

Augmented Reality II

- Kalman Filters -

Gudrun Klinker

May 25, 2004

Outline

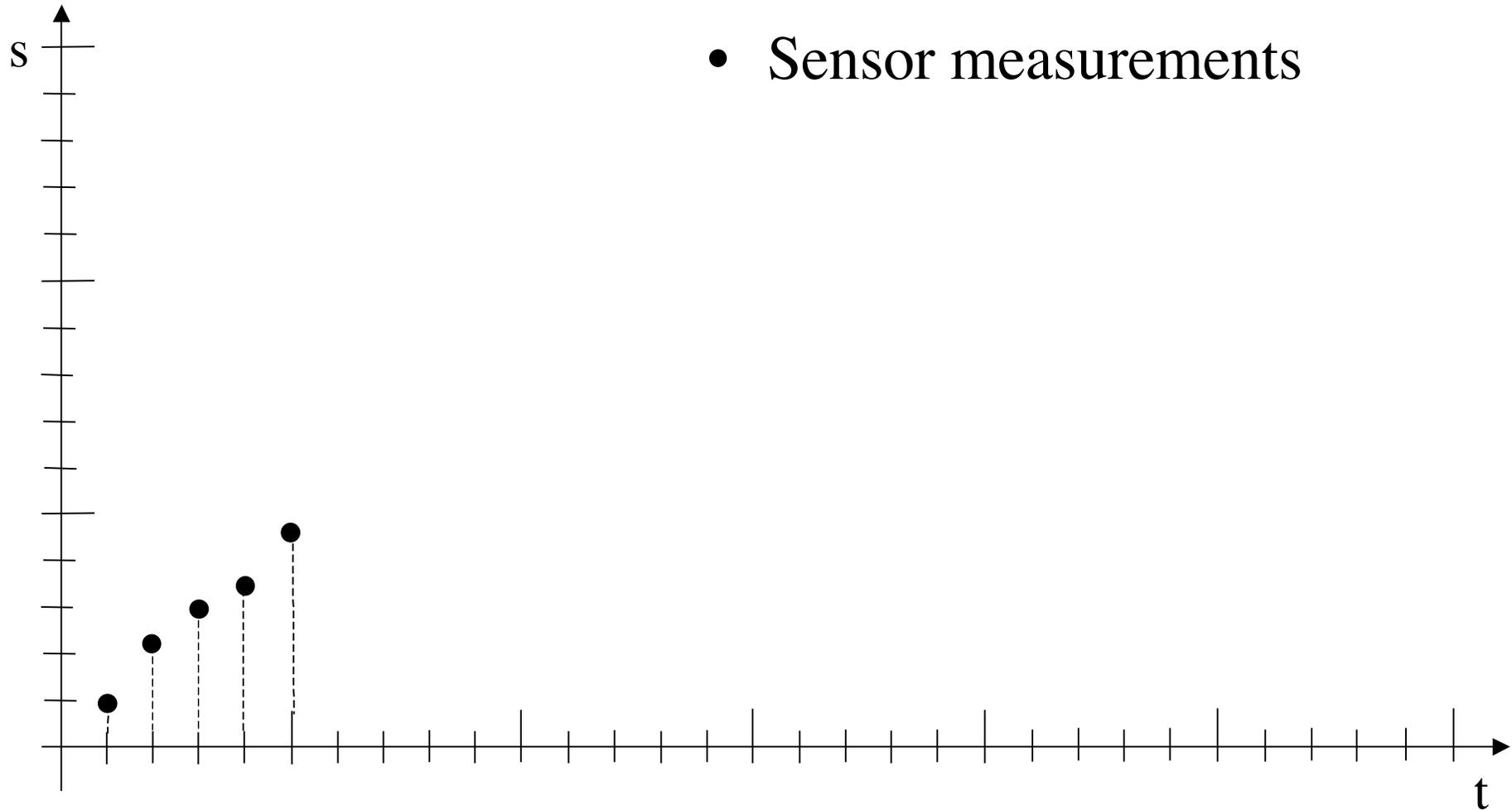
- Motivation
- Discrete Kalman Filter
 - Modeled Process
 - Computing Model Parameters
 - Algorithm
- Extended Kalman Filter
- Kalman Filter for Sensor Fusion

Literature

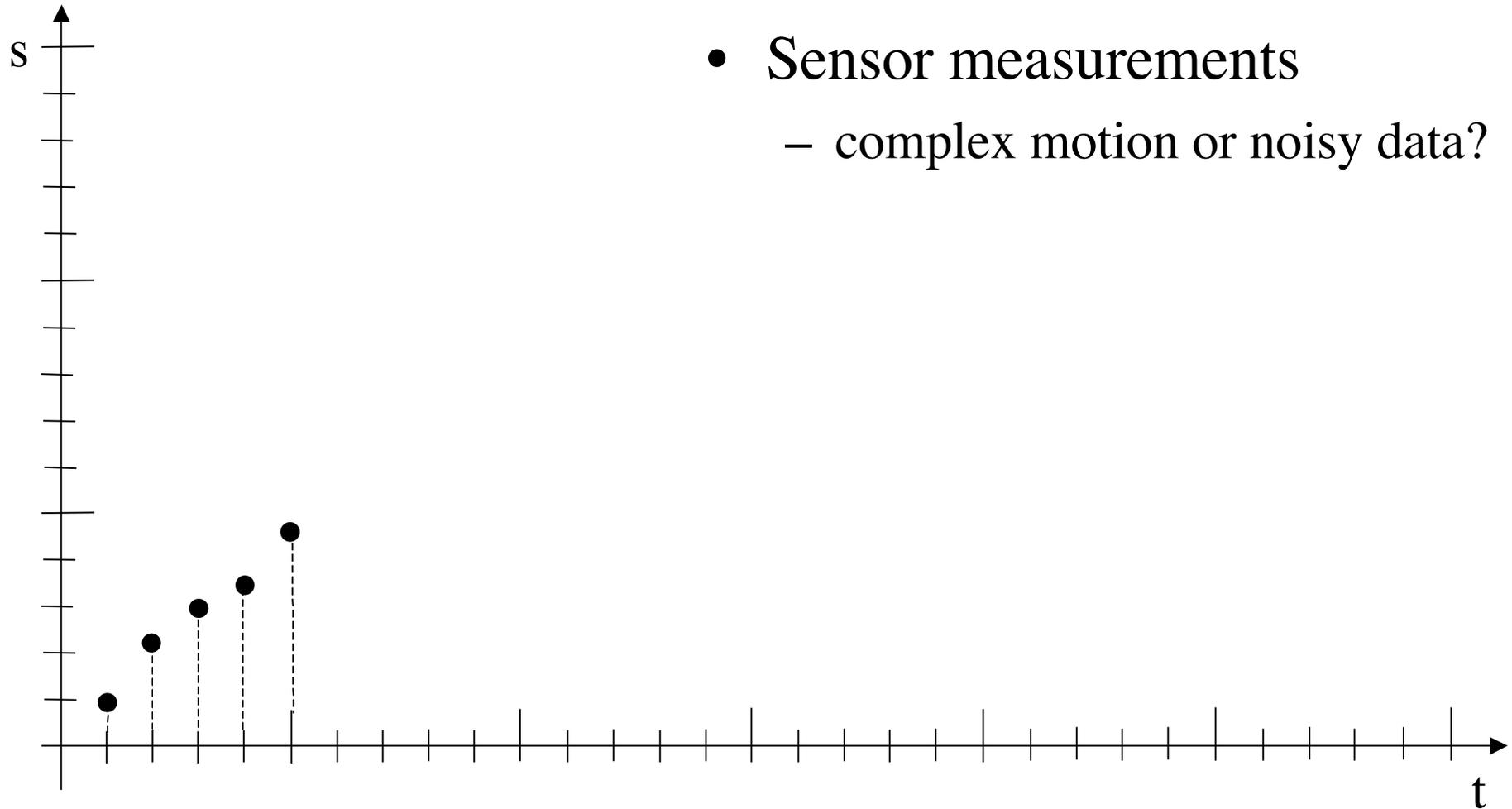
- G. Welch and G. Bishop, “An Introduction to the Kalman Filter”, SIGGRPAPH 2001 Course 8.
- <http://www.cs.unc.edu/~welch/kalman>
- A. Gelb (editor), “Applied Optimal Estimation”

Motivation

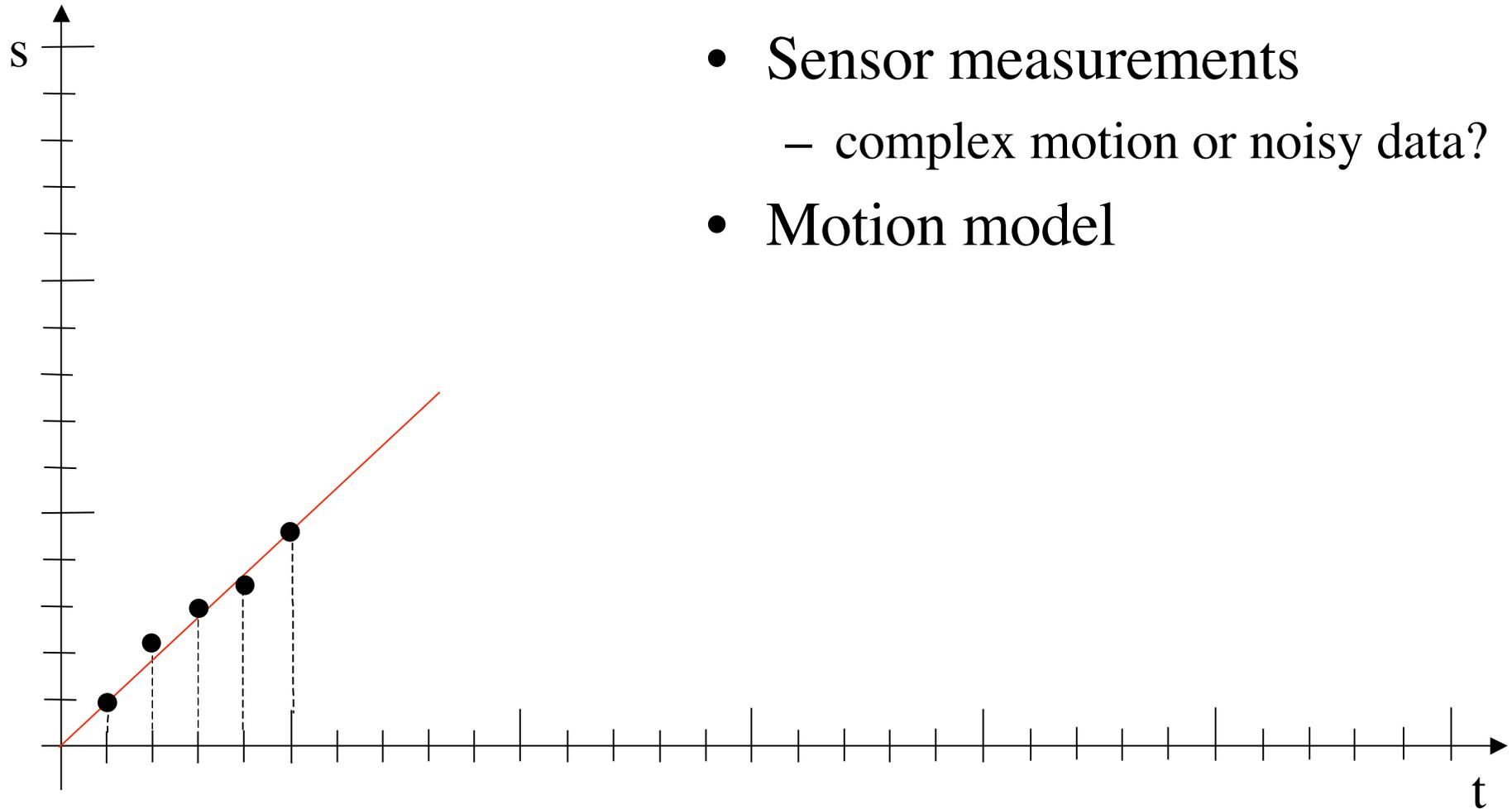
- Sensor measurements



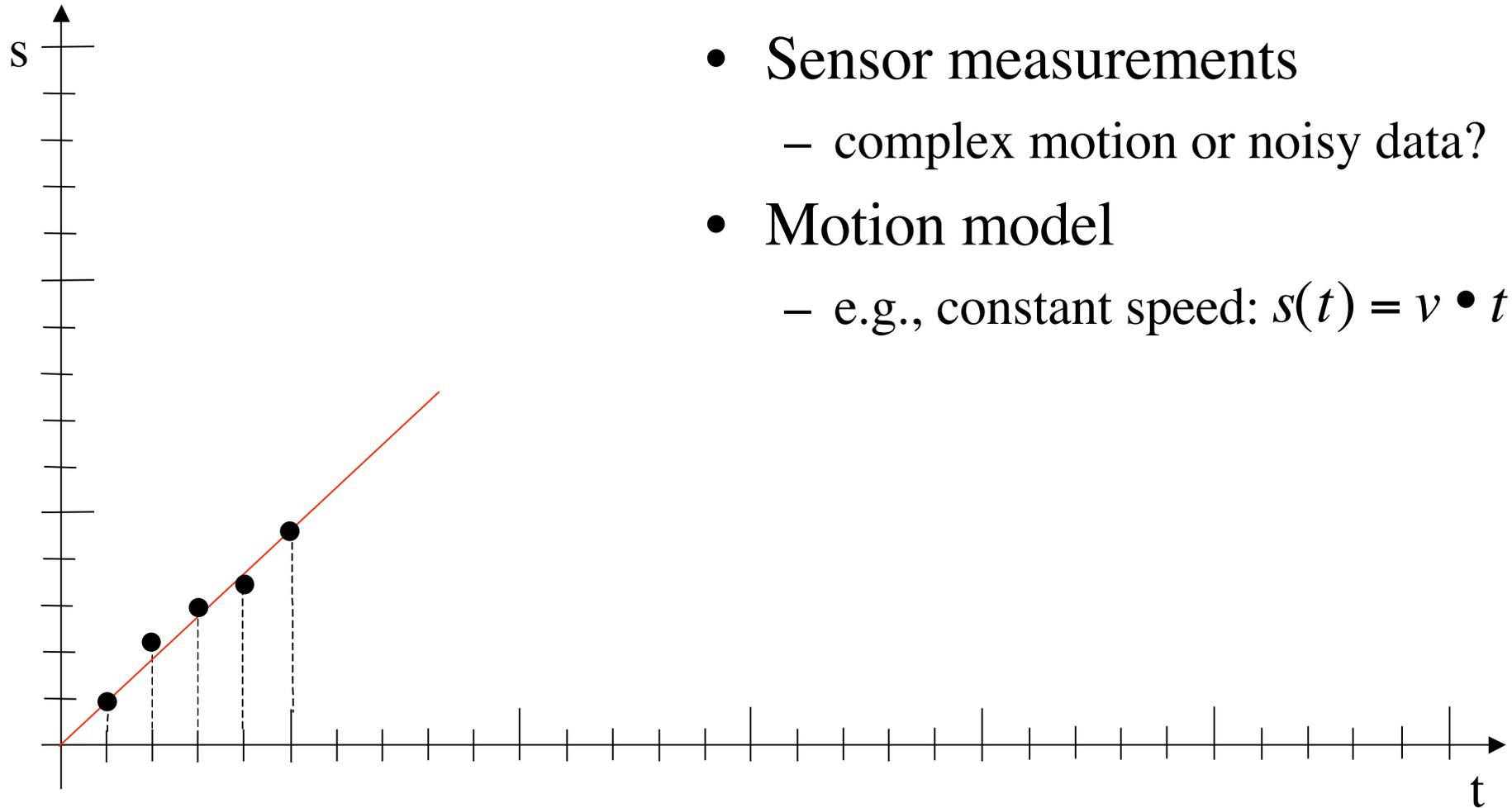
Motivation



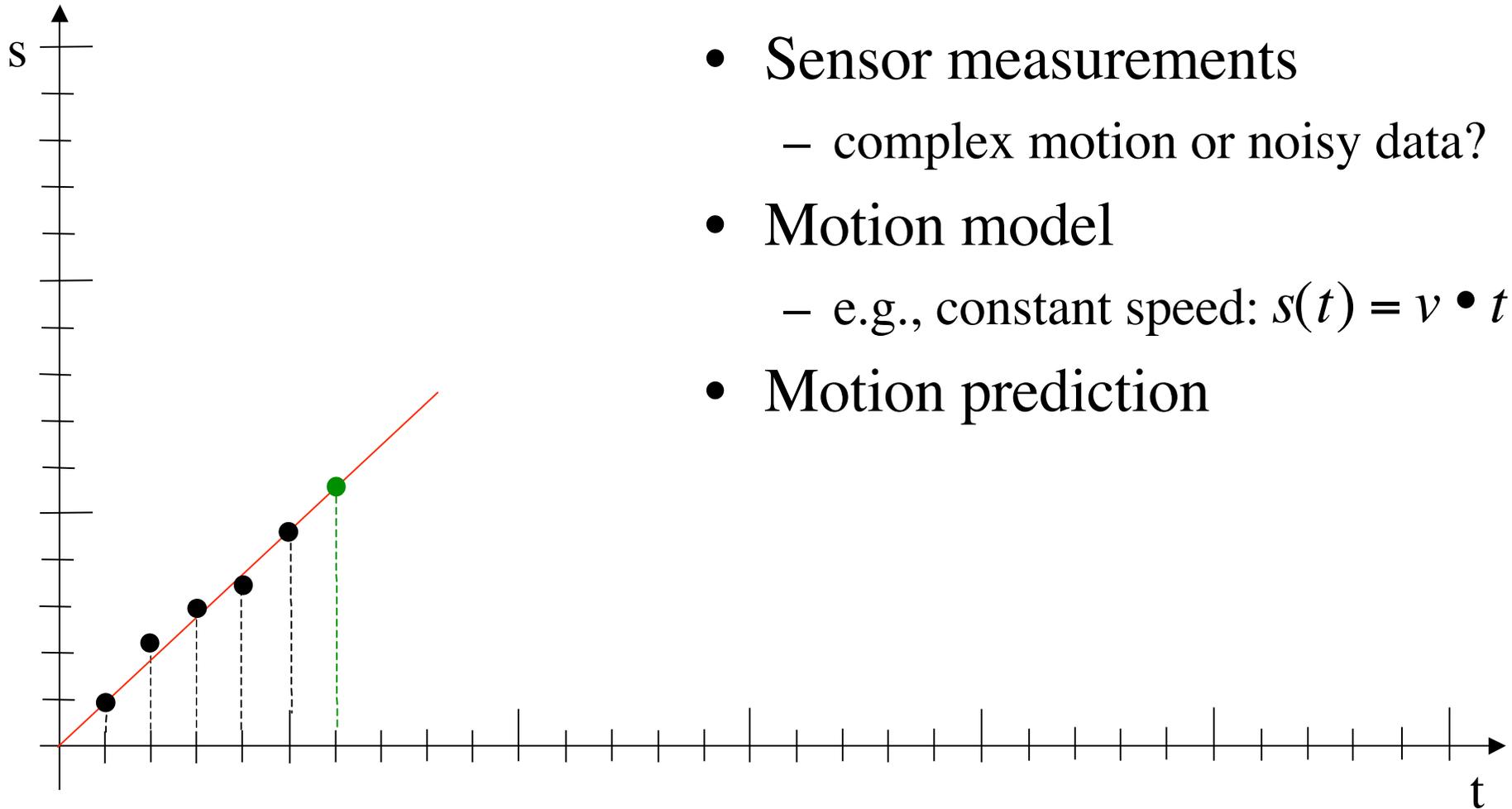
Motivation



Motivation

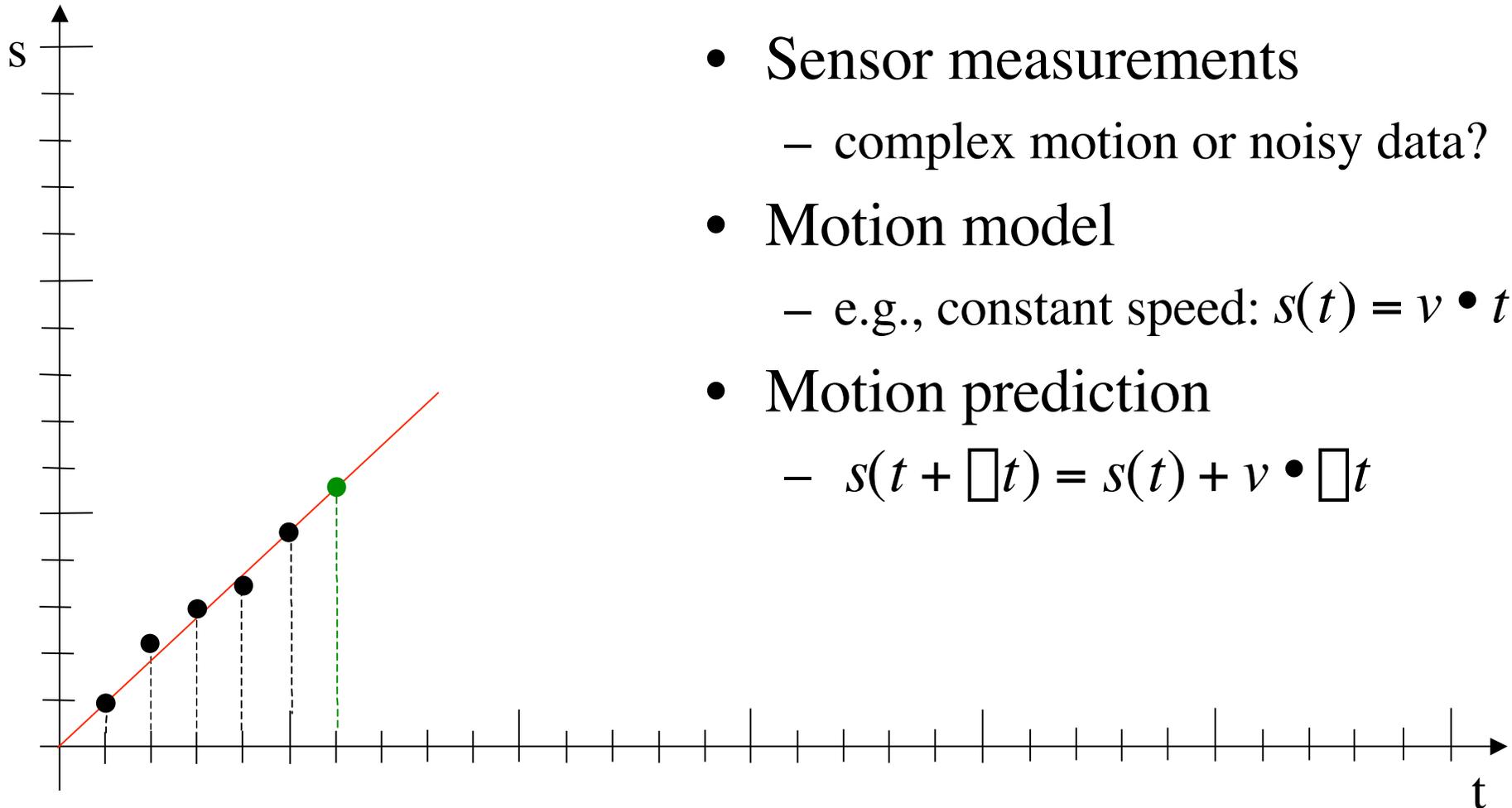


Motivation



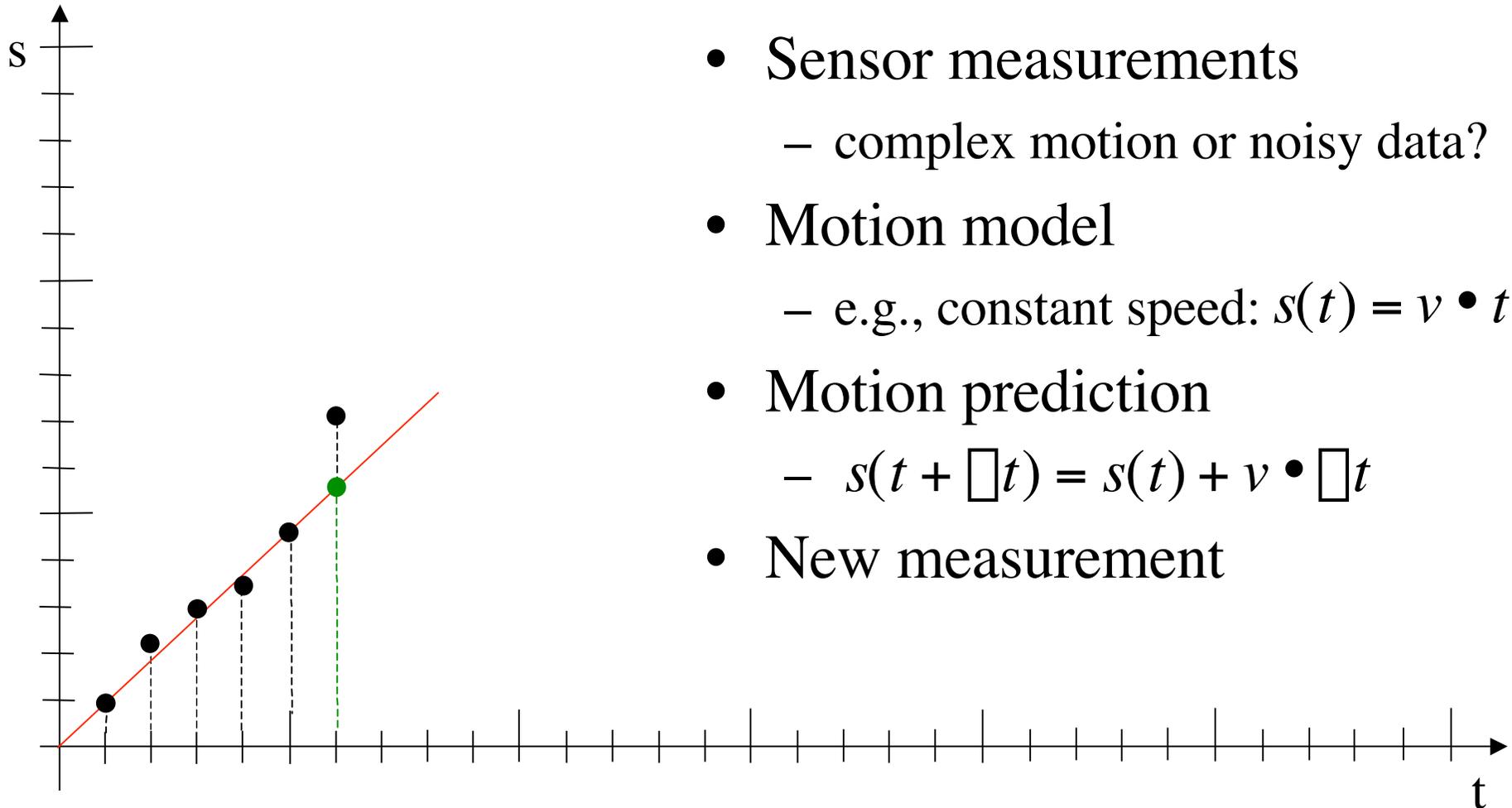
- Sensor measurements
 - complex motion or noisy data?
- Motion model
 - e.g., constant speed: $s(t) = v \cdot t$
- Motion prediction

Motivation



- Sensor measurements
 - complex motion or noisy data?
- Motion model
 - e.g., constant speed: $s(t) = v \cdot t$
- Motion prediction
 - $s(t + \Delta t) = s(t) + v \cdot \Delta t$

Motivation



Rudolf Emil Kalman

- Born in Budapest, 1930
- B.S., M.S from MIT
- Ph.D. from Columbia U.
- Professor at Stanford U. and U. Florida
- Many awards
- Seminal paper: “A new Approach to Linear Filtering and Prediction Problems”, Transactions ASME, 1960.



Kalman Filter

Optimal data processing algorithm

- Major use: **filter out noise** of measurement data (but can also be applied to other fields, e.g. Sensor Fusion)
- Result: Computes an optimal estimation of the **state** of an observed **system** based on measurements
- **Iterative**
- **Optimal**: incorporates all information (i.e. measurement data) that can be provided to it
- Does not need to keep all previous measurement data in **storage!**

Discrete Kalman Filter

- Modeled Process -

Discrete Kalman Filter

- Modeled Process -

- System state vector \mathbf{x}_k at time step k
 - $1 \times n$ vector
 - process noise \mathbf{w}_k with $p(\mathbf{w}) \sim N(0, Q)$ and $n \times n$ covariance matrix Q (often assumed to be constant)

Discrete Kalman Filter

- Modeled Process -

- System state vector \mathbf{x}_k at time step k
 - $1 \times n$ vector
 - process noise \mathbf{w}_k with $p(\mathbf{w}) \sim N(0, Q)$ and $n \times n$ covariance matrix Q (often assumed to be constant)
- State transition matrix A
 - $n \times n$ matrix
 - often assumed to be constant

Discrete Kalman Filter

- Modeled Process -

- System state vector \mathbf{x}_k at time step k
 - $1 \times n$ vector
 - process noise \mathbf{w}_k with $p(\mathbf{w}) \sim N(0, Q)$ and $n \times n$ covariance matrix Q (often assumed to be constant)
- State transition matrix A
 - $n \times n$ matrix
 - often assumed to be constant
- Control input vector \mathbf{u}_k (optional)
 - 1×1 vector
 - $n \times 1$ matrix B

Discrete Kalman Filter

- Modeled Process -

- System state vector \mathbf{x}_k at time step k
 - $1 \times n$ vector
 - process noise \mathbf{w}_k with $p(\mathbf{w}) \sim N(0, Q)$ and $n \times n$ covariance matrix Q (often assumed to be constant)
- State transition matrix A
 - $n \times n$ matrix
 - often assumed to be constant
- Control input vector \mathbf{u}_k (optional)
 - 1×1 vector
 - $n \times 1$ matrix B
- Incremental state change: $\mathbf{x}_k = A\mathbf{x}_{k-1} + B\mathbf{u}_k + \mathbf{w}_{k-1}$

Discrete Kalman Filter

- Modeled Process -

- Measurement vector \mathbf{z}_k
 - 1 x m vector
 - measurement noise \mathbf{v}_k with $p(\mathbf{v}) \sim \mathcal{N}(0, \mathbf{R})$ and m x m covariance matrix \mathbf{R} (often assumed to be constant)

Discrete Kalman Filter

- Modeled Process -

- Measurement vector \mathbf{z}_k
 - 1 x m vector
 - measurement noise \mathbf{v}_k with $p(\mathbf{v}) \sim \mathcal{N}(0, \mathbf{R})$ and m x m covariance matrix \mathbf{R} (often assumed to be constant)
- Measurement prediction matrix \mathbf{H}
 - m x n matrix
 - often assumed to be constant

Discrete Kalman Filter

- Modeled Process -

- Measurement vector \mathbf{z}_k
 - 1 x m vector
 - measurement noise \mathbf{v}_k with $p(\mathbf{v}) \sim \mathcal{N}(0, \mathbf{R})$ and m x m covariance matrix \mathbf{R} (often assumed to be constant)
- Measurement prediction matrix \mathbf{H}
 - m x n matrix
 - often assumed to be constant
- Measurement prediction: $\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k$

Discrete Kalman Filter

- Computing Model Parameters -

Discrete Kalman Filter

- Computing Model Parameters -

- Procedure:

- use best estimate $\hat{\mathbf{x}}_{k|1}$ of state \mathbf{x}_{k-1} at time step $k-1$

Discrete Kalman Filter

- Computing Model Parameters -

- Procedure:

- use best estimate $\hat{\mathbf{x}}_{k-1}$ of state \mathbf{x}_{k-1} at time step $k-1$
- predict state $\hat{\mathbf{x}}_k$ at time step k (“a priori state estimate”)

Discrete Kalman Filter

- Computing Model Parameters -

- Procedure:

- use best estimate $\hat{\mathbf{x}}_{k-1}$ of state \mathbf{x}_{k-1} at time step $k-1$
- predict state $\hat{\mathbf{x}}_k$ at time step k (“a priori state estimate”)
- obtain real measurement \mathbf{z}_k

Discrete Kalman Filter

- Computing Model Parameters -

- Procedure:

- use best estimate $\hat{\mathbf{x}}_{k-1}$ of state \mathbf{x}_{k-1} at time step $k-1$
- predict state $\hat{\mathbf{x}}_k$ at time step k (“a priori state estimate”)
- obtain real measurement \mathbf{z}_k
- compute state update $\hat{\mathbf{x}}_k$ at time step k (“a posteriori state estimate”) using gain matrix \mathbf{K}

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k + \mathbf{K}(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k)$$

Discrete Kalman Filter

- Computing Model Parameters -

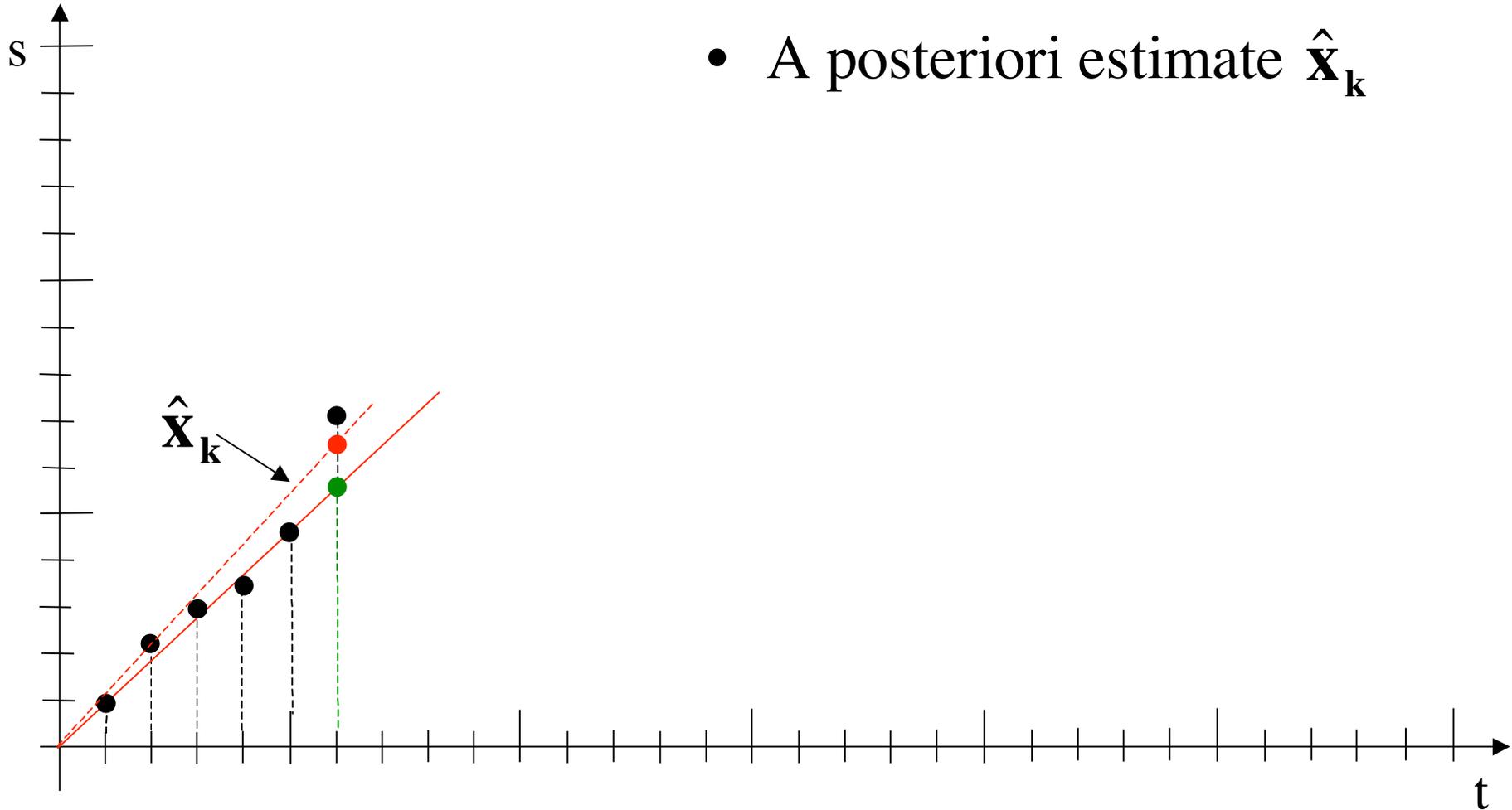
- A priori state estimate $\hat{\mathbf{x}}_k^{\square}$
- A priori estimate error $\mathbf{e}_k^{\square} = \mathbf{x}_k - \hat{\mathbf{x}}_k^{\square}$
- A priori estimate covariance $\mathbf{P}_k^{\square} = E[\mathbf{e}_k^{\square} \mathbf{e}_k^{\square T}]$

- A posteriori state estimate $\hat{\mathbf{x}}_k$
- A posteriori estimate error $\mathbf{e}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k$
- A posteriori estimate covariance $\mathbf{P}_k = E[\mathbf{e}_k \mathbf{e}_k^T]$

Discrete Kalman Filter

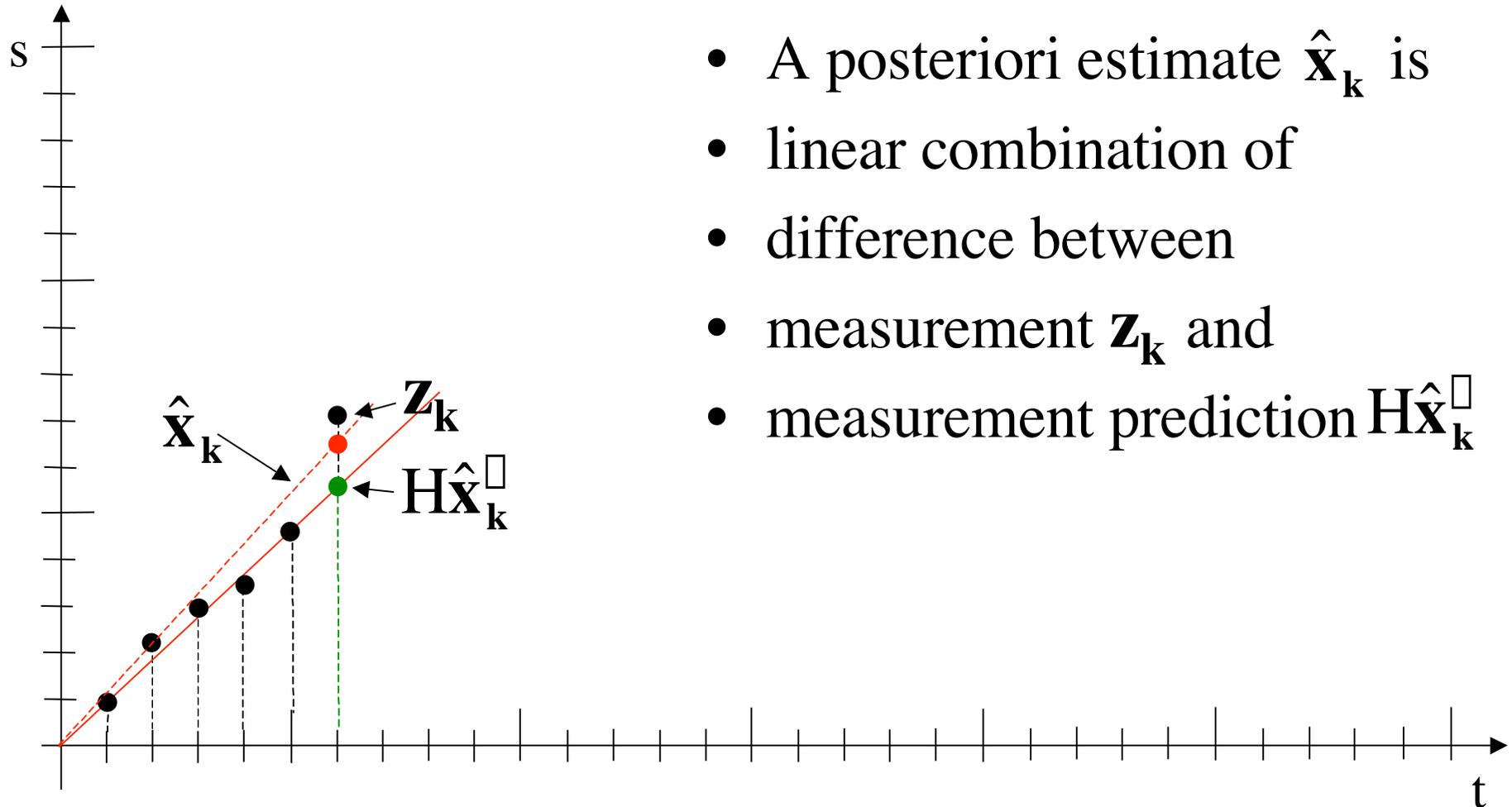
- Computing Model Parameters-

- A posteriori estimate $\hat{\mathbf{x}}_k$



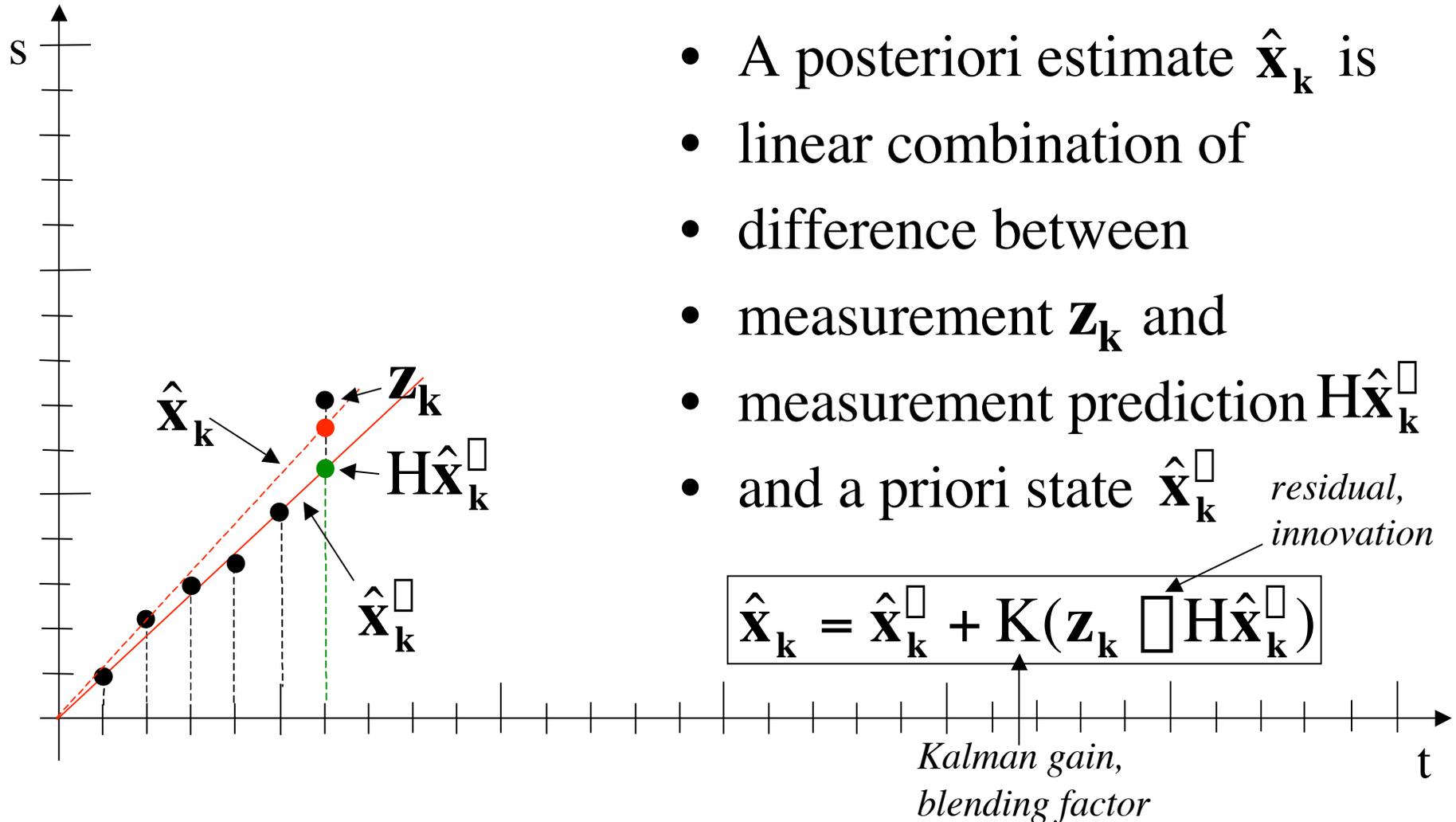
Discrete Kalman Filter

- Computing Model Parameters-



Discrete Kalman Filter

- Computing Model Parameters-



- A posteriori estimate $\hat{\mathbf{x}}_{\mathbf{k}}$ is
- linear combination of
- difference between
- measurement $\mathbf{z}_{\mathbf{k}}$ and
- measurement prediction $H\hat{\mathbf{x}}_{\mathbf{k}}^{\square}$
- and a priori state $\hat{\mathbf{x}}_{\mathbf{k}}^{\square}$

Discrete Kalman Filter

- Computing Model Parameters -

Kalman gain

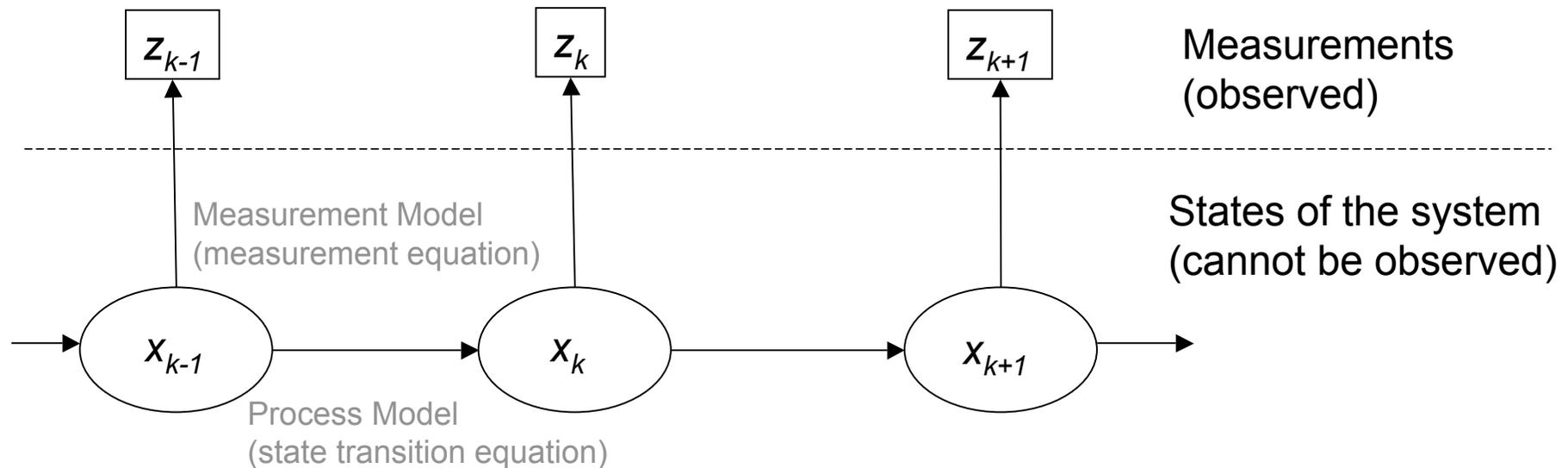
$$K = \frac{P_k H^T}{H P_k H^T + R}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k + K(\mathbf{z}_k - H\hat{\mathbf{x}}_k)$$

- $n \times m$ matrix
- minimizes the a posteriori error covariance equation $P_k = E[\mathbf{e}_k \mathbf{e}_k^T]$
- residual, when R small
- a priori estimate, when P_k small

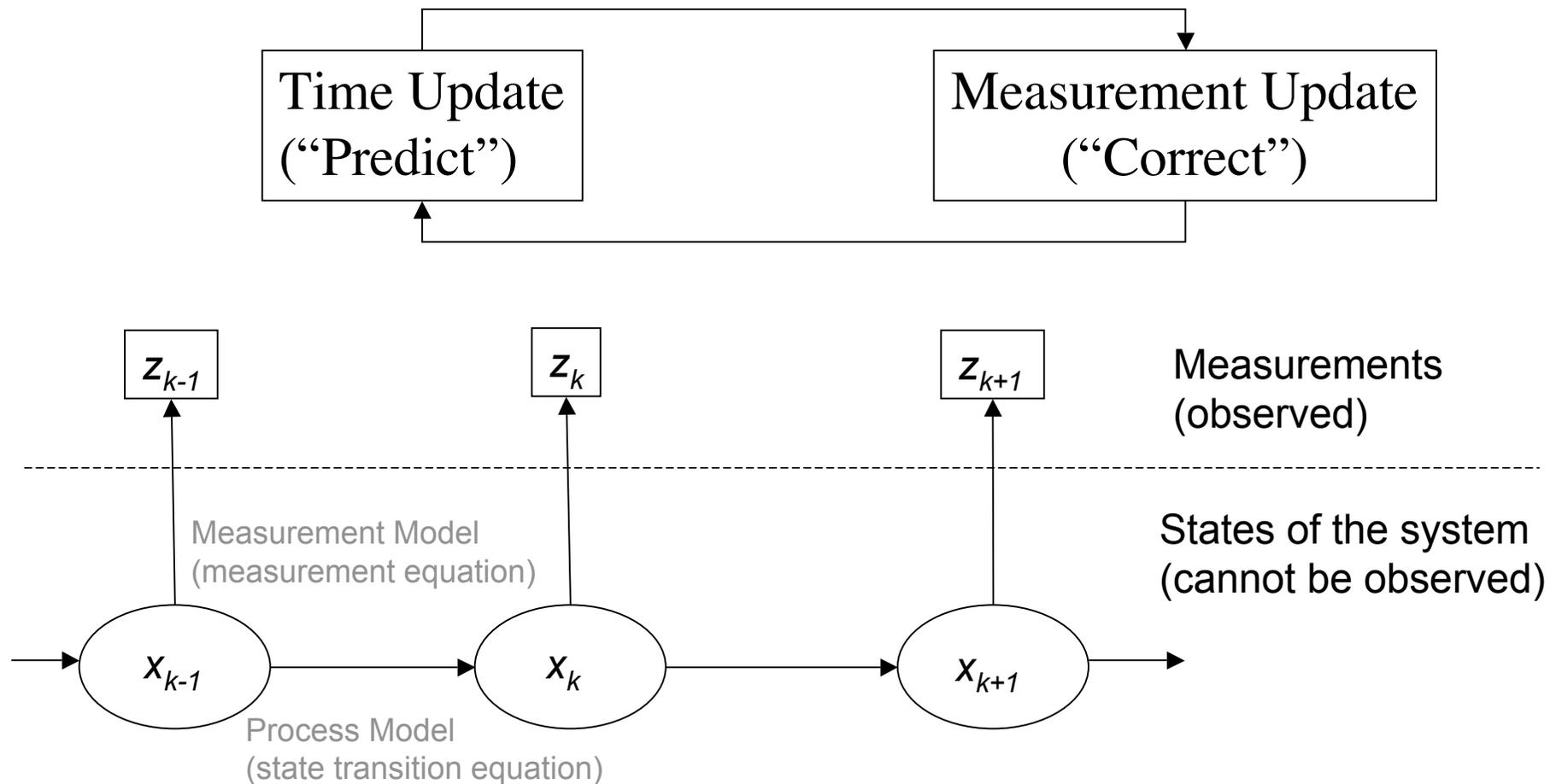
Discrete Kalman Filter

- Algorithm -



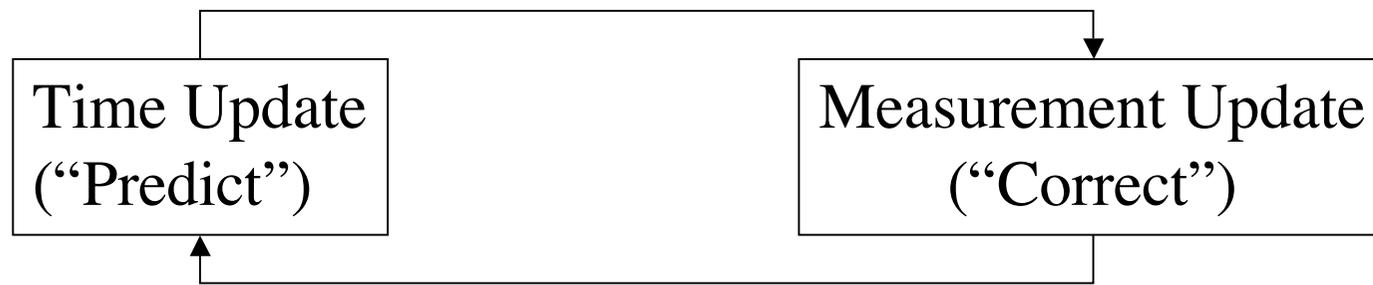
Discrete Kalman Filter

- Algorithm -



Discrete Kalman Filter

- Algorithm -



- Time update:
“predict”

$$\hat{\mathbf{x}}_k^{\square} = \mathbf{A}\hat{\mathbf{x}}_{k-1} + \mathbf{B}\mathbf{u}_k$$

$$\mathbf{P}_k^{\square} = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q}$$

- Measurement update:
“correct”

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

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^{\square} + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^{\square})$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^{\square}$$

Extended Kalman Filter (EKF)

- Non-linear process model
- Non-linear measurement model
- Linearize estimation around the current estimate using partial derivatives of the process and measurement functions
- Fundamental flaw: distributions (densities) of random variables are no longer normal
- EKF “ad hoc” state estimator that approximates optimality of Bayes’ rule by linearizations

How to use a Kalman Filter

- Find a state representation
- Find a process model
- Find a measurement model

There are many ways to apply a Kalman Filter, i.e. it depends on the chosen models!

Kalman Filter for Sensor Fusion

SCAAT (Welch and Bishop)

- State x : pose and derivatives

$$\vec{x}(t) = (x, y, z, \varphi, \theta, \psi, \dot{x}, \dot{y}, \dot{z}, \dot{\varphi}, \dot{\theta}, \dot{\psi})$$

- Process model:

$$\vec{x}(t) = A(\Delta t) \vec{x}(t - \Delta t) + \vec{w}(\Delta t)$$

- State transition via A :

$$y(t) = y(t - \Delta t) + \dot{y}(t) \Delta t;$$

$$\dot{y}(t) = \dot{y}(t - \Delta t);$$

- System noise:

$$\vec{w}_i(t) \sim N(0, Q_i(t))$$

Kalman Filter for Sensor Fusion

SCAAT (Welch and Bishop)

- Individual sensor model for sensor i

$$\vec{z}_i(t) = h_i(\vec{x}(t), \vec{b}(t), \vec{c}(t)) + \vec{v}_i(t)$$

- Measurement function h_i (with Jacobian H_i)

$$H_i(\hat{x}(t), b(t), c(t))[k, l] = \frac{\partial}{\partial \vec{x}[l]} h_i(\vec{x}(t), \vec{b}(t), \vec{c}(t))[k]$$

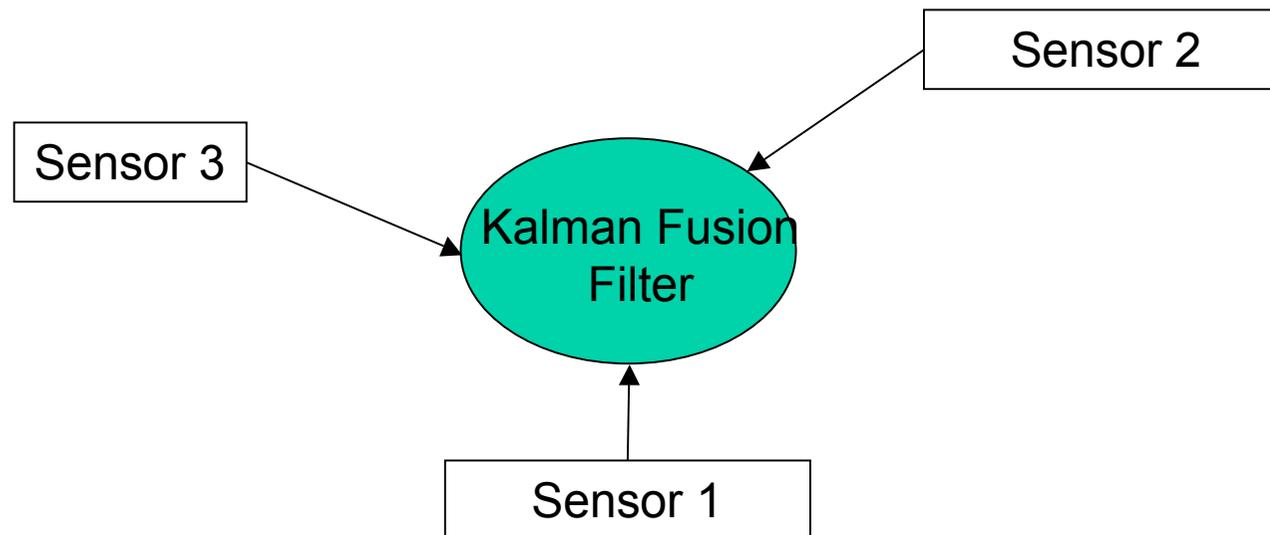
- Measurement noise

$$\vec{v}_i(t) \square N(0, R_i(t))$$

Kalman Filter for Sensor Fusion

SCAAT (Welch and Bishop)

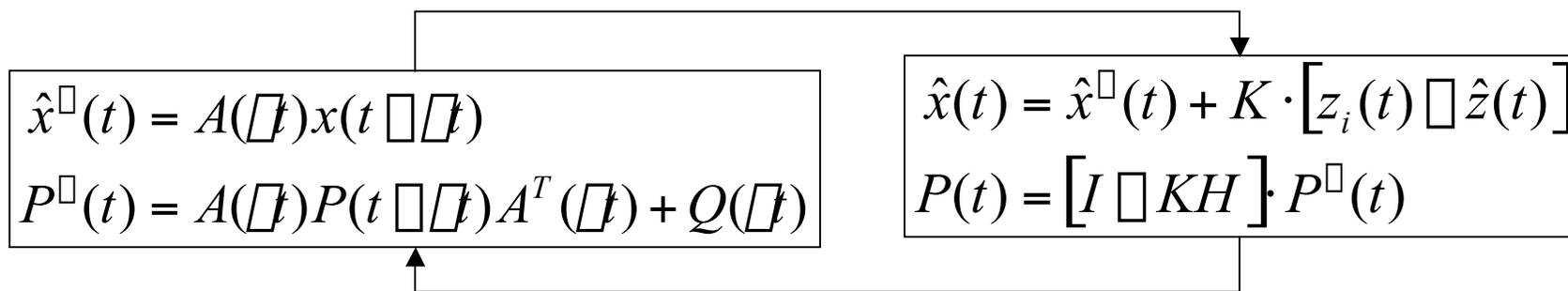
- “Single constraint at a time”
- Asynchronous algorithm
- Each time a new measurement z becomes available, a new estimate x is computed



Kalman Filter for Sensor Fusion

SCAAT (Welch and Bishop)

Algorithm:



1. Predict

2. Correct

$$K = \frac{P^\square(t)H^T}{HP^\square(t)H^T + R_i(t)}$$

Kalman Gain

$$\hat{z}(t) = h_i(\hat{x}^\square(t), b(t), c(t))$$

Predicted measurement i

$$H = H_i(\hat{x}^\square(t), b(t), c(t))$$

Corresponding Jacobian

