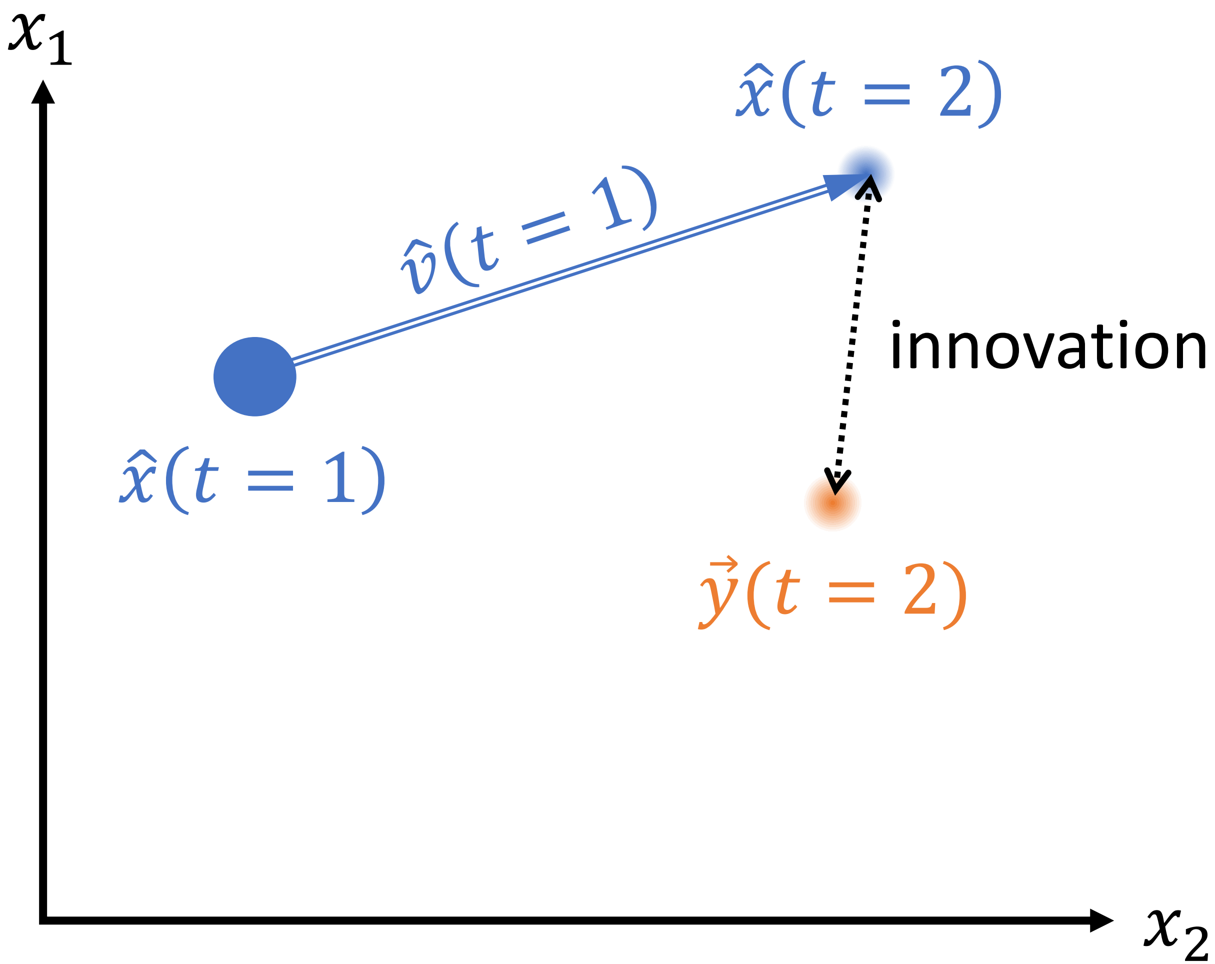
Kalman-Filter

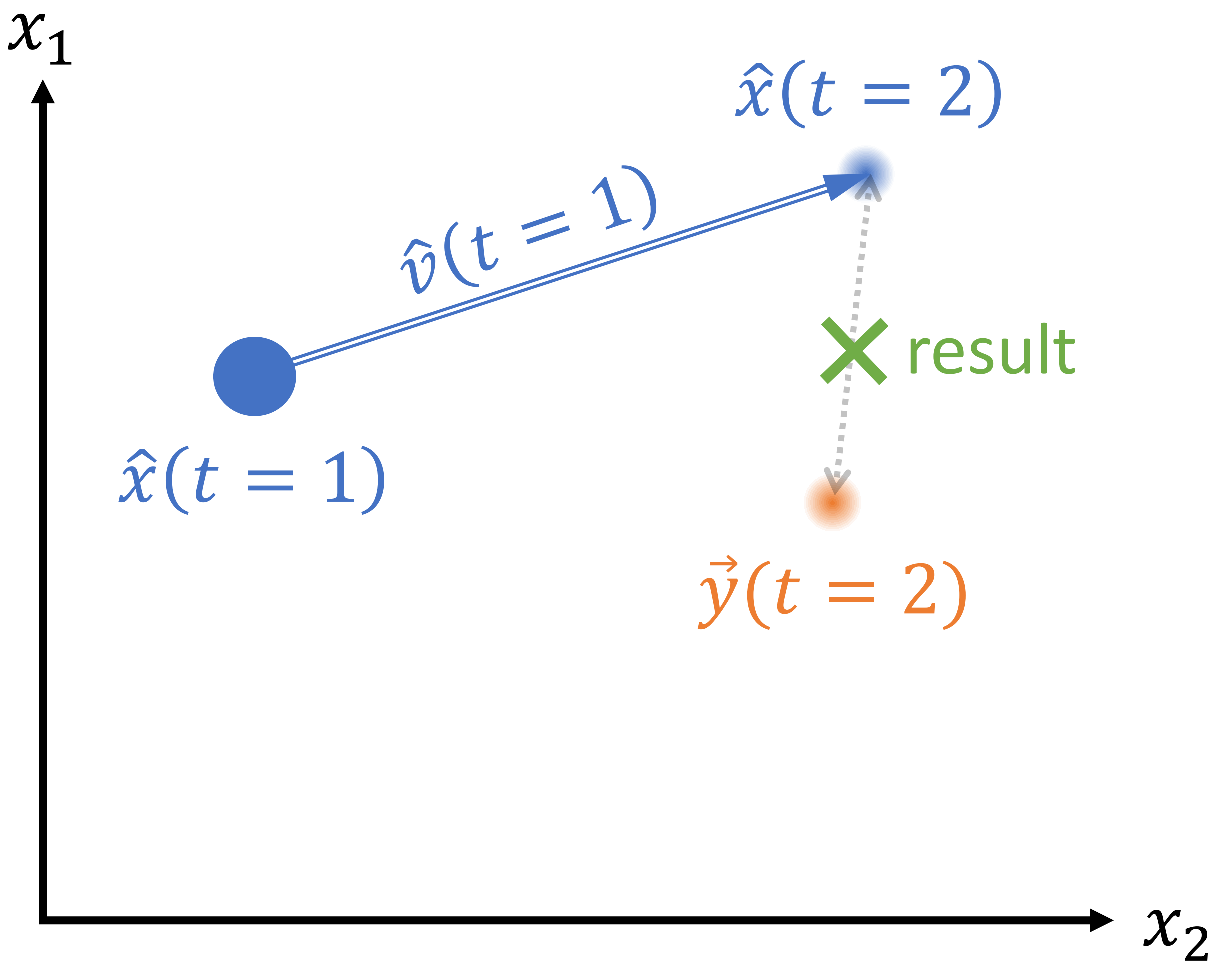
Overview

1. introduction
2. g-h-filter
3. the hidden markov model
4. kalman-filter

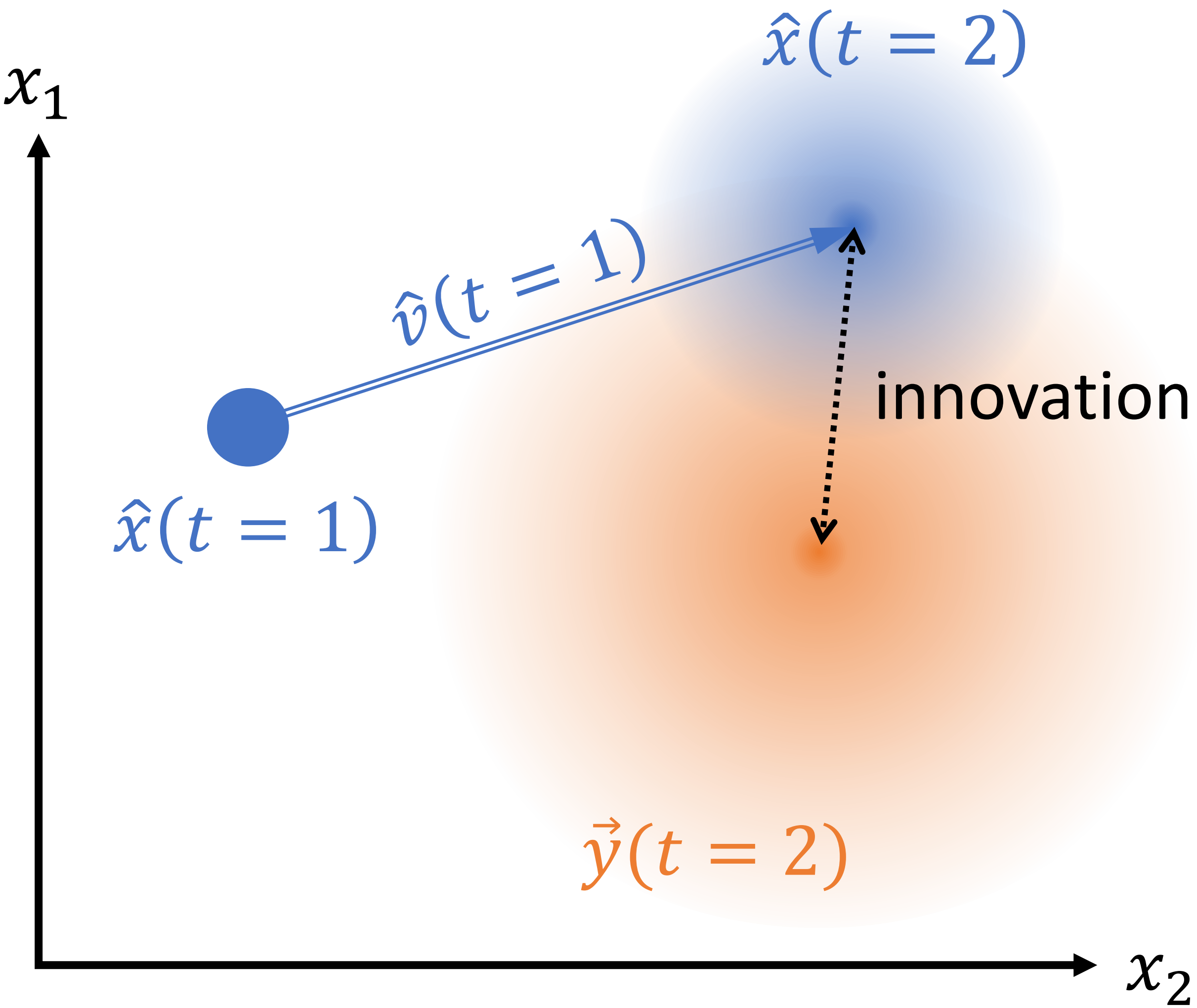
Introduction



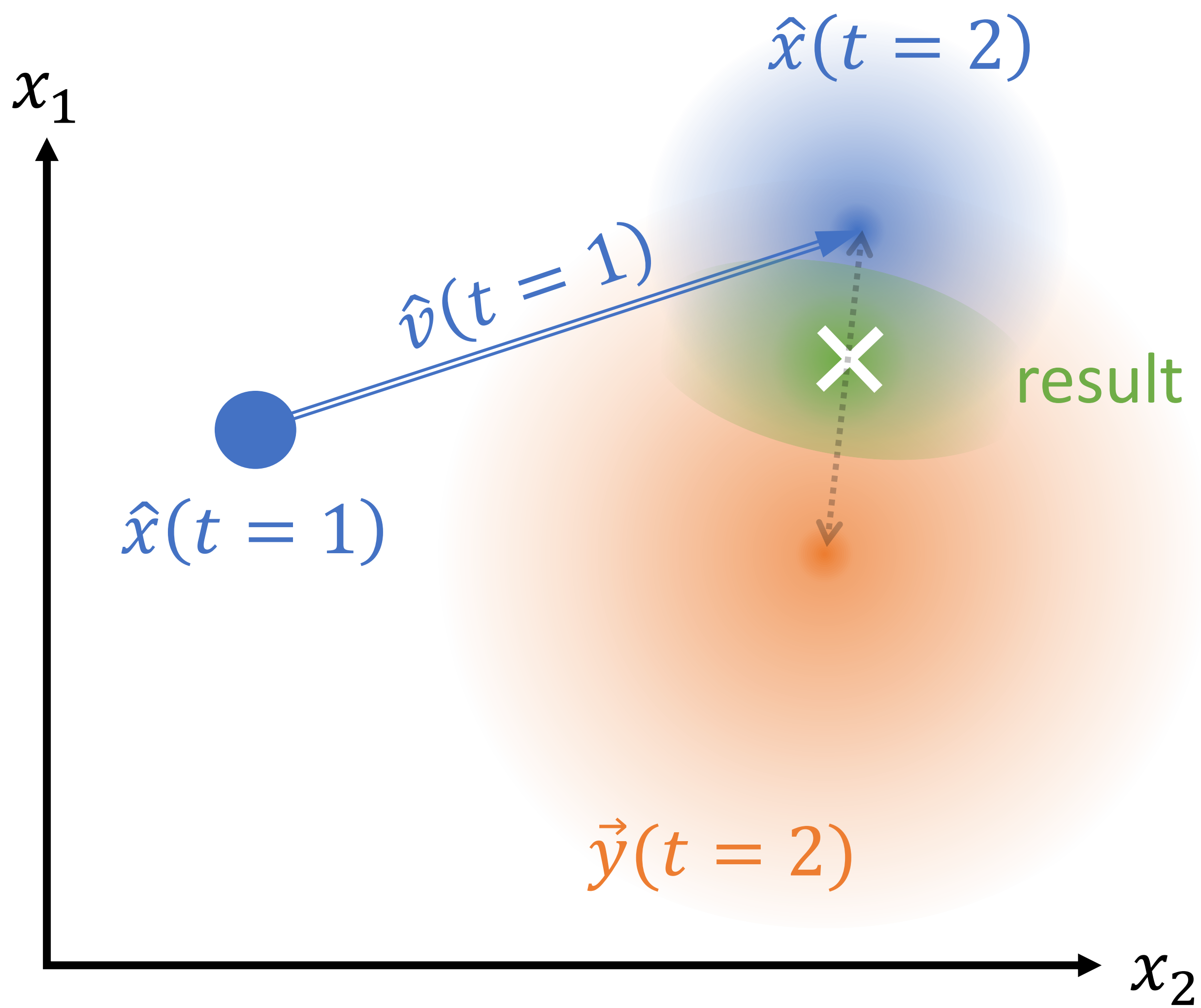
Introduction



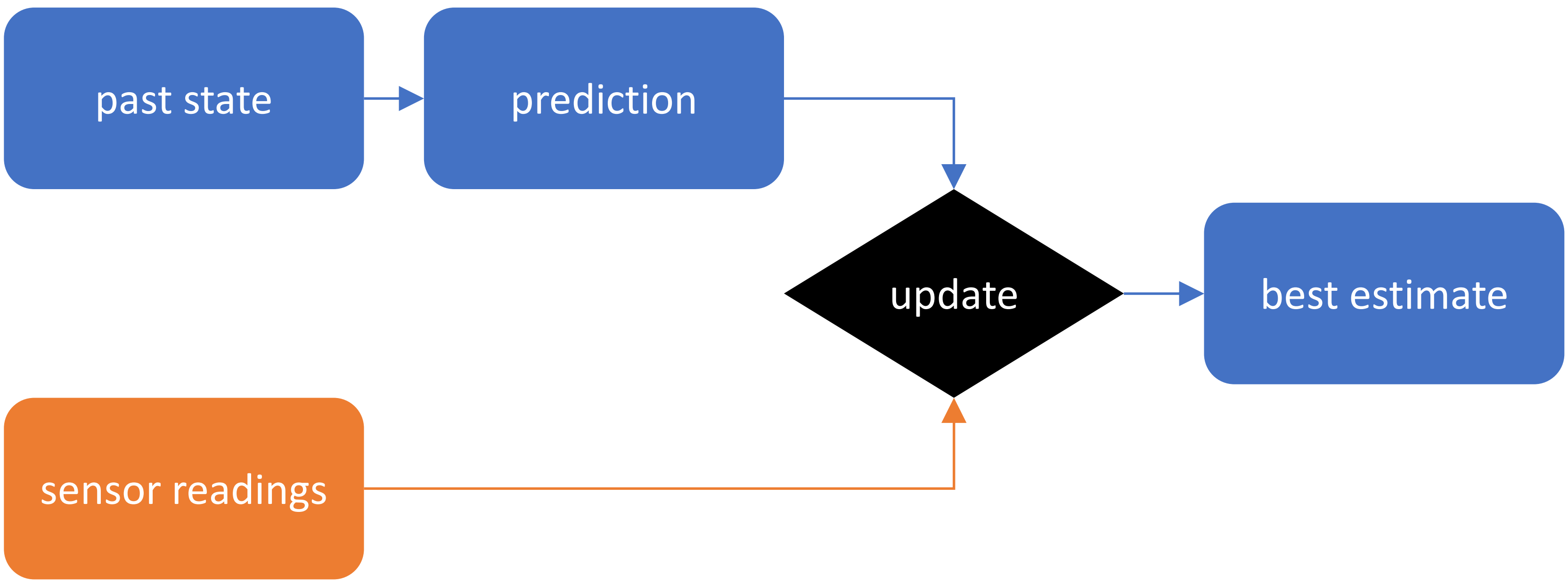
Introduction



Introduction



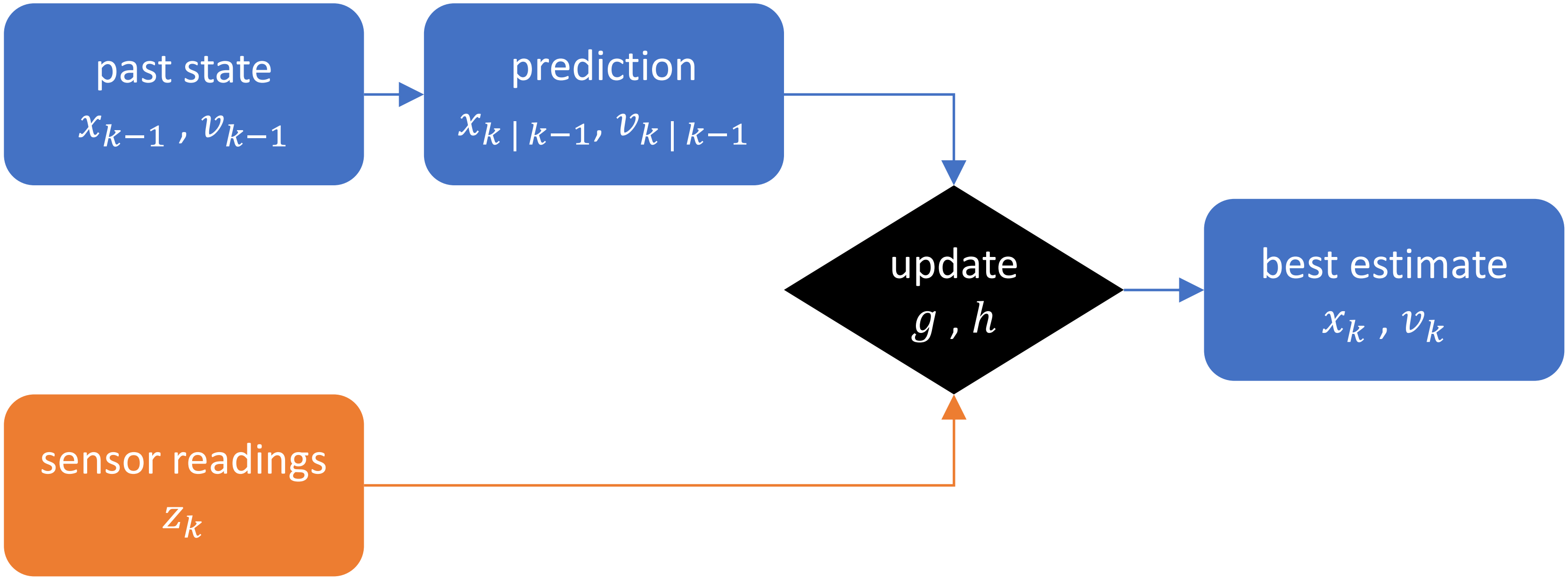
G-H-Filter


past state
prediction
update
best estimate
sensor readings

G-H-Filter

state : (x_{k-1}, v_{k-1}).

input : measurement z_k after time Δt



G-H-Filter

state : (x_{k-1}, v_{k-1}).

input : measurement z_k after time Δt

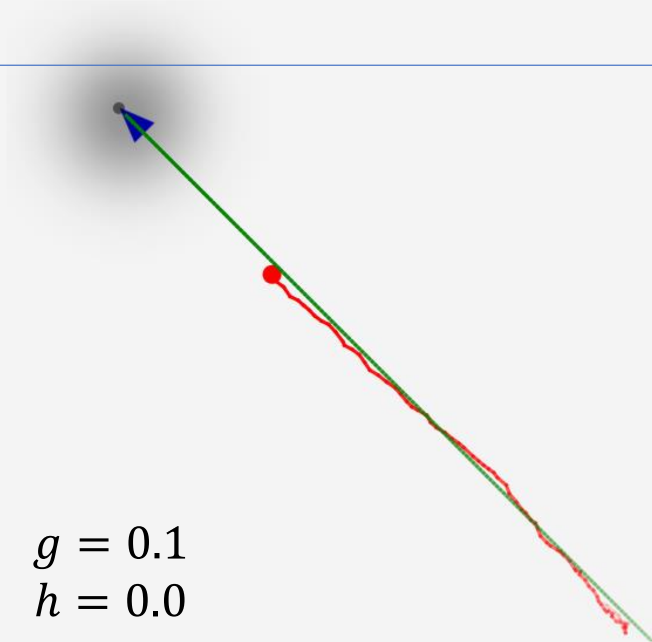
step 1: **prediction**

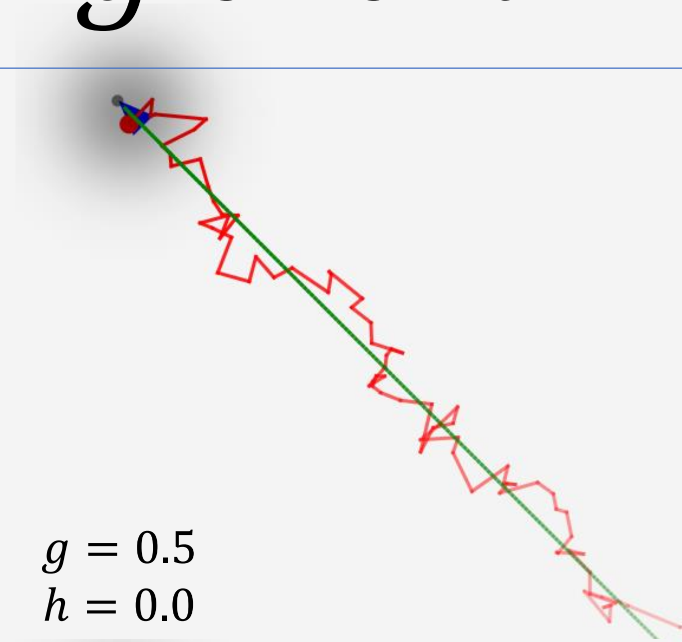
- $x_{k\,|\,k-1} = x_{k-1} + v_{k-1} \cdot \Delta t$
- $v_{k\,|\,k-1} = v_{k-1}$ (velocity assumed constant)

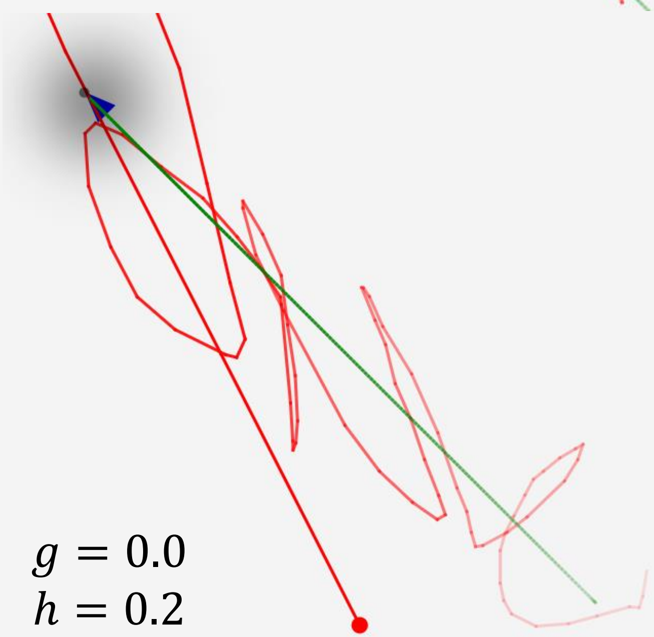
step 2: **update**

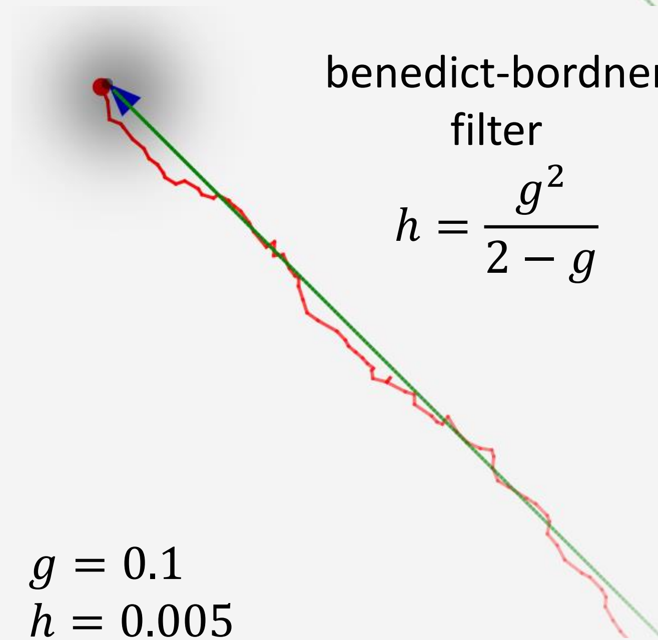
- $\tilde{y}_k = \left(z_k - x_{k\,|\,k-1}\right)$: <u>innovation</u> / pre-fit <u>residual</u>
- $x_k = x_{k\,|\,k-1} + \boldsymbol{g} \cdot \tilde{y}_k$
- $v_k = v_{k\,|\,k-1} + \boldsymbol{h} \cdot \tilde{y}_k / \Delta t$

Choice of g and h





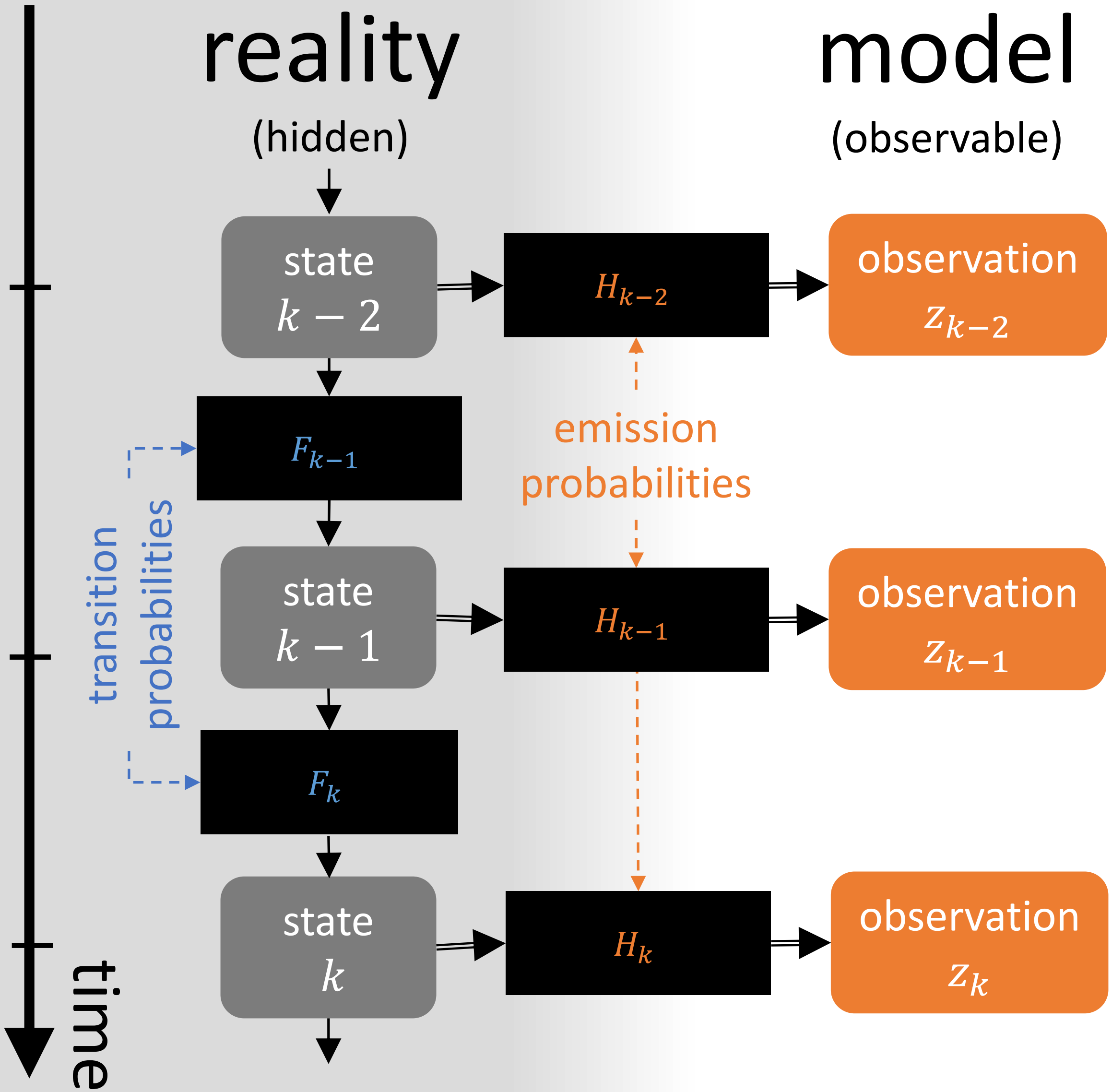




benedict-bordner filter

$$h = \frac{g^2}{2 - g}$$

Hidden Markov Model



N possible states

probabilities:

- $F_{ij} = p(x_k = j \mid x_{k-1} = i)$

(markov-matrix) $\in \mathbb{R}^{N\times N}$

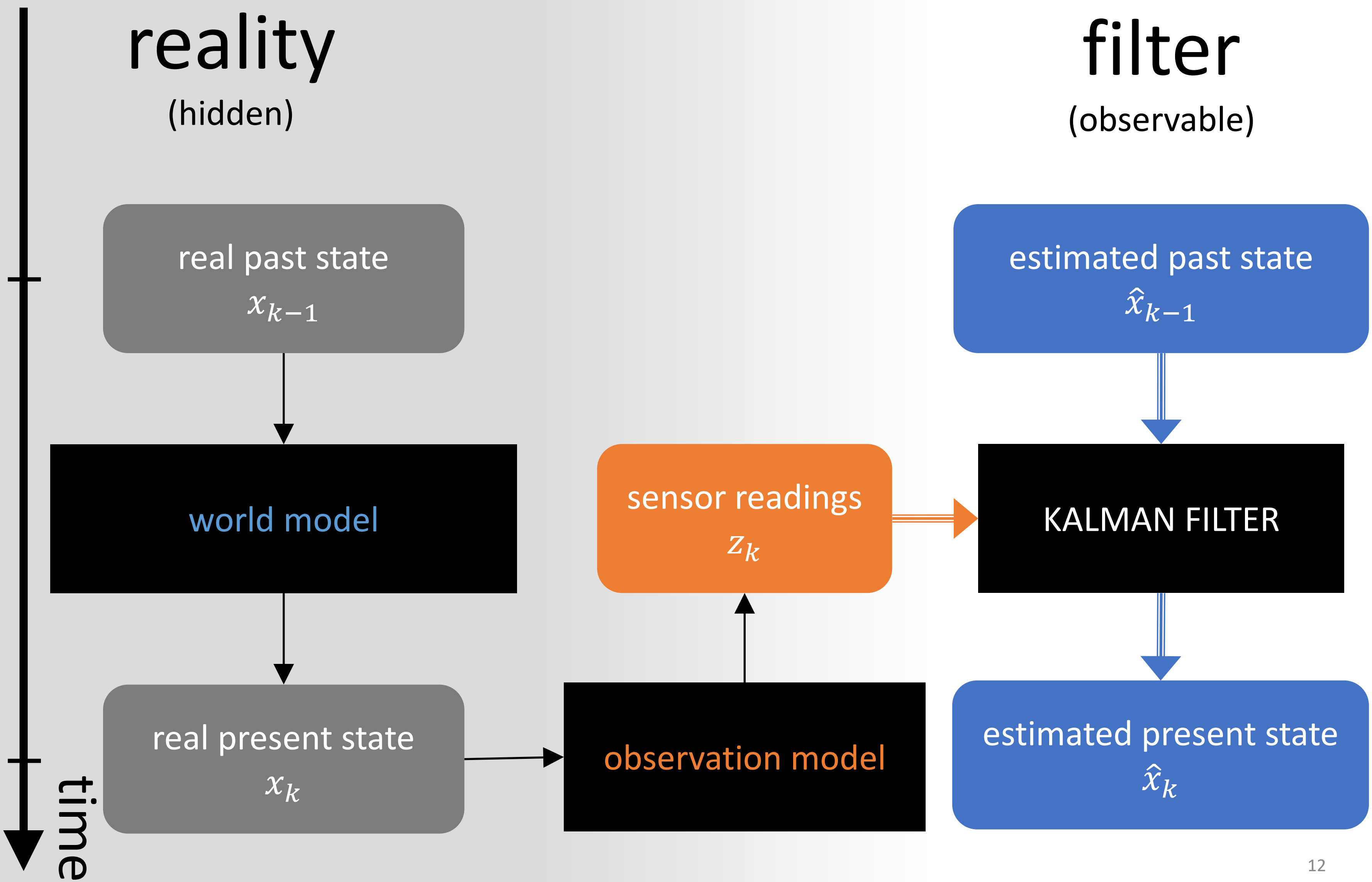
T possible observations

probabilities:

- $H_{i,j} = p(z_i | x = j)$

(markov-matrix) $\in \mathbb{R}^{T\times N}$

Kalman Filter Models



From Markov to Kalman

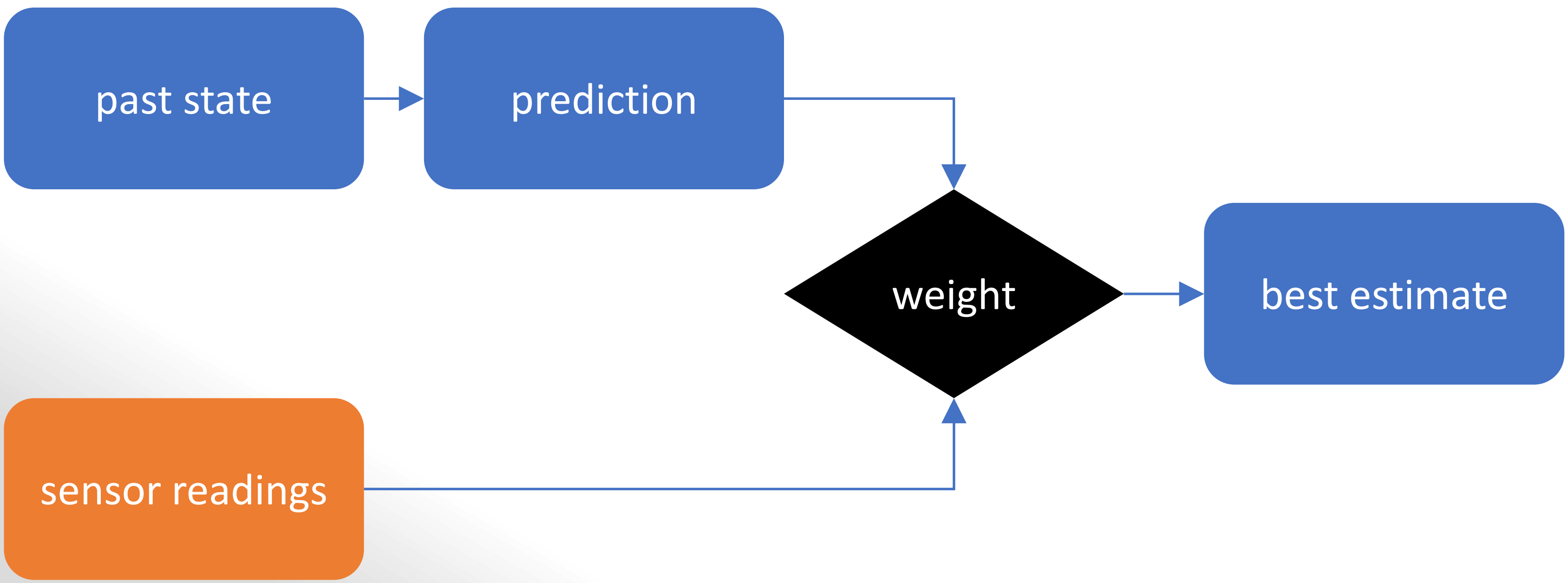
hidden markov model

- countable state-space S
  - often finite with $|S| = N$
  - one dimension
- finite observation-space
  - dimension 1, size T
- state is N-dimensional vector of probabilities

kalman-filter model

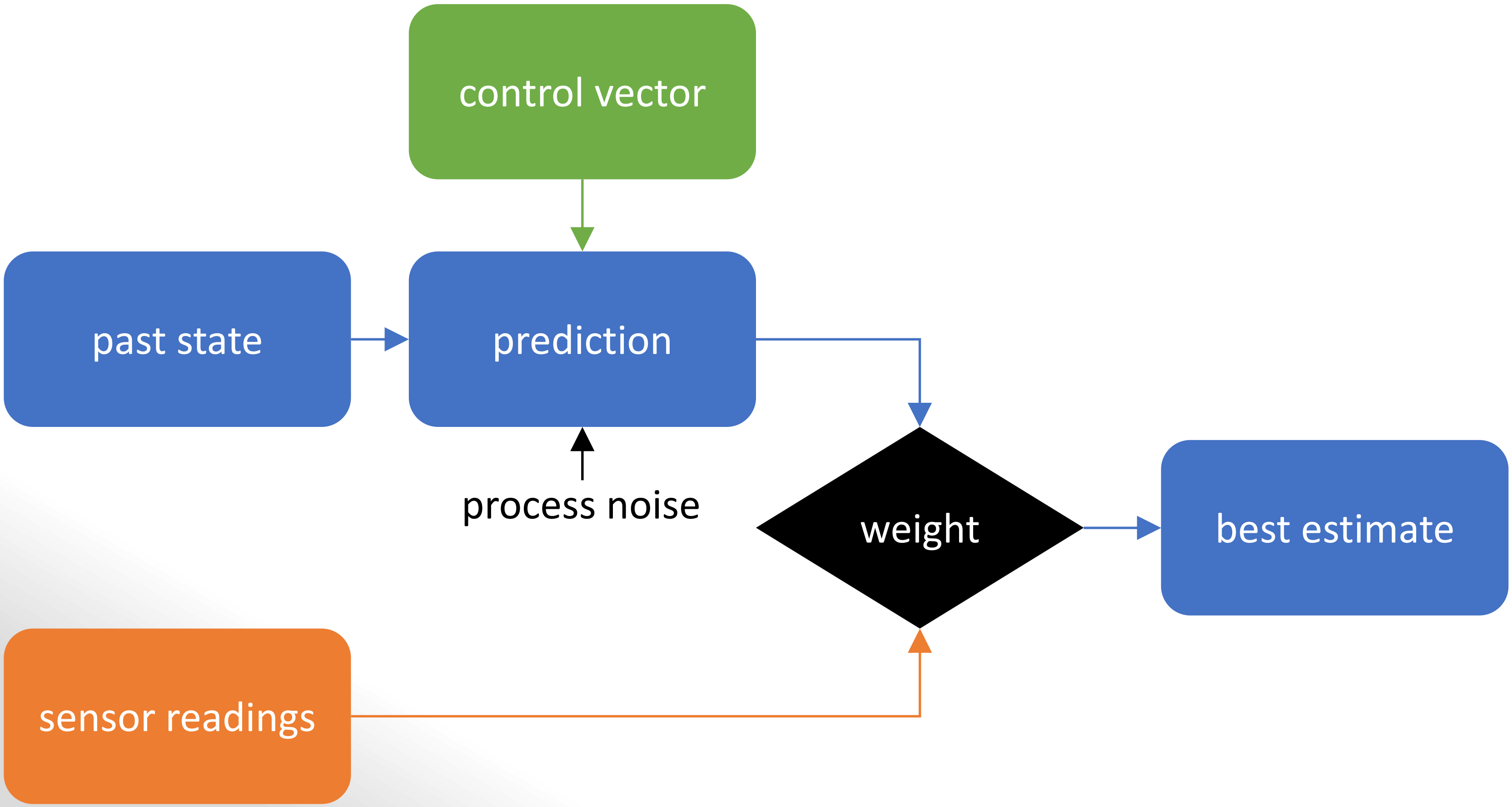
- continuous state-space
  - $S = \mathbb{R}^N$
  - N dimensions
- continuous obs.-space
  - dimension T
- state is N mean values and covariance matrix
- adds control
- adds process noise

From G-H to Kalman



reality

From G-H to Kalman



reality

Kalman Filter: World Model

the kalman-filter assumes the following state progression (based solely on hidden state x_{k-1}):

$$x_k = \underbrace{F_k x_{k-1}}_{\text{transition}} + \underbrace{B_k u_k}_{\text{control}} + \underbrace{w_k}_{\text{noise}}$$

- F_k : state-transition model (e.g. classical mechanics)
- B_k : control-input model (e.g. motor affects position)
- u_k : control vector (e.g. how much the motor is driven)
- w_k : process noise – with covariance Q_k

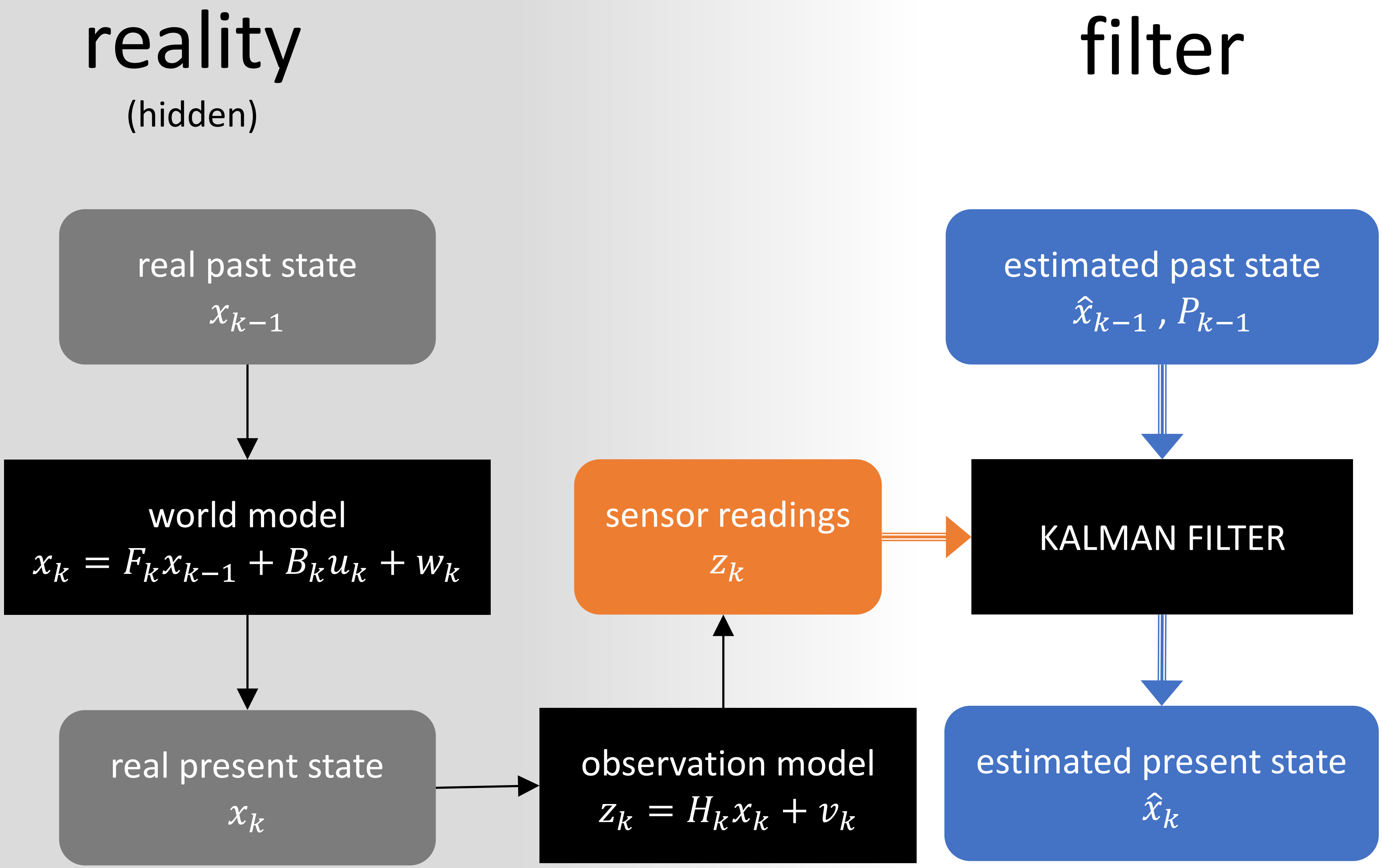
Kalman Filter: Observation Model

an observation (measurement) z_k of the hidden true state x_k is modeled as

$$z_k = \underbrace{H_k x_k}_{\text{observation}} + \underbrace{v_k}_{\text{noise}}$$

- H_k : observation model (state-space -> observe-space)
- v_k : observation noise – with covariance R_k

Kalman Filter Models



Kalman Step 0 : Past state

reality

real past state
x_{k-1}

Mean

$\hat{x}_{k\ -1}$

Covariance

$P_{k\ -1}$

estimated past state
$\hat{x}_{k-1}\ ,\ P_{k-1}$

Kalman Step 1 : Prediction

reality

real past state

x_{k-1}

world model

$x_k = F_k x_{k-1} + B_k u_k + w_k$

real present state

x_k

Mean

$$\hat{x}_{k\,|\,k-1} = F_k \hat{x}_{k-1} + B_k u_k$$

Covariance

$$P_{k\,|\,k-1} = F_k P_{k-1} F_k^T + \mathrm{Q}_k$$

Kalman Step 2 : Update

reality

sensor readings
z_k

observation model
$z_k = H_k x_k + v_k$

Innovation Mean

$$\tilde{y}_k = z_k - H_k \hat{x}_{k \mid k-1}$$

Innovation Covariance

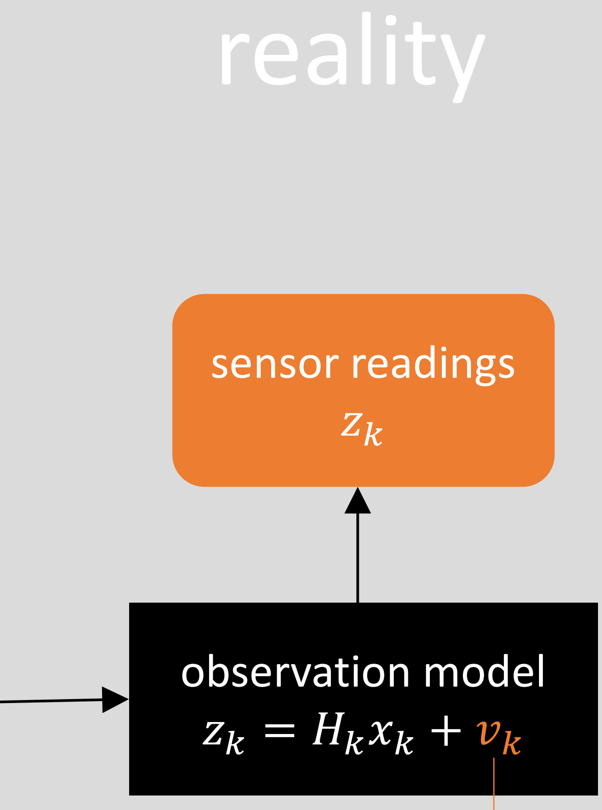
$$S_k = R_k - H_k P_{k \mid k-1} H_k^T$$

inverse covariance
aka **precision**

Kalman Gain

$$K_k = P_{k \mid k-1} H_k^T S_k^{-1}$$

Kalman Step 2 : Update



Mean (used as estimate)

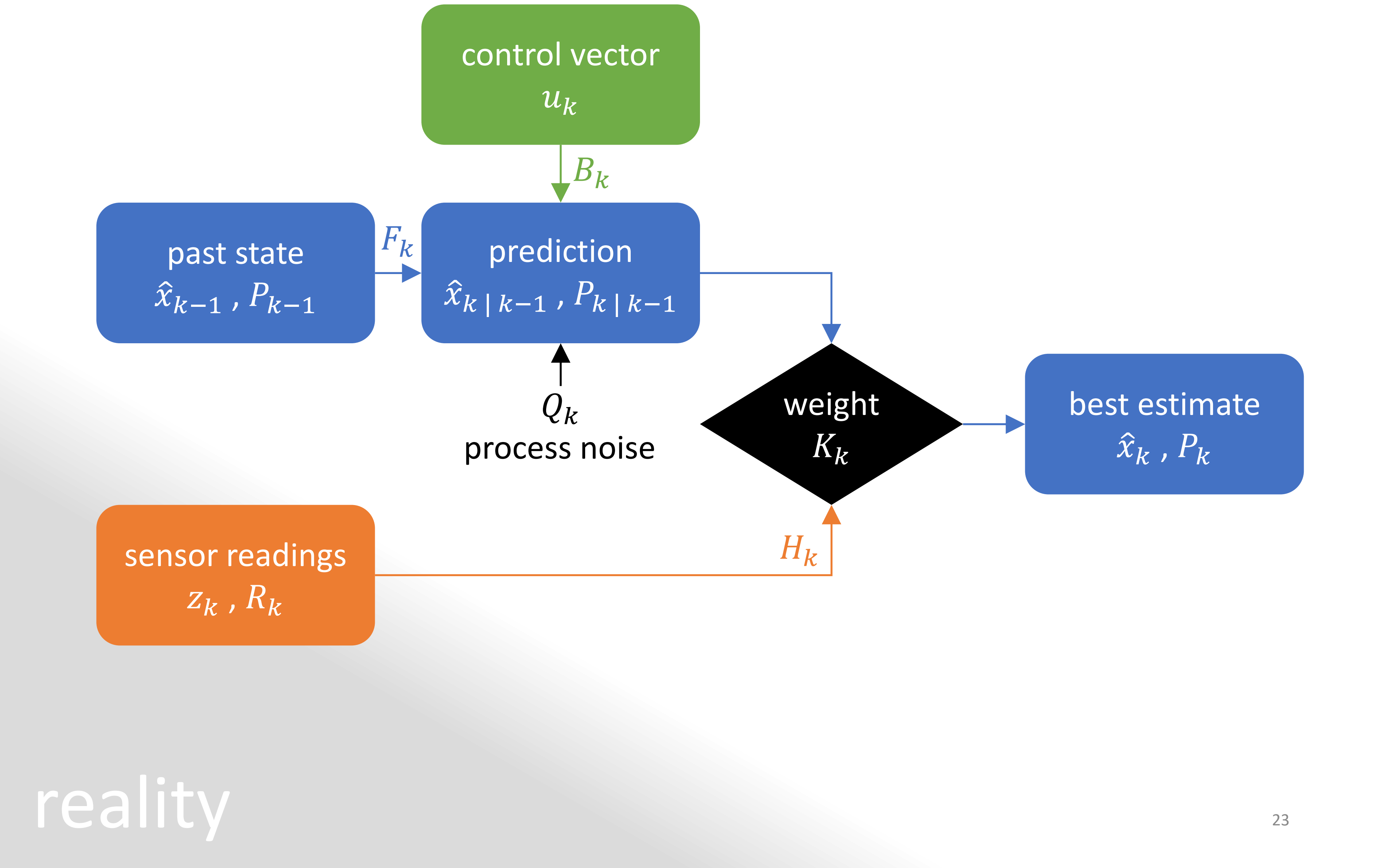
$$\hat{x}_k = \hat{x}_{k\,|\,k-1} + K_k \tilde{y}_k$$

Covariance

$$P_k = (I - K_k H_k) P_{k\,|\,k-1} (I - K_k H_k)^T \\ + K_k R_k K_k^T$$

- innovation mean $\tilde{y}_k$
- kalman gain K_k

Kalman-Filter



Sources (online)

- https://en.wikipedia.org/wiki/Kalman_filter
- https://en.wikipedia.org/wiki/Hidden_Markov_model
- https://en.wikipedia.org/wiki/Alpha_beta_filter
- Tim Babb – How a Kalman-Filter works, in pictures (http://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/)
- Roger R. Labbe – Kalman and Bayesian Filters in Python (https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python)

All Links last visited on September 24th, 2018

Sources (books)

- Eli Brookner - Tracking and Kalman Filtering Made Easy (Wiley Interscience 2002)

Thank you!

Any Questions?