Wah

National University of Singapore

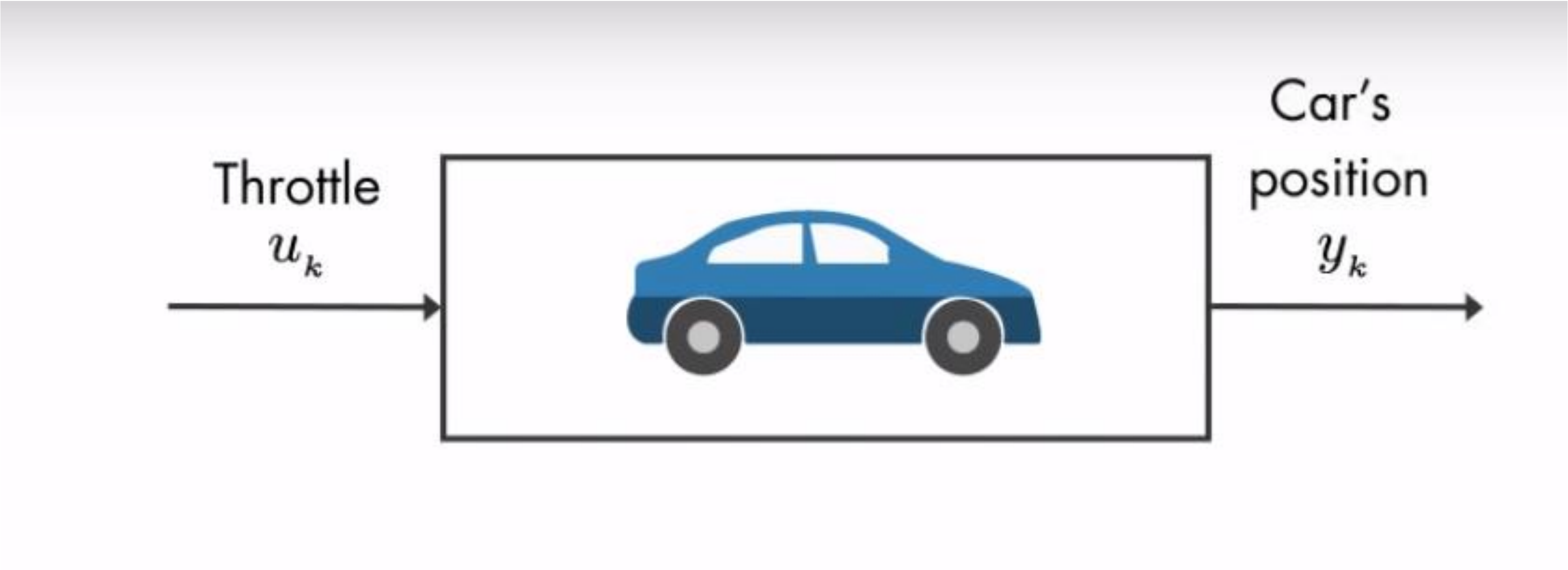
Using Kalman Filter to estimate the position



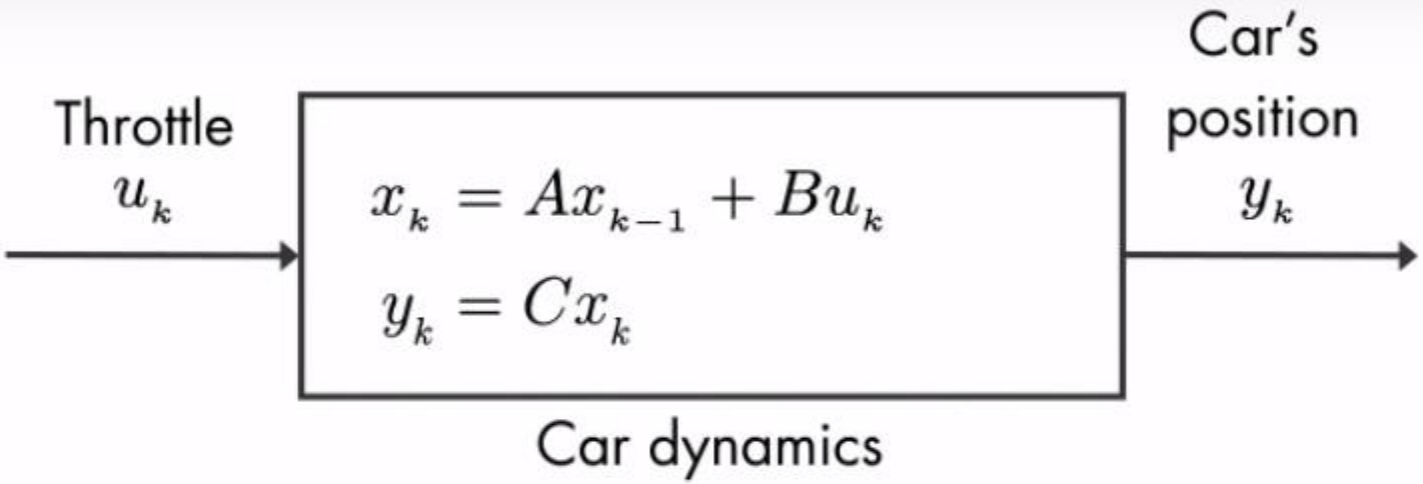
The estimated position has to be with the smallest variance and best final position



Mathematical model of the car

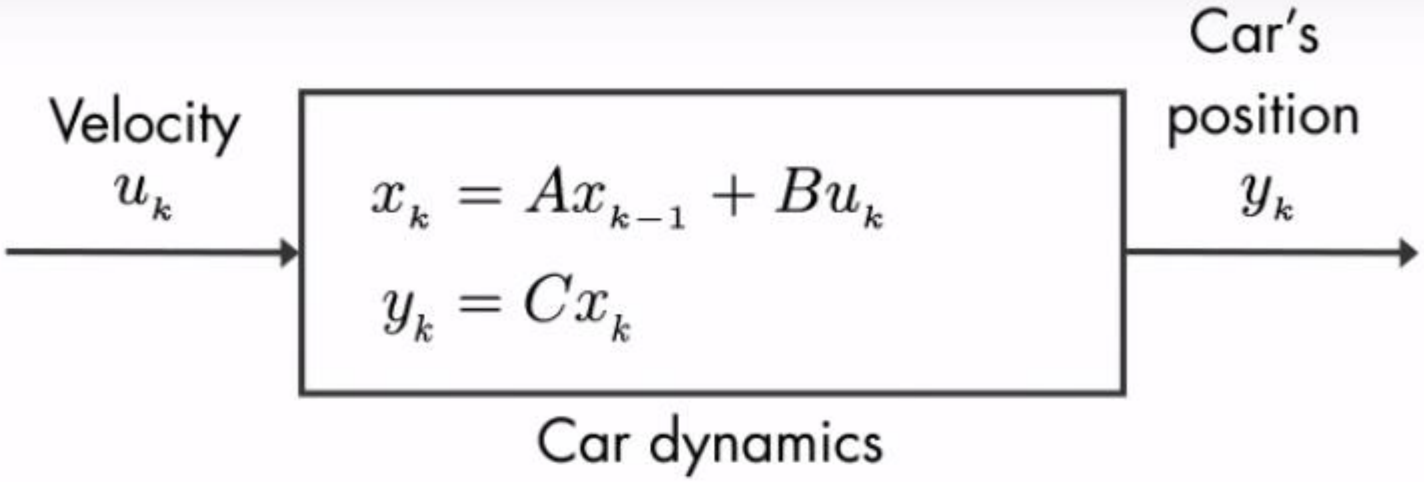


Mathematical model of the car



$$x_k = \begin{bmatrix} velocity \\ position \end{bmatrix}$$

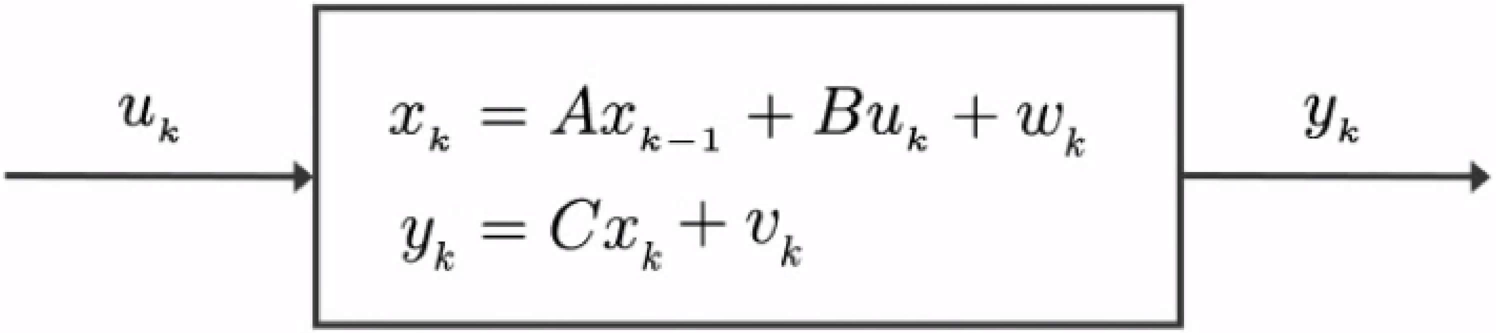
Mathematical model of the car



$$x_k = [position]$$

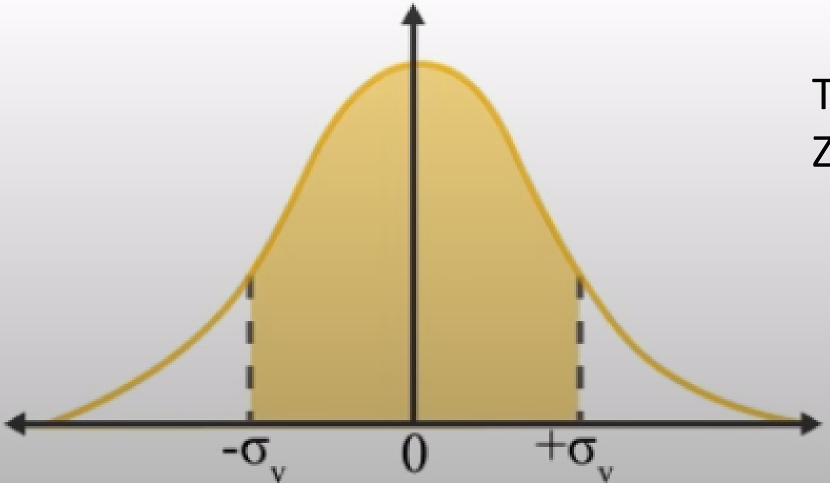
$$C = 1$$

Mathematical model of the car



Car dynamics

$$v \sim N(0, R)$$



W – process noise
(e.g. wind, changes in car velocity)
V - measurement noise (e.g. instrument error);

Typically use Gaussian distribution.
Zero mean and covariance R

Mathematical model of the car
Optimal State Estimate

$u_k \rightarrow$

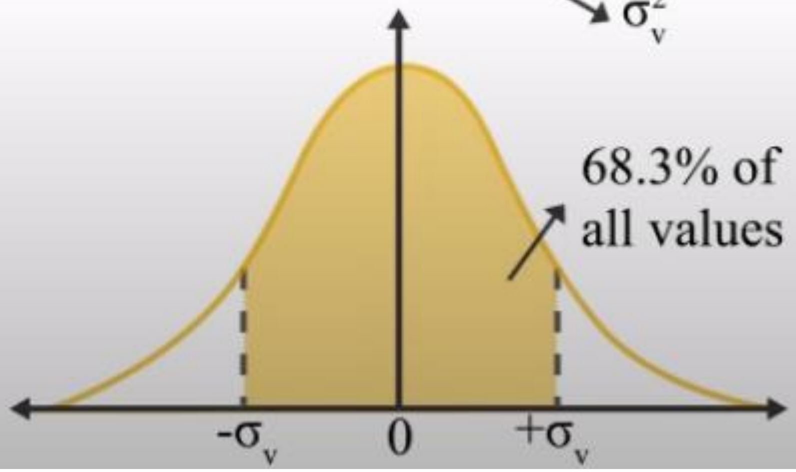
$$x_k = Ax_{k-1} + Bu_k + w_k$$
$$y_k = Cx_k + v_k$$

$\rightarrow y_k$

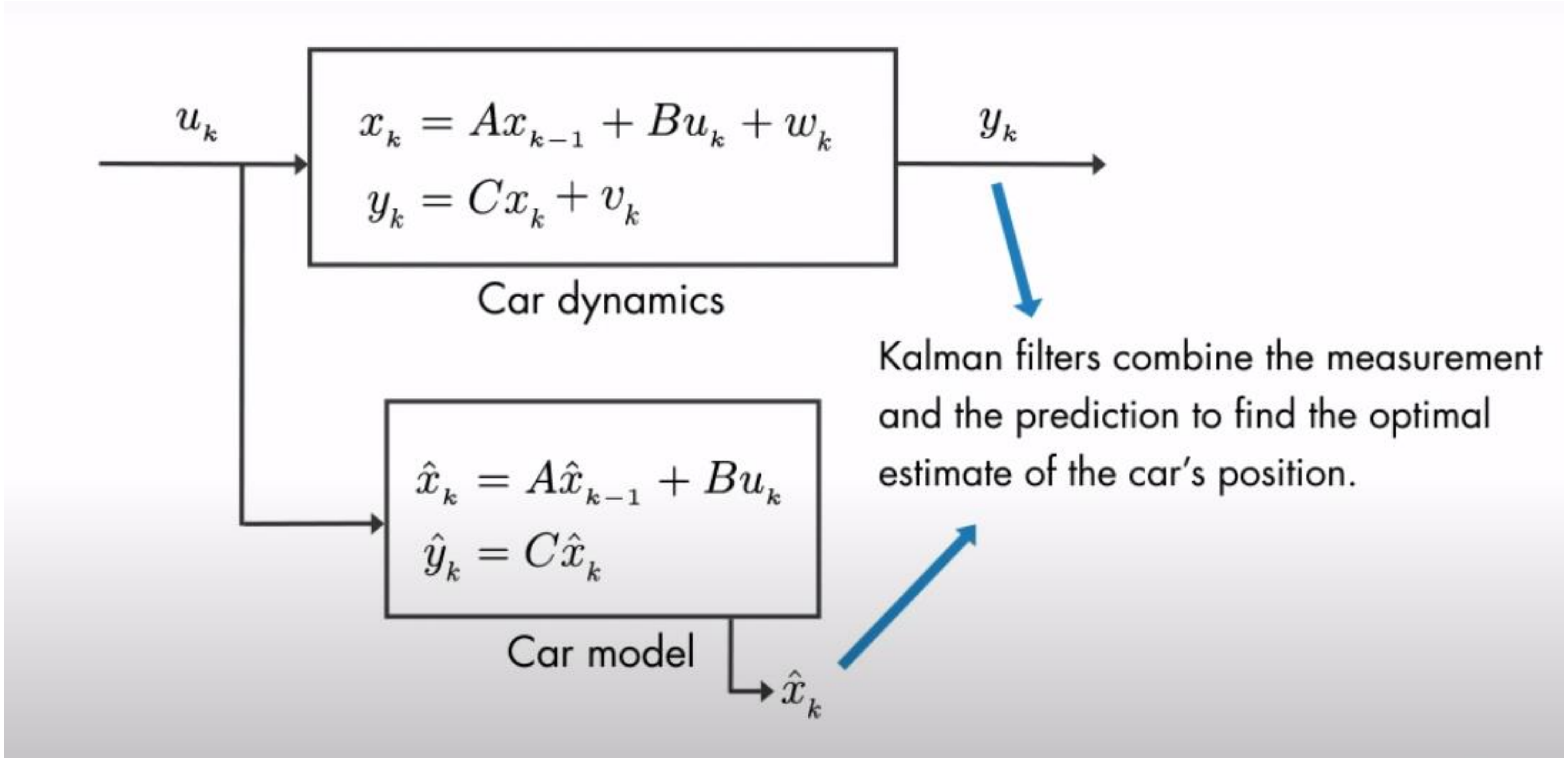
Car dynamics

$v \sim N(0, R) \rightarrow \sigma_v^2$

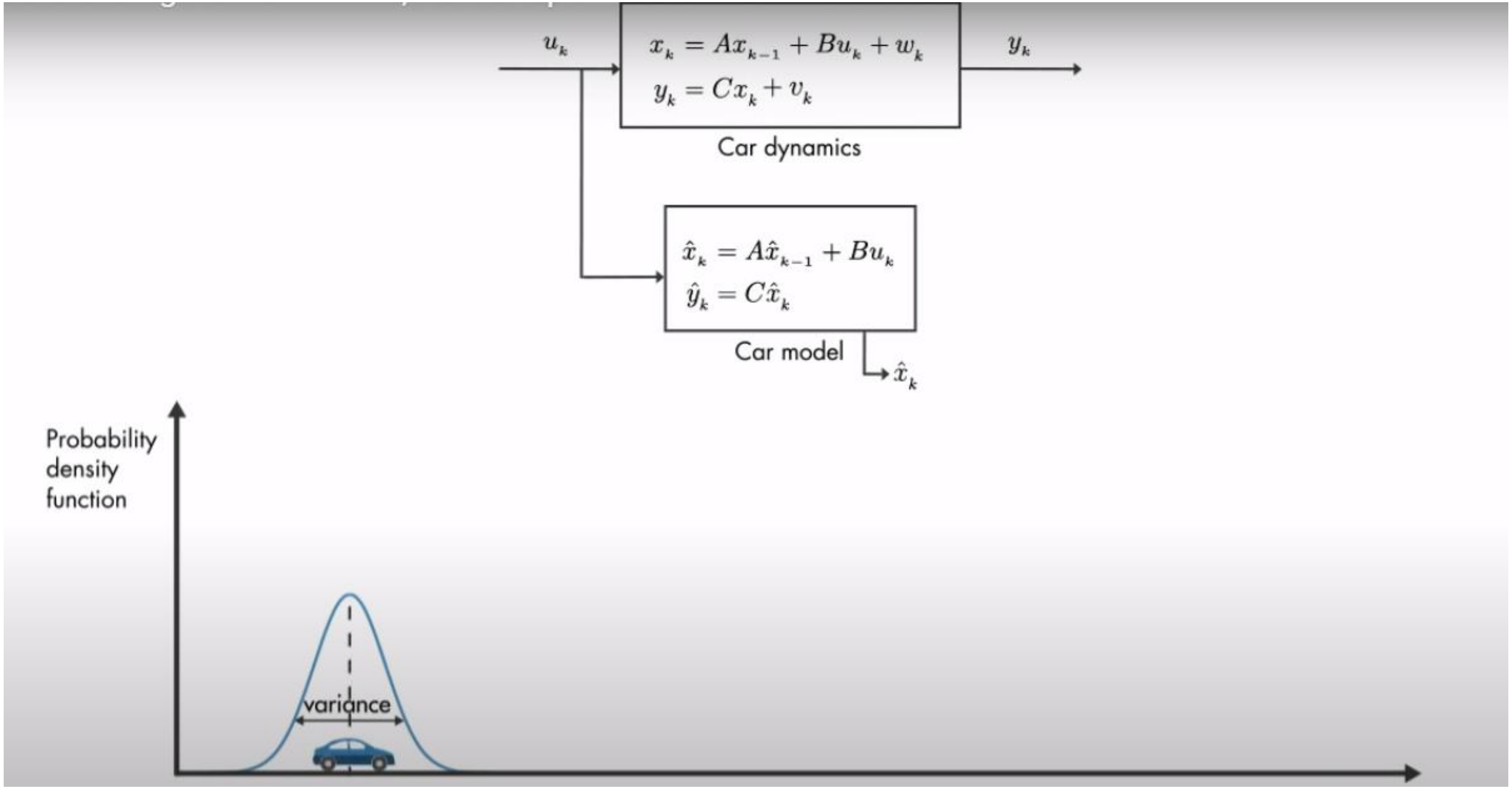
$w \sim N(0, Q) \rightarrow \sigma_w^2$



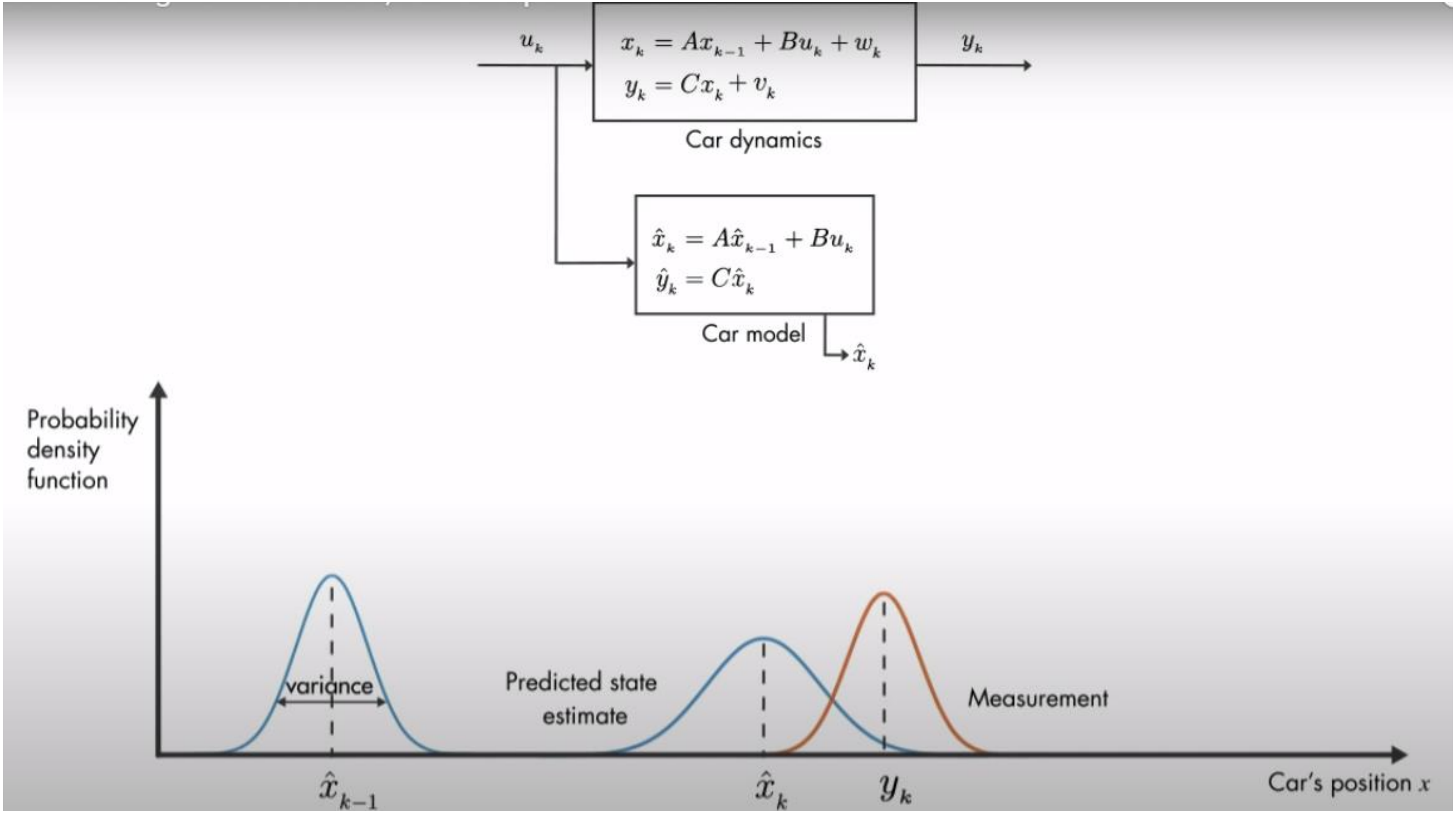
Mathematical model of the car
Optimal State Estimate



Mathematical model of the car
Optimal State Estimate

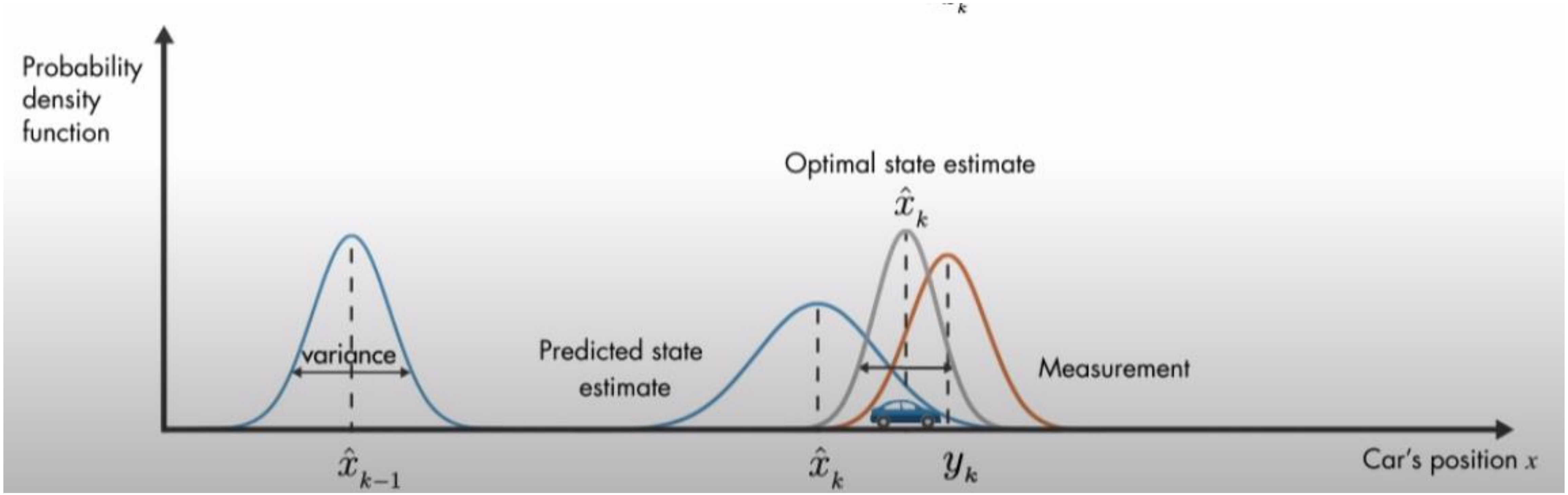


X_{k-1}



u_k
$x_k = Ax_{k-1} + Bu_k + w_k$
$y_k = Cx_k + v_k$
y_k
Car dynamics
$\hat{x}_k = A\hat{x}_{k-1} + Bu_k$
$\hat{y}_k = C\hat{x}_k$
Car model
$\hat{x}_k$
Probability density function
variance
Predicted state estimate
Measurement
$\hat{x}_{k-1}$
$\hat{x}_k$
y_k
Car's position x

Mathematical model of the car
Optimal State Estimate



Implementing the Kalman Filter Algorithm

State observer	$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + K(y_k - C\hat{x}_k)$
Kalman filter	$\hat{x}_k = A\hat{x}_{k-1} + Bu_k + K_k(y_k - C(A\hat{x}_{k-1} + Bu_k))$

Notice that the Kalman filter and the state observer equation (see earlier slides) are the same.
Kalman filter is a type of state observer. The state observer is a deterministic system and Kalman filter is a stochastic system

Implementing the Kalman Filter Algorithm

NUS National University of Singapore

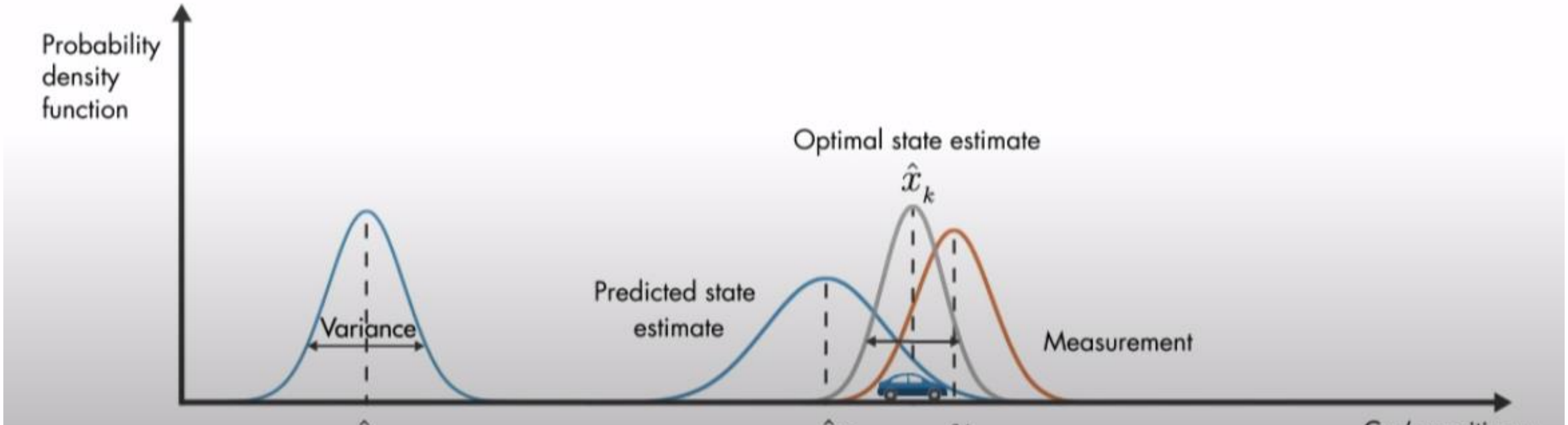
Kalman filter $$\hat{x}_k = \underbrace{A\hat{x}_{k-1} + Bu_k}_{\hat{x}_k^- \text{ : A Priori Estimate}} + K_k(y_k - C(A\hat{x}_{k-1} + Bu_k))$$

Kalman filter $$\hat{x}_k = \hat{x}_k^- + K_k(y_k - C\hat{x}_k^-)$$

Implementing the Kalman Filter Algorithm

Kalman filter

$$\overbrace{\hat{x}_k}^{\text{A Posteriori Estimate}} = \underbrace{\hat{x}_k^-}_{\text{Predict}} + \underbrace{K_k(y_k - C\hat{x}_k^-)}_{\text{Update}}$$



Implementing the Kalman Filter Algorithm

National University of Singapore

Prediction

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$$

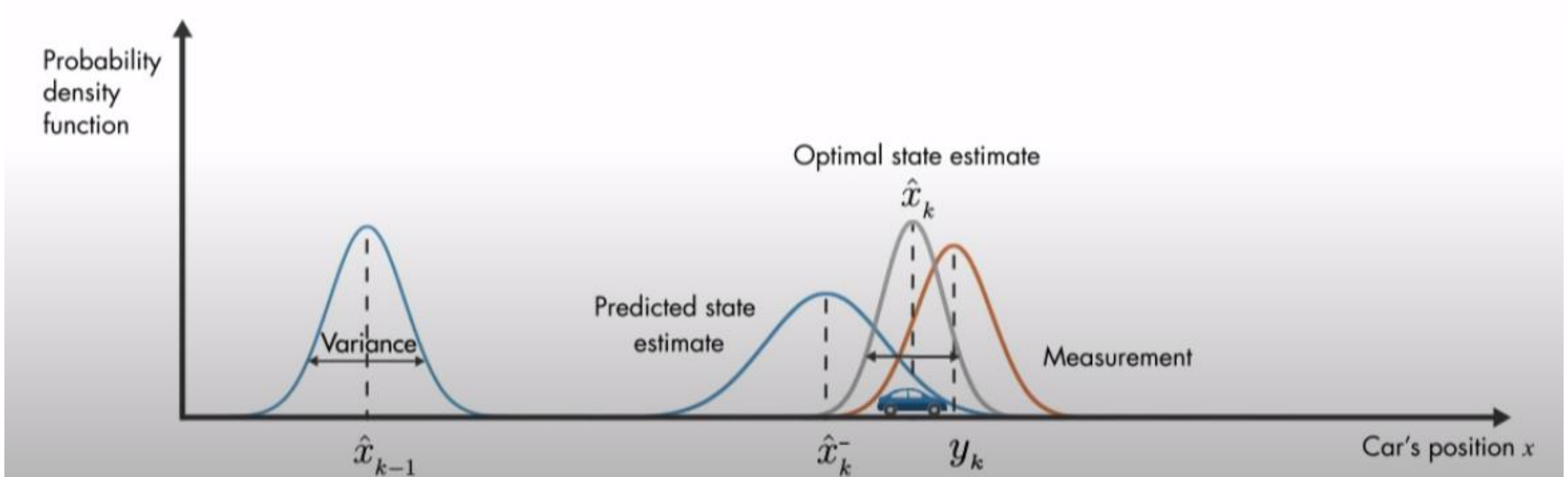
$$P_k^- = AP_{k-1}A^T + Q$$

Update

$$K_k = \frac{P_k^- C^T}{CP_k^- C^T + R}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - C\hat{x}_k^-)$$

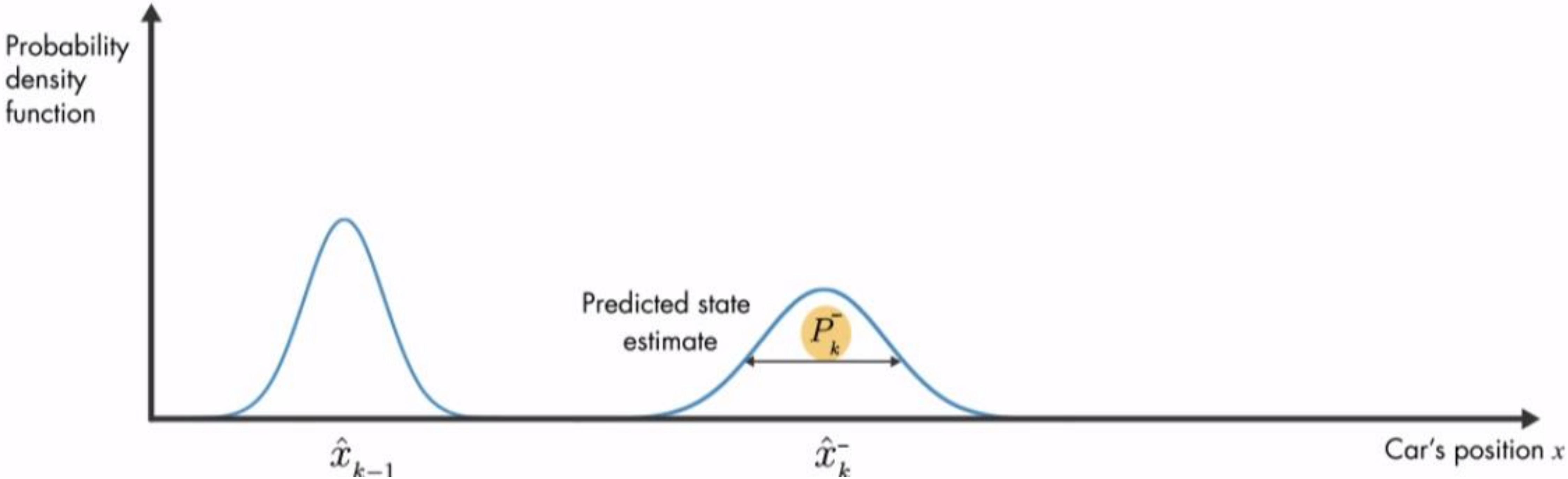
$$P_k = (I - K_k C)P_k^-$$



Prediction

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$$

$$P_k^- = AP_{k-1}A^T + Q$$



Prediction

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$$

$$P_k^- = AP_{k-1}A^T + Q$$

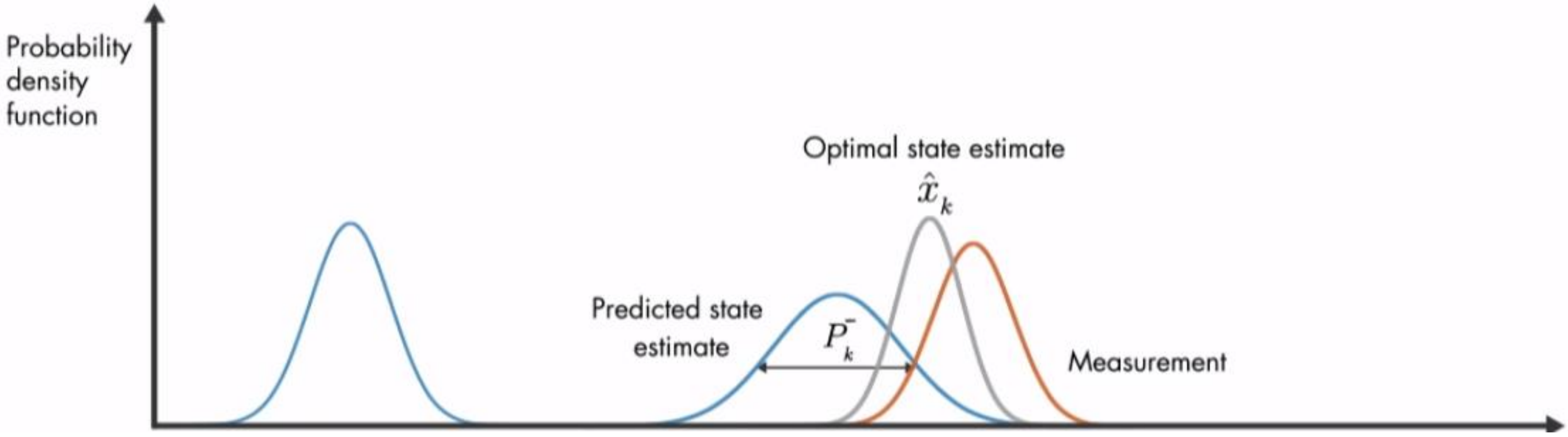
Initial estimates for $\hat{x}_{k-1}$ and P_{k-1}

Update

$$K_k = \frac{P_k^- C^T}{CP_k^- C^T + R}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - C\hat{x}_k^-)$$

$$P_k = (I - K_k C)P_k^-$$



Prediction

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$$

$$P_k^- = AP_{k-1}A^T + Q$$

Initial estimates for $\hat{x}_{k-1}$ and P_{k-1}

Update

$$K_k = \frac{P_k^- C^T}{CP_k^- C^T + R}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - C\hat{x}_k^-)$$

$$P_k = (I - K_k C)P_k^-$$

Prediction

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$$

$$P_k^- = AP_{k-1}A^T + Q$$

Initial estimates for $\hat{x}_{k-1}$ and P_{k-1}

Update

$$K_k = \frac{P_k^- C^T}{CP_k^- C^T + R}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - C\hat{x}_k^-)$$

$$P_k = (I - K_k C)P_k^-$$

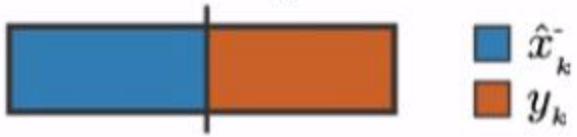
Update

$$K_k = \frac{P_k^- C^T}{C P_k^- C^T + R}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - C\hat{x}_k^-)$$

$$P_k = (I - K_k C)P_k^-$$

Calculation of $\hat{x}_k$



Update

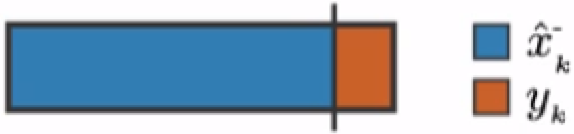
$$K_k = \frac{P_k^- C^T}{C P_k^- C^T + R}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - C\hat{x}_k^-)$$

$$P_k = (I - K_k C)P_k^-$$

Calculation of $\hat{x}_k$

Calculation of $\hat{x}_k$



Update

$$K_k = \frac{P_k^- C^T}{C P_k^- C^T + R}$$

$$\hat{x}_k = \hat{x}_k^- + K_k (y_k - C\hat{x}_k^-)$$

$$P_k = (I - K_k C) P_k^-$$

Calculation of $\hat{x}_k$

$$\lim_{R\to 0} K_k = \lim_{R\to 0} \frac{P_k^- C^T}{C P_k^- C^T + R} = \lim_{R\to 0} \frac{P_k^- C^T}{C P_k^- C^T + 0}$$

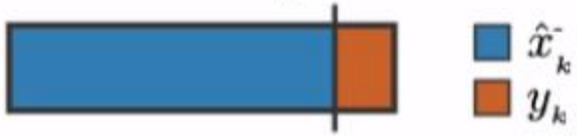
Update

$$K_k = \frac{P_k^- C^T}{C P_k^- C^T + R}$$

$$\hat{x}_k = \hat{x}_k^- + K_k (y_k - C\hat{x}_k^-)$$

$$P_k = (I - K_k C) P_k^-$$

Calculation of $\hat{x}_k$



$$\lim_{R\to 0} K_k = \lim_{R\to 0} \frac{P_k^- C^T}{C P_k^- C^T + R} = \lim_{R\to 0} \frac{\cancel{P_k^- C^T}}{C \cancel{P_k^- C^T} + 0} = C^{-1} \qquad C^{-1} = 1$$

$$\hat{x}_k = \hat{x}_k^- + K_k (y_k - C\hat{x}_k^-) = \hat{x}_k^- + C^{-1}(y_k + C\hat{x}_k^-)$$

$$= \cancel{\hat{x}_k^-} + C^{-1} y_k + \cancel{C^{-1} C \hat{x}_k^-}$$

Update

$$K_k = \frac{P_k^- C^T}{C P_k^- C^T + R}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - C\hat{x}_k^-)$$

$$P_k = (I - K_k C)P_k^-$$

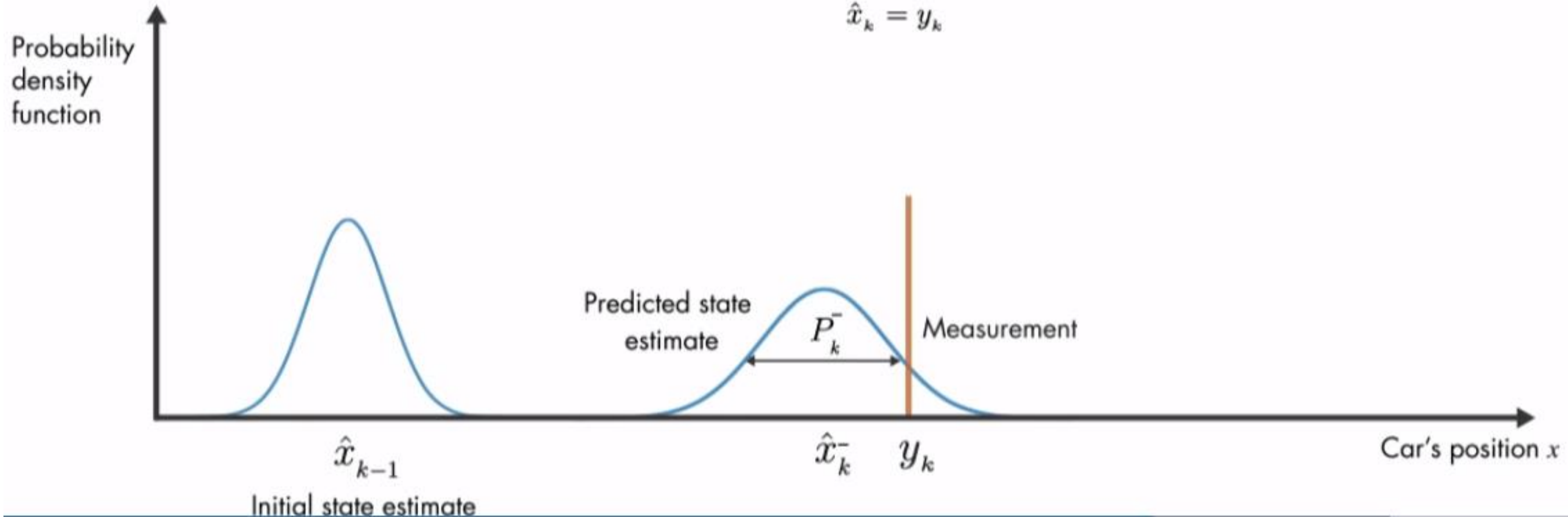
Calculation of $\hat{x}_k$

$$\lim_{R\to 0} K_k = \lim_{R\to 0} \frac{P_k^- C^T}{C P_k^- C^T + R} = \lim_{R\to 0} \frac{\cancel{P_k^- C^T}}{C \cancel{P_k^- C^T} + 0} = C^{-1} \qquad C^{-1} = 1$$

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - C\hat{x}_k^-) = \hat{x}_k^- + C^{-1}(y_k + C\hat{x}_k^-)$$

$$= \cancel{\hat{x}_k^-} + C^{-1}y_k + \cancel{C^{-1}C\hat{x}_k^-}$$

$$\hat{x}_k = y_k$$



Update

$$K_k = \frac{P_k^- C^T}{C P_k^- C^T + R}$$

$$\hat{x}_k = \hat{x}_k^- + K_k (y_k - C\hat{x}_k^-)$$

$$P_k = (I - K_k C) P_k^-$$

Calculation of $\hat{x}_k$

$$\lim_{P_k^- \to 0} K_k = \lim_{P_k^- \to 0} \frac{P_k^- C^T}{C P_k^- C^T + R} = \lim_{P_k^- \to 0} \frac{0}{0 + R} = 0$$

$$\hat{x}_k = \hat{x}_k^- + K_k (y_k - C\hat{x}_k^-) = \hat{x}_k^- + 0(y_k + C\hat{x}_k^-)$$

Update

$$K_k = \frac{P_k^- C^T}{C P_k^- C^T + R}$$

$$\hat{x}_k = \hat{x}_k^- + K_k (y_k - C\hat{x}_k^-)$$

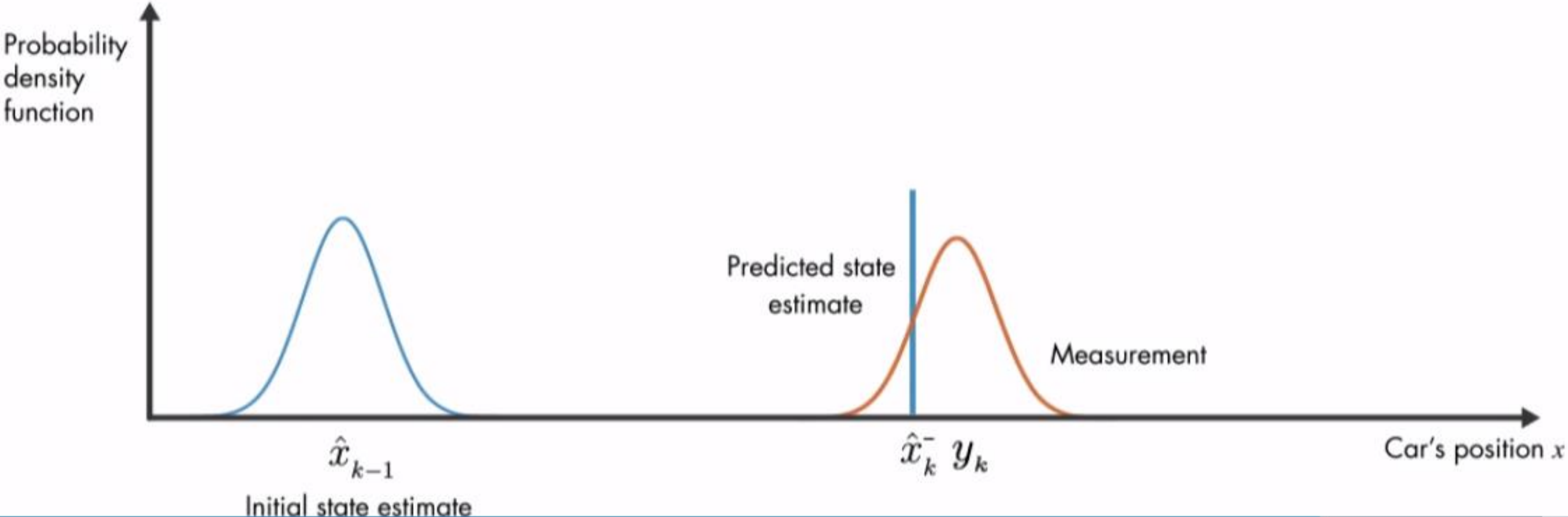
$$P_k = (I - K_k C) P_k^-$$

Calculation of $\hat{x}_k$

$$\lim_{P_k^- \to 0} K_k = \lim_{P_k^- \to 0} \frac{P_k^- C^T}{C P_k^- C^T + R} = \lim_{P_k^- \to 0} \frac{0}{0 + R} = 0$$

$$\hat{x}_k = \hat{x}_k^- + K_k (y_k - C\hat{x}_k^-) = \hat{x}_k^- + 0(y_k + C\hat{x}_k^-)$$

$$\hat{x}_k = \hat{x}_k^-$$



Update

$$K_k = \frac{P_k^- C^T}{C P_k^- C^T + R}$$

$$\hat{x}_k = \hat{x}_k^- + K_k (y_k - C\hat{x}_k^-)$$

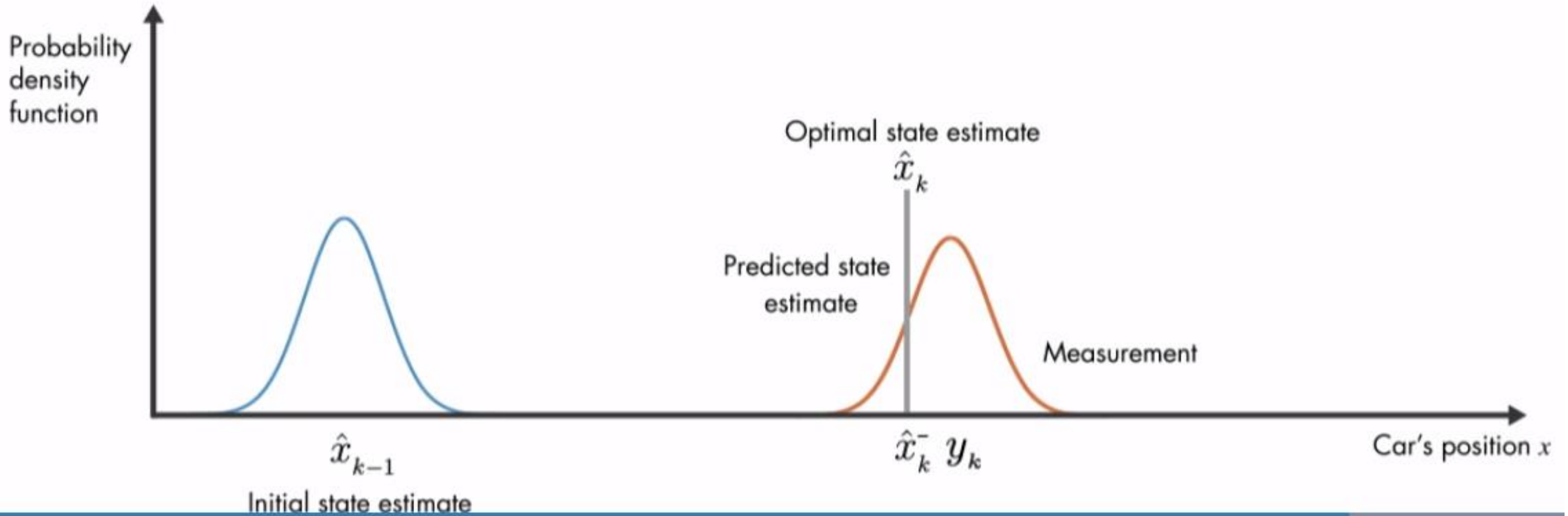
$$P_k = (I - K_k C) P_k^-$$

Calculation of $\hat{x}_k$

$$\lim_{P_k^- \to 0} K_k = \lim_{P_k^- \to 0} \frac{P_k^- C^T}{C P_k^- C^T + R} = \lim_{P_k^- \to 0} \frac{0}{0 + R} = 0$$

$$\hat{x}_k = \hat{x}_k^- + K_k (y_k - C\hat{x}_k^-) = \hat{x}_k^- + 0(y_k + C\hat{x}_k^-)$$

$$\hat{x}_k = \hat{x}_k^-$$



Prediction

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$$

$$P_k^- = AP_{k-1}A^T + Q$$

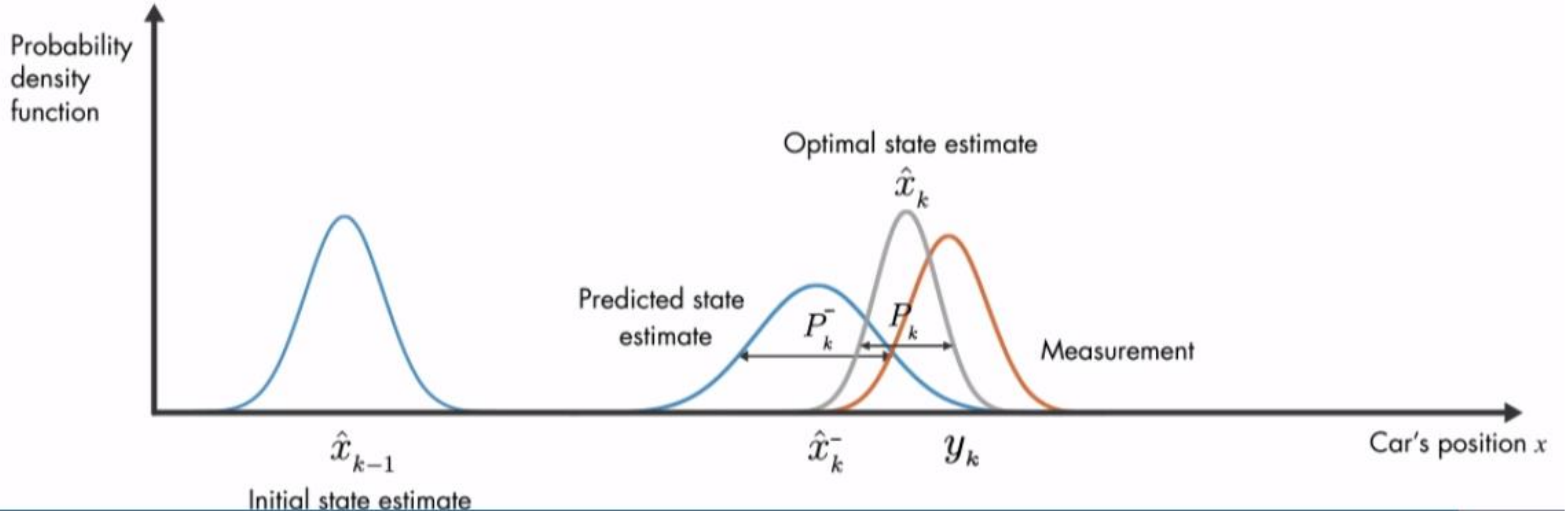
Initial estimates for $\hat{x}_{k-1}$ and P_{k-1}

Update

$$K_k = \frac{P_k^- C^T}{CP_k^- C^T + R}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - C\hat{x}_k^-)$$

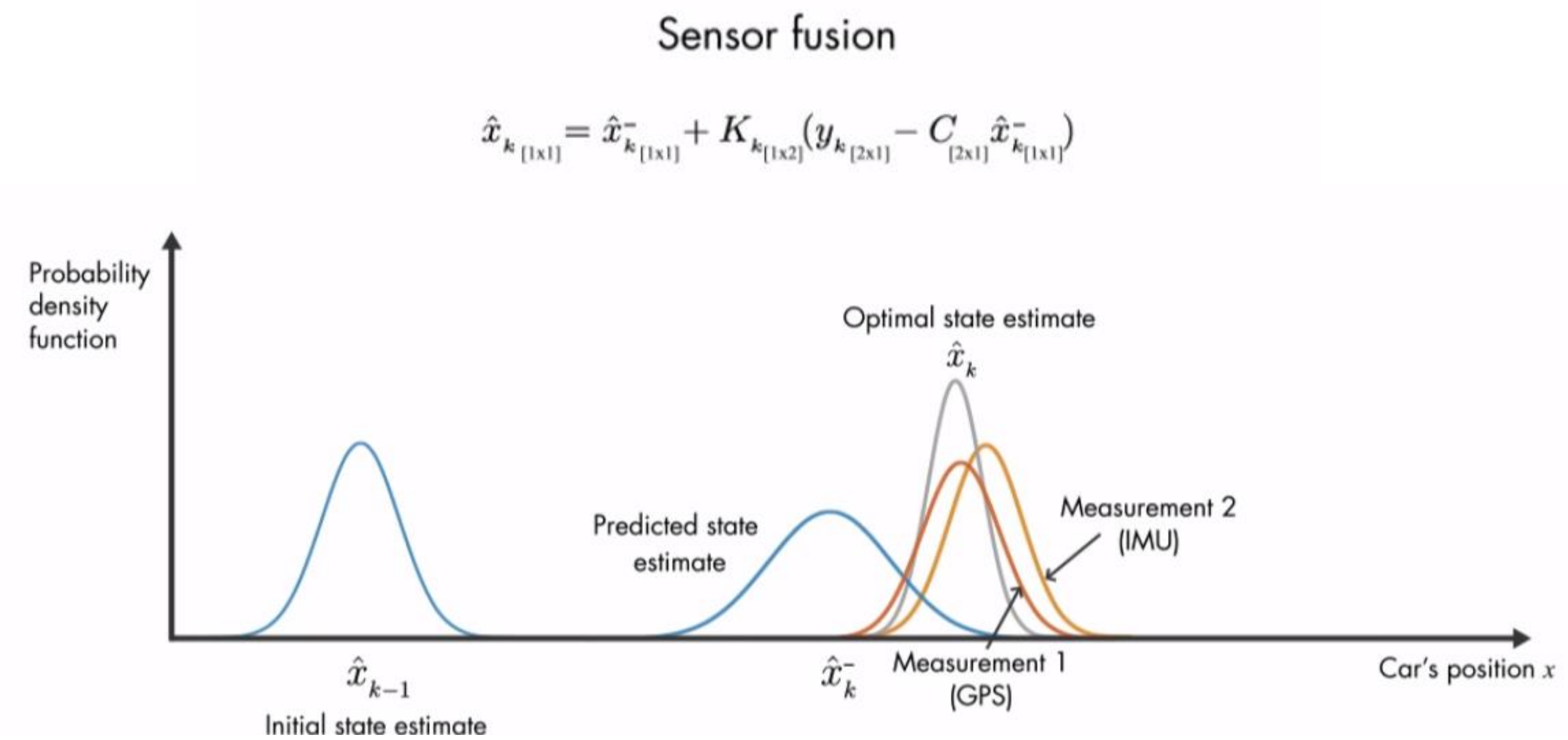
$$P_k = (I - K_k C)P_k^-$$



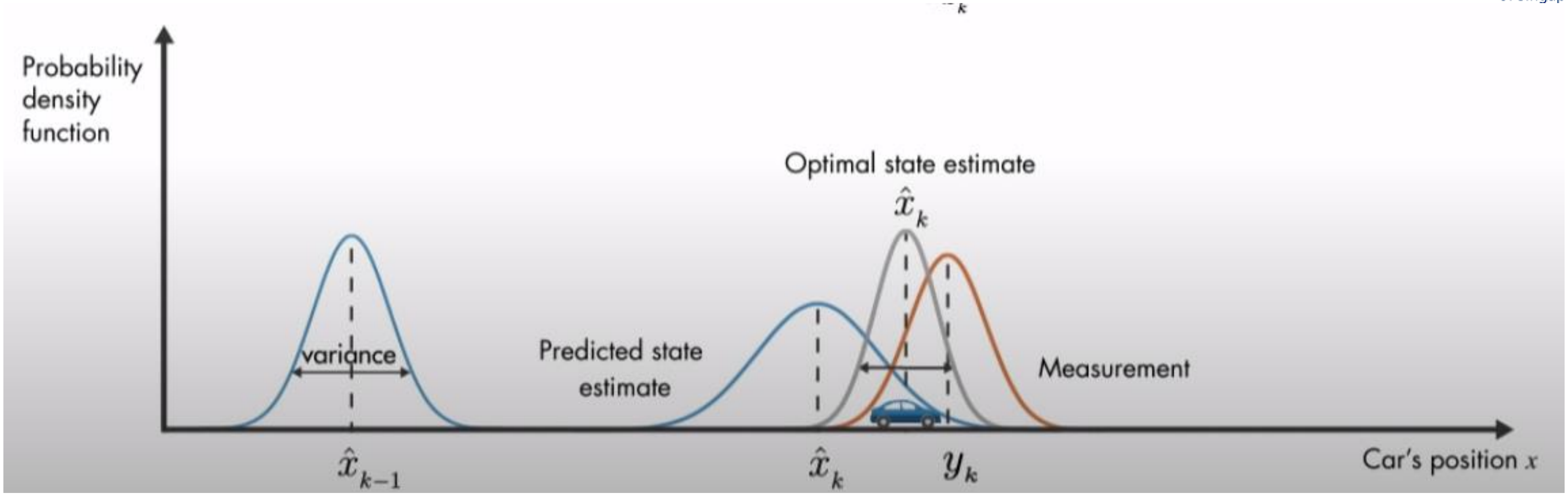
How Kalman filter can be used as for sensor fusion?

What is sensor fusion?

Kalman Filter Algorithm for Sensor Fusion



Optimal Kalman Filter estimator



However, most real world problems are nonlinear i.e. how can we use Kalman filter in non-linear systems?

Kalman Filter Algorithm for Nonlinear System

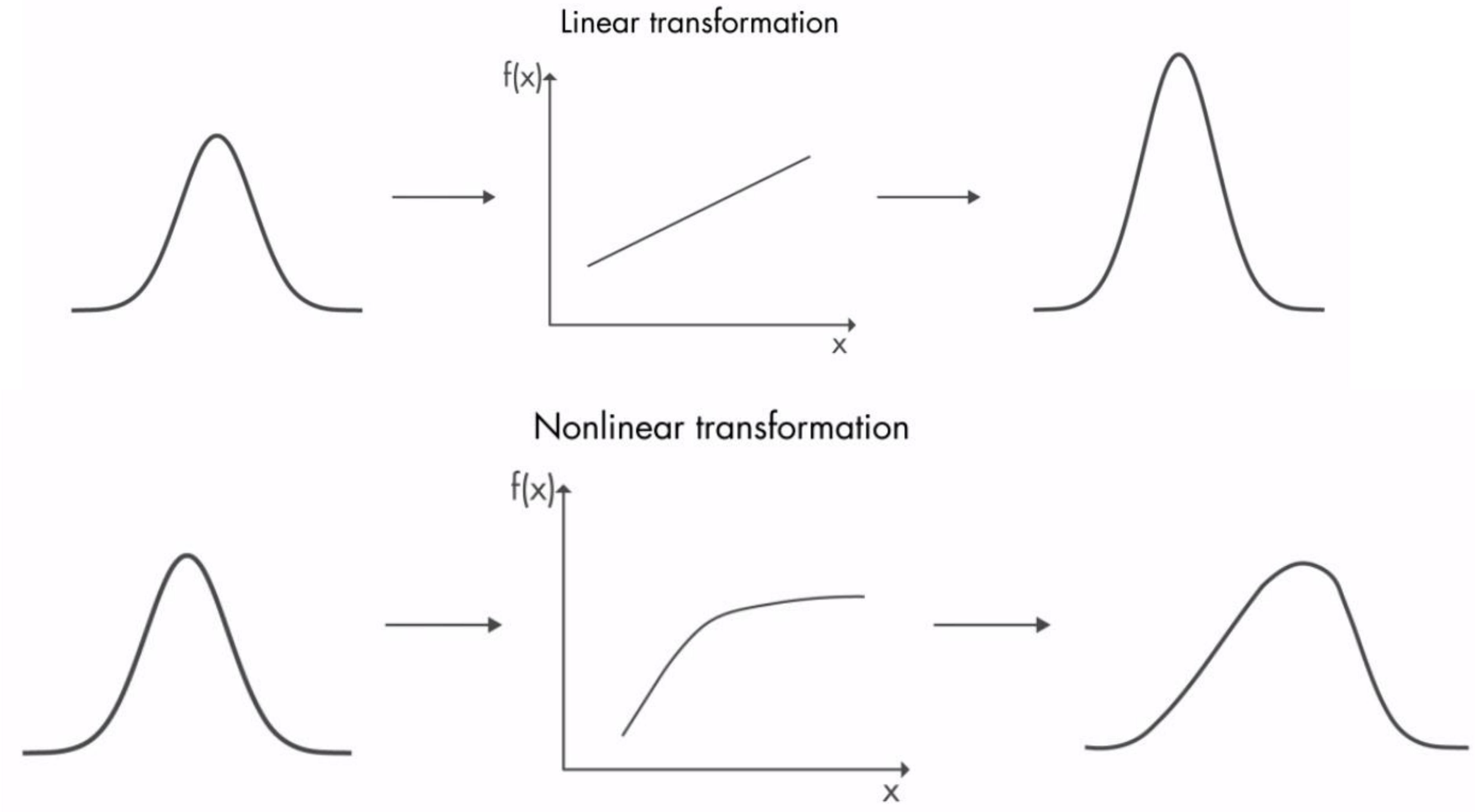
Kalman filters are defined for linear systems.

$$x_k = Ax_{k-1} + Bu_k + w_k$$
$$y_k = Cx_k + v_k$$

$$x_k = f(x_{k-1}, u_k) + w_k$$
$$y_k = g(x_k) + v_k$$

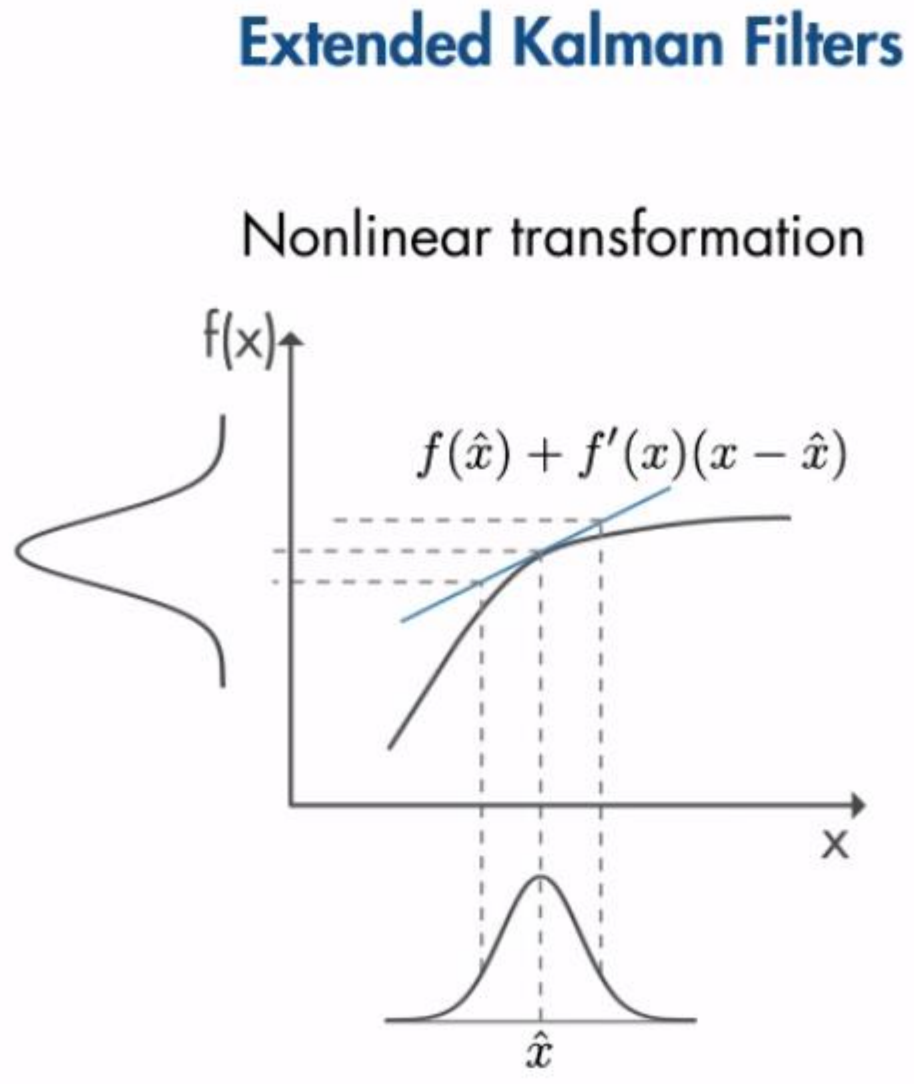
Nonlinear functions

Kalman Filter Algorithm for Nonlinear System

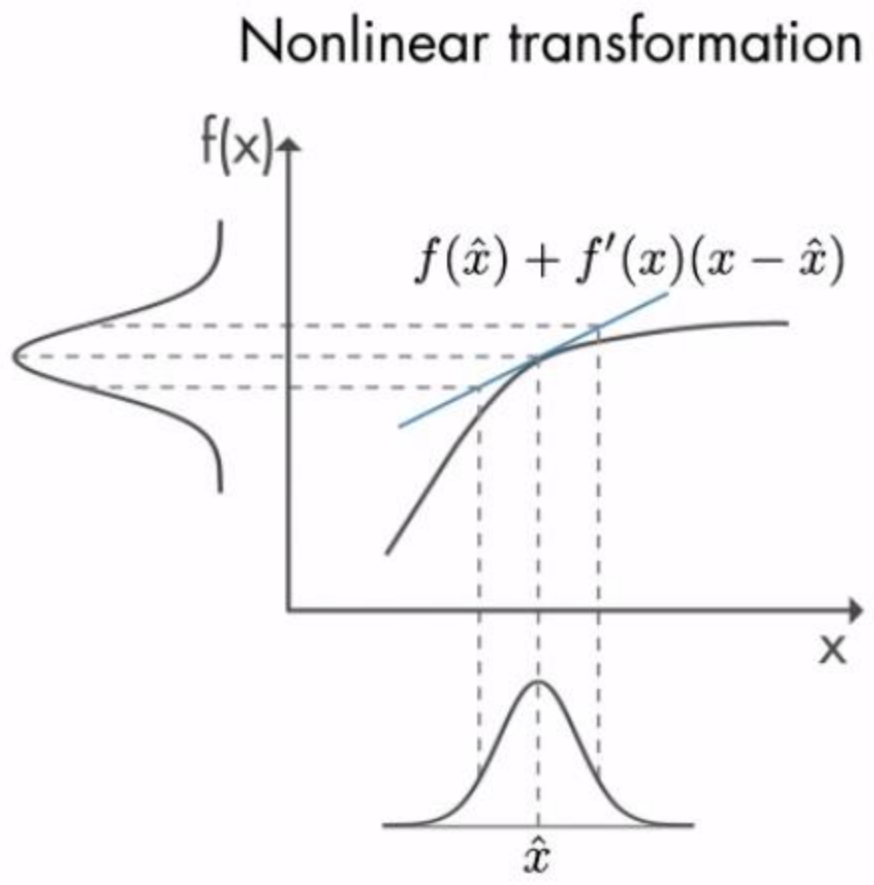


Kalman filter algorithm may not converge in nonlinear function

Kalman Filter Algorithm for Nonlinear System



Extended Kalman Filters



System:

$$x_k = f(x_{k-1}, u_k) + w_k$$
$$y_k = g(x_k) + v_k$$

Jacobians:

$$F = \left.\frac{\partial f}{\partial x}\right|_{\hat{x}_{k-1}, u_k}$$
$$G = \left.\frac{\partial g}{\partial x}\right|_{\hat{x}_k}$$

Linearized system:

$$\Delta x_k \approx F \Delta x_{k-1} + w_k$$
$$\Delta y_k \approx G \Delta x_k + v_k$$

Drawbacks to Using Extended Kalman Filters (EKFs):

- It is difficult to calculate the Jacobians (if they need to be found analytically)
- There is a high computational cost (if the Jacobians can be found numerically)
- EKF only works on systems that have a differentiable model
- EKF is not optimal if the system is highly nonlinear

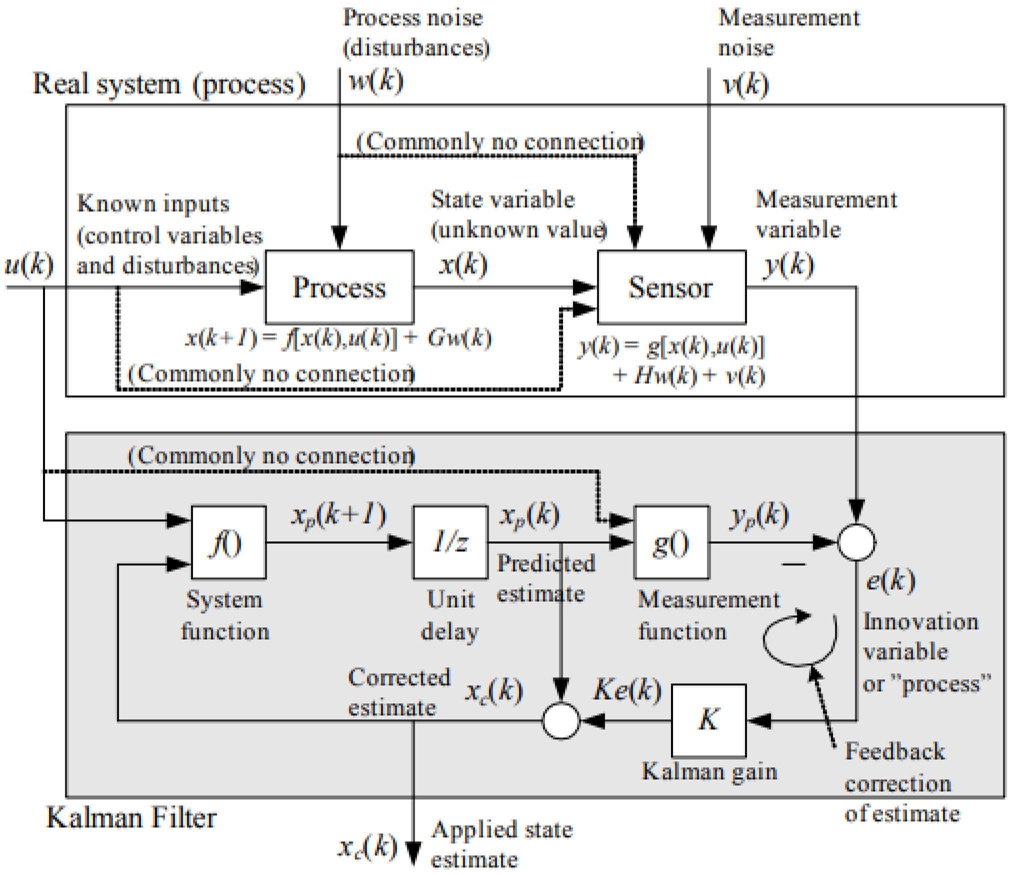


Figure 8.1: The Kalman Filter algorithm (8.35) – (8.38) represented by a block diagram

K = Kalman Gain (capital K)

k = time step (sometime t is used)

U(k) = control input at step k.

X(k) is the state at time step k.

8:49 PM

Kalman filter consists of two stage process

1. Prediction stage
2. Update stage (also known as correction stage)

Kalman filter is a recusive least square estimator + a motion model (or state matrix or Process model)

Designing a Kalman Filter

NUS
National University of Singapore

INITIAL STATE →

$$X_0, P_0$$

X represents the State Matrix
P is the Process Covariance Matrix

PREVIOUS STATE →

$$X_{t-1}, P_{t-1}$$

t is the iteration index
or time step

NEW PREDICTED STATE

small p represents the matrix has been updated with a new prediction

$$X_t^p = AX_{t-1} + Bu_t + w_t$$

$$P_t^p = AP_{t-1}A^T + Q_t$$

Prediction stage

↓

CALCULATE KALMAN GAIN (K) AND MEASURED DATA (Y)

$$K = \frac{P_t^p H^T}{HP_t^p H^T + R}$$

H matrix helps transform the matrix format of P into the format desired for the K matrix

$$Y_t = CY_t^m + Z^m$$

Y is a matrix containing measurement data (m)
C is a matrix transform to allow it be summed with Z
Z is the error term of the measurement

↓

UPDATE PROCESS AND STATE MATRIX

$$P_t = (I - KH)P_t^p$$

$$X_t = X_t^p + K[Y - HX_t^p]$$

Update/correction stage

← **CURRENT STATE BECOMES PREVIOUS STATE**

$$X_t = X_{t-1}$$

$$P_t = P_{t-1}$$

↑ REPEAT

How to tune the Kalman filter?

Q – Process noise (process auto covariance)
R - Measurement noise (or measurement noise auto-covariance).

However, adjusting Q is the more importance parameter than adjusting R (as R can be calculated). How do you adjust Q? The larger Q the larger Kalman Gain K and the stronger updating of the estimates.

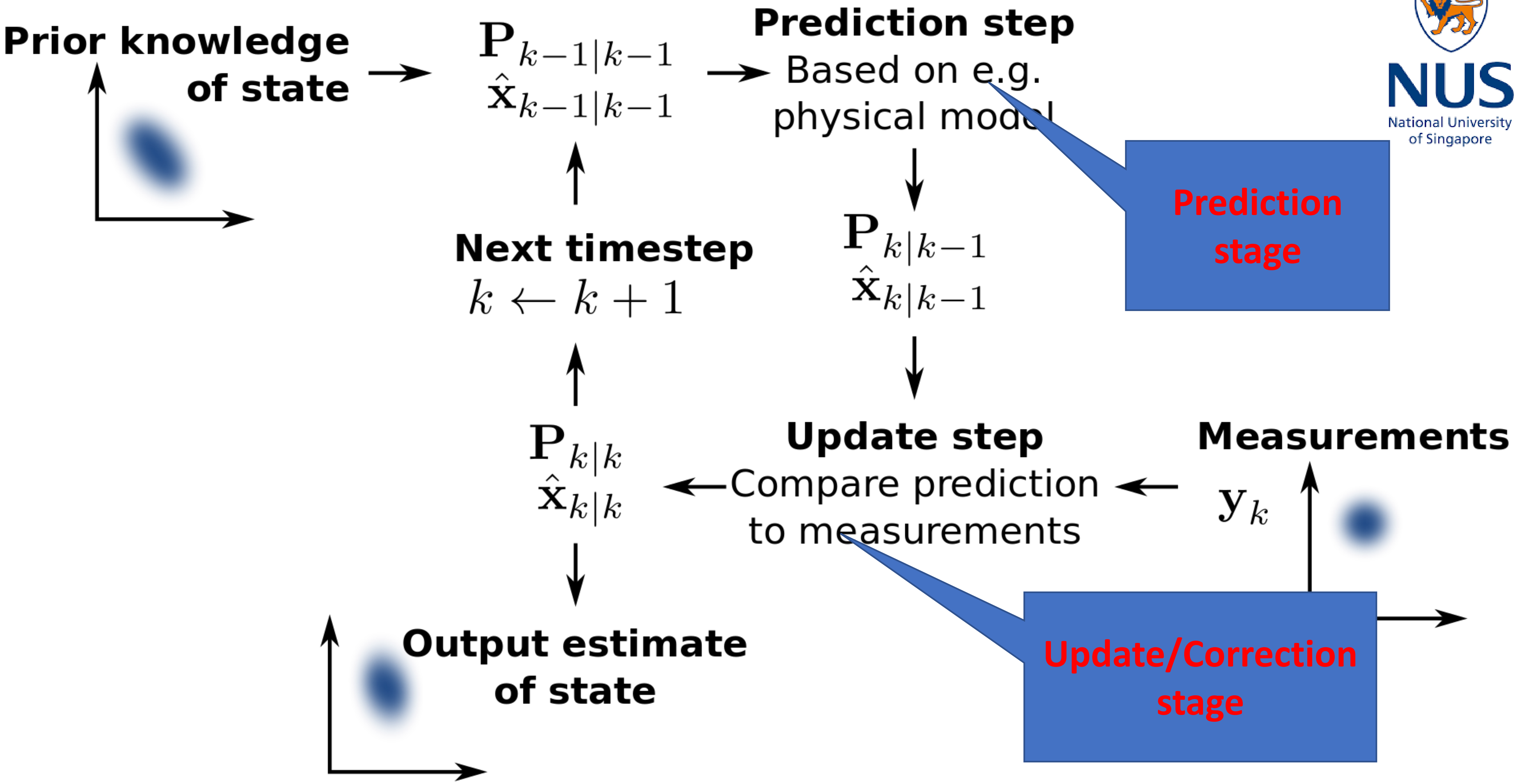
If you do not have any idea about numerical values, you can start by setting all the diagonal elements to one in the [matrix] and the rest of the [matrix] is 0.

Hence Q is Q = Q_o [Matrix] where

Q_o is the only tuning parameter.

And can start of trying Q_o = 0.01.

The time step t is replaced by k in this diagram

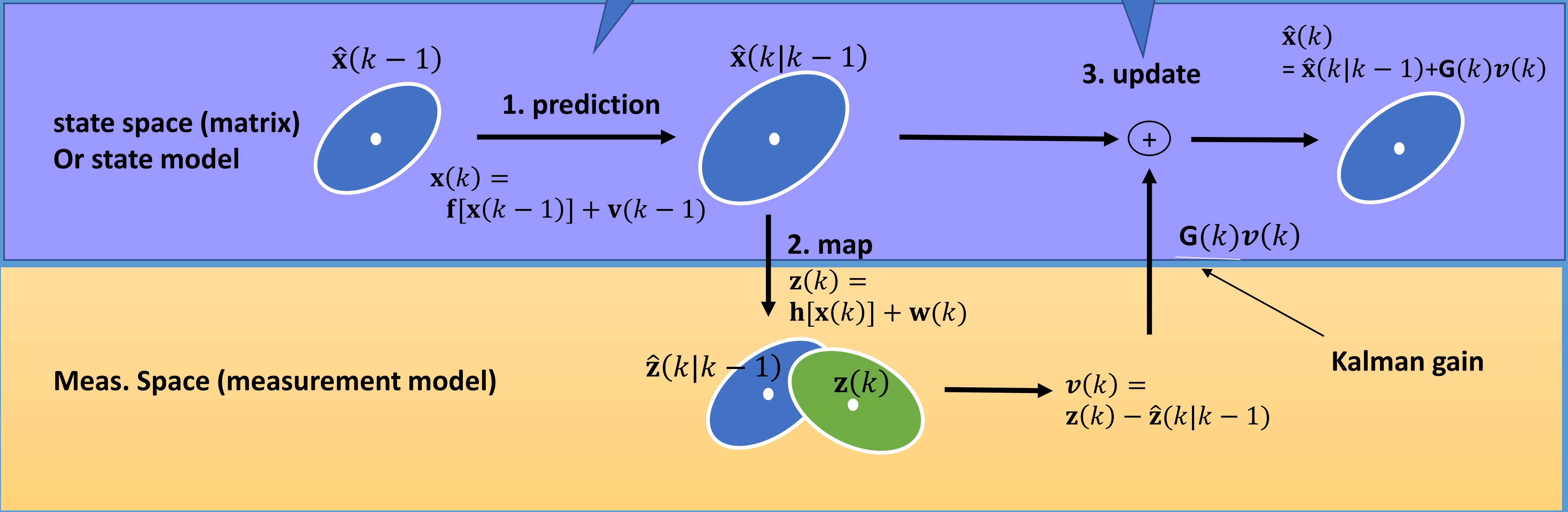


Kalman Filter (KF)

Prediction stage

Update/Correction stage

Given $\hat{\mathbf{x}}(k-1), \mathrm{z}(k)$ and their errors, to estimate $\hat{\mathbf{x}}(k)$ and its error



Thank you for your attention

Please refer to the following link for more examples and explanation:

1. https://www.coursera.org/lecture/state-estimation-localization-self-driving-cars/lesson-1-the-linear-kalman-filter-7DFmY

2. https://www.mathworks.com/videos/understanding-kalman-filters-part-1-why-use-kalman-filters--1485813028675.html

3. https://www.youtube.com/watch?v=s_9InuQAx-g