COMP0130 Robot Vision and Navigation

2B: The Kalman Filter and its use for GNSS

Dr Paul D Groves

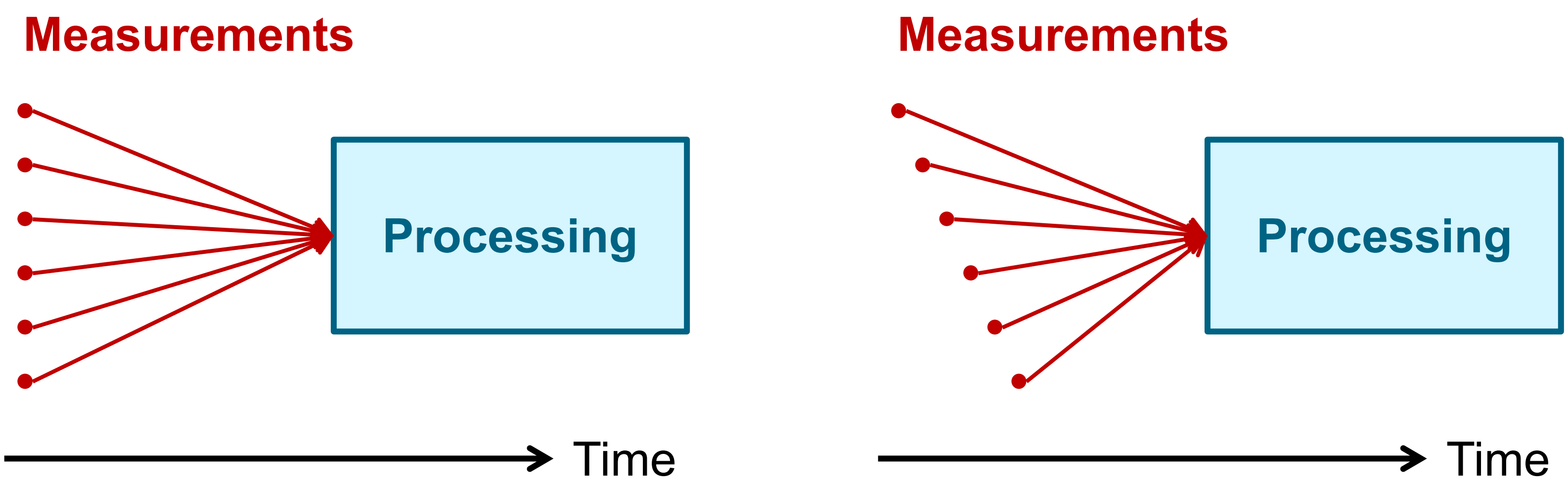
Lecture 2B Objectives

- Introduce sequential least-squares estimation to efficiently process measurements made at different times
- Introduce the Kalman filter for estimating time-varying states
- Apply the Kalman filter to GNSS positioning

Contents

1. Sequential Least-Squares

Simultaneous Estimation



UNTIL NOW We have assumed all measurements are processed simultaneously, which is appropriate if:

- All of your measurements are made at the same time, *OR*
- You collect data from an experiment and process it afterwards, *OR*
- You collect measurements on a site visit & process them back at base.

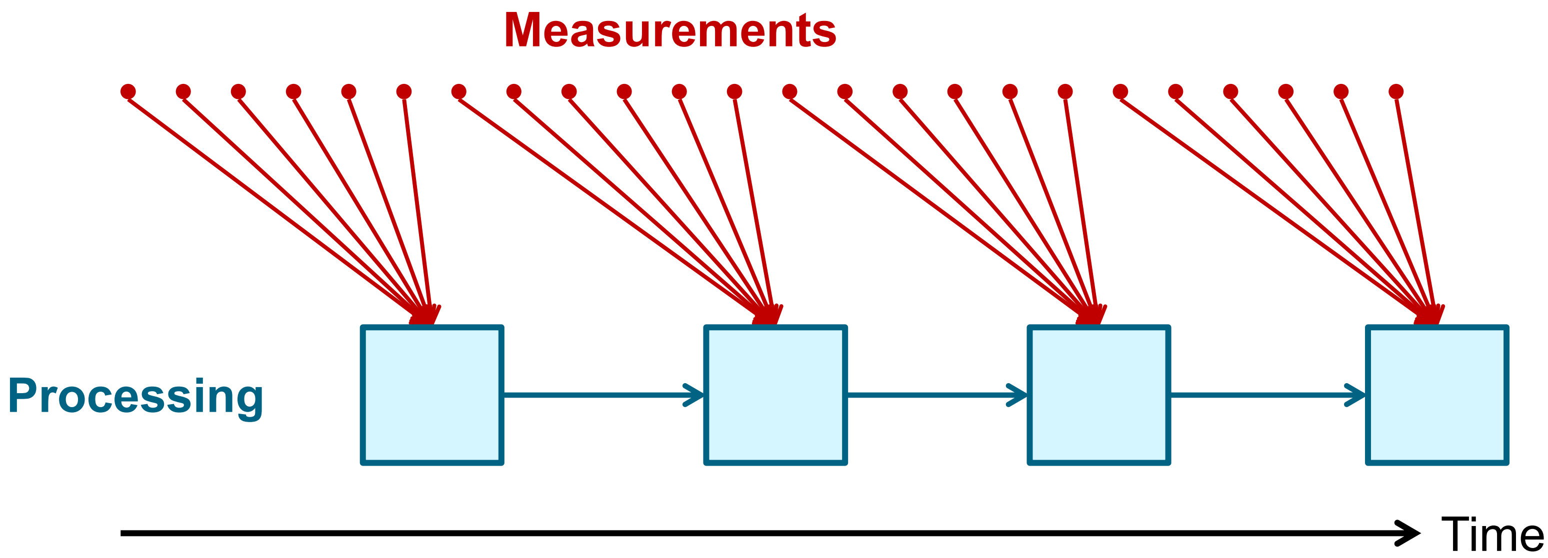
1. Sequential Least-Squares

Sequential Estimation Problems (1)

UNTIL NOW We have assumed all measurements are processed simultaneously.

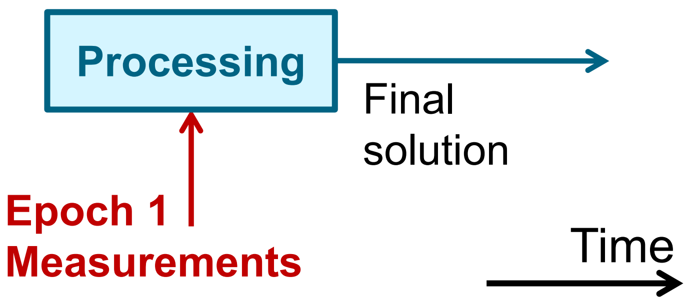
BUT What if the measurements become available at different times?

- **Example**: you are processing data in real time and want intermediate solutions to the state estimates to monitor progress.



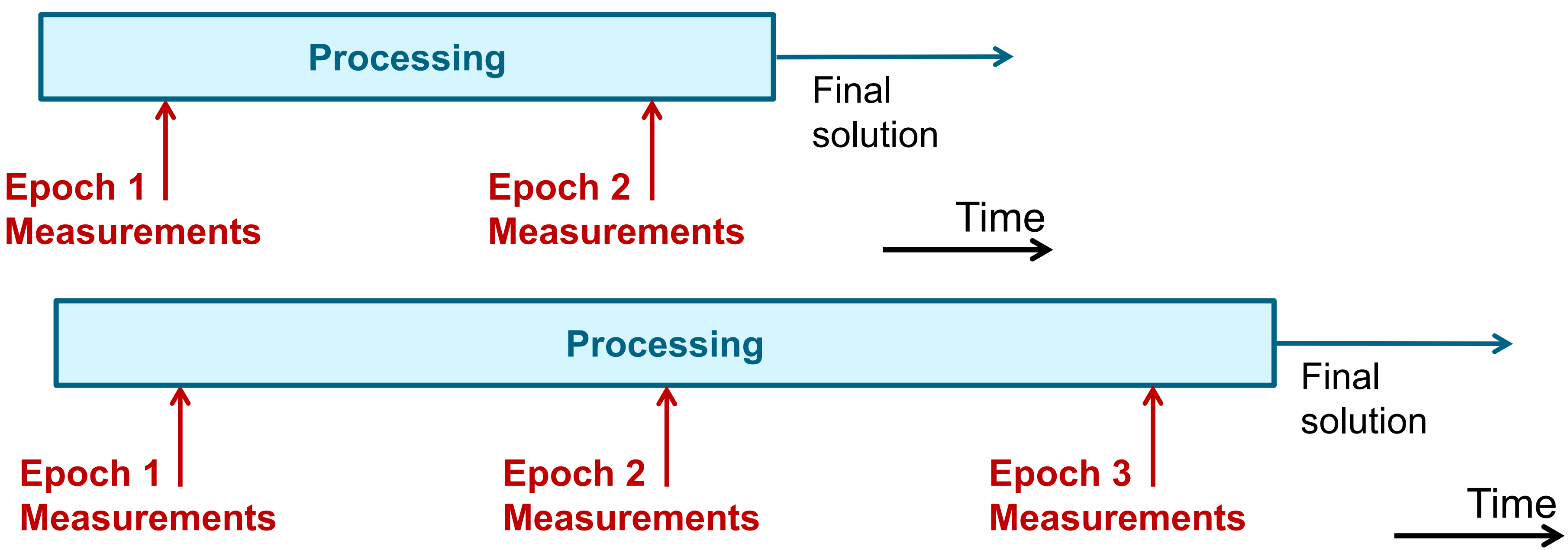
1. Sequential Least-Squares

Processing Sequential Measurements (1)



Using standard least-squares, we must compute a **complete new solution** whenever there are new measurements

Processing load thus increases with time

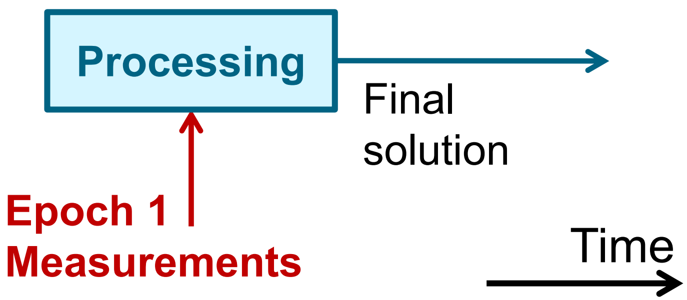


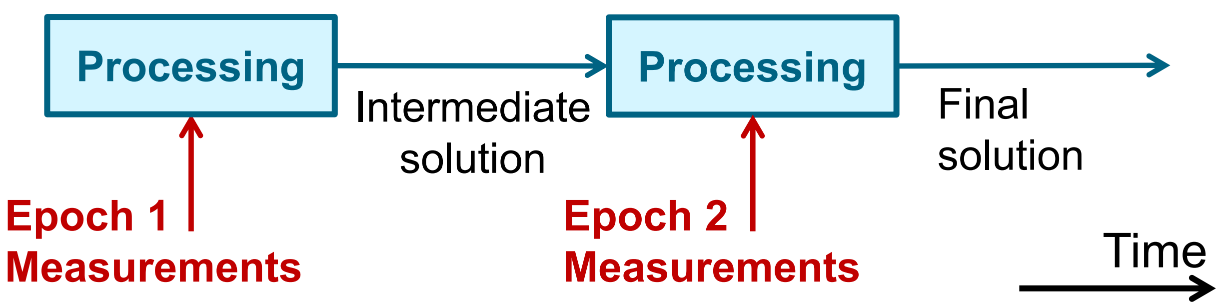
1. Sequential Least-Squares

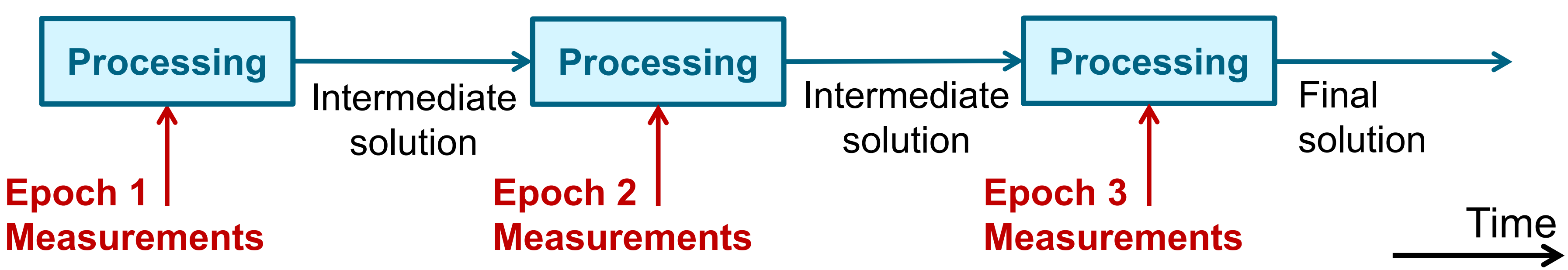
Processing Sequential Measurements (2)

Can we reformulate least-squares so that we only have to process the new measurements?

Processing load is independent of time

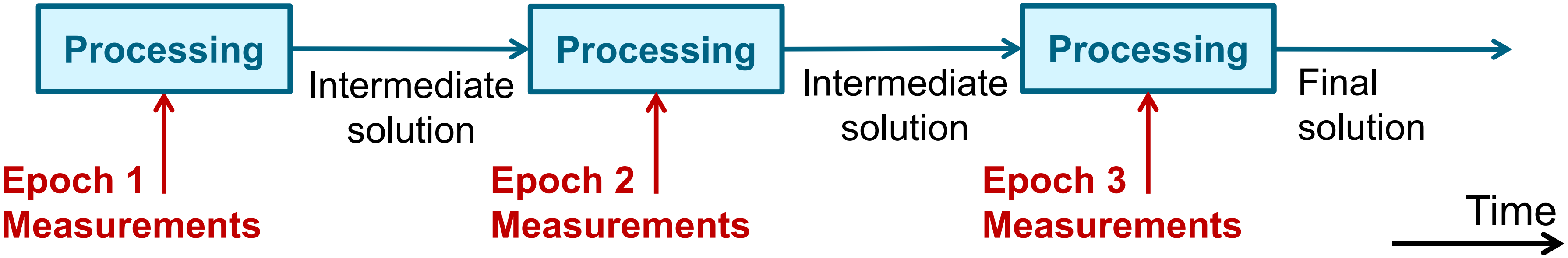






1. Sequential Least-Squares

Measurement Noise Covariance (1)



Successive sets of measurements may be processed sequentially **only** if they are **independent**, i.e.

$$\mathbf{C_z} = \begin{pmatrix} \mathbf{R}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}_3 \end{pmatrix}$$

$\mathbf{R}$ is the measurement ***noise*** covariance

- It only models errors that are uncorrelated across successive epochs
- Any time-correlated errors **must** be modelled as states (*see Week 3*)

A key assumption of the Sequential Least-Squares Derivation (4 on Moodle)

1. Sequential Least-Squares

Measurement Noise Covariance (2)

At epoch k

Epoch k measurement function

Measurements at epoch k

$$\tilde{\mathbf{z}}_k = \mathbf{h}_k(\mathbf{x}) + \mathbf{w}_{m,k}$$

Noise on epoch k Measurements

True states

The **measurement noise covariance matrix**, $\mathbf{R}_k$, is the expectation of the square of the measurement noise vector

$$\mathbf{R}_k = \mathrm{E}\left(\mathbf{w}_{m,k}\mathbf{w}_{m,k}^{\mathrm{T}}\right) = \mathrm{E}\left(\left(\tilde{\mathbf{z}}_k - \mathbf{h}_k(\mathbf{x})\right)\left(\tilde{\mathbf{z}}_k - \mathbf{h}_k(\mathbf{x})\right)^{\mathrm{T}}\right)$$

It may be

- The same for all epochs
- A function of geometry
- A function of dynamics
- A function of signal to noise

Measurement noise is uncorrelated between epochs

$$\mathrm{E}\left(\mathbf{w}_{m,k}\mathbf{w}_{m,j}^{\mathrm{T}}\right) = \mathbf{0} \quad j \neq k$$

1. Sequential Least-Squares

Introducing the Gain Matrix, K (1)

A state estimation problem may be expressed as

Measurements

Estimated states

$$\hat{\mathbf{x}}^{+} = \hat{\mathbf{x}}^{-} + \mathbf{K}\left(\tilde{\mathbf{z}} - \mathbf{h}\left(\hat{\mathbf{x}}^{-}\right)\right)$$

Predictions of measurements using predicted states

Predicted states

Gain matrix

Qualitatively...

$\mathbf{K}$ = Transformation from measurement domain to state domain × Weighting of each measurement in the state vector estimate

1. Sequential Least-Squares

Introducing the Gain Matrix, K (2)

For basic single-epoch least-squares..

$$\hat{\mathbf{x}}^{+} = \hat{\mathbf{x}}^{-} + \mathbf{K}\left(\tilde{\mathbf{z}} - \mathbf{h}\left(\hat{\mathbf{x}}^{-}\right)\right)$$

where

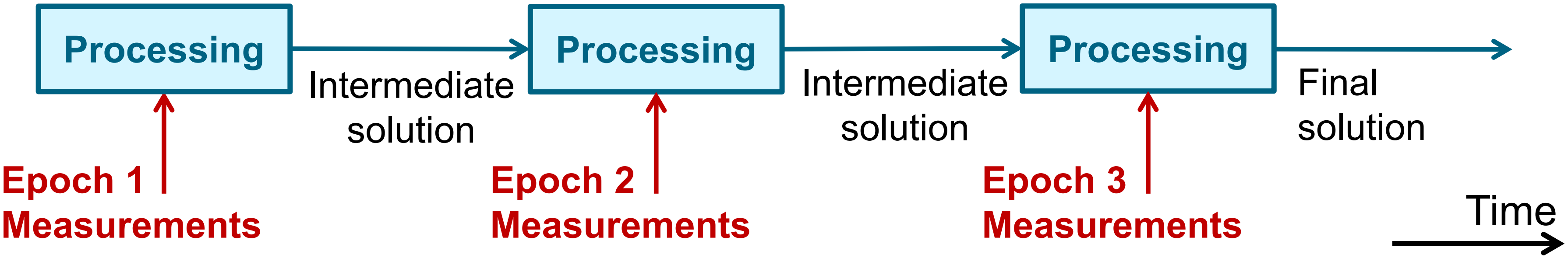
Gain matrix: $$\mathbf{K} = \left(\mathbf{H}^{\mathrm{T}}\mathbf{C}_{\mathbf{z}}^{-1}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{C}_{\mathbf{z}}^{-1}$$

Measurement matrix: $$\mathbf{H}(\hat{\mathbf{x}}^{-}) = \left.\frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}^{-}}$$

Measurement innovation: $$\mathbf{b} = \tilde{\mathbf{z}} - \mathbf{h}\left(\hat{\mathbf{x}}^{-}\right)$$

1. Sequential Least-Squares

Introducing the Gain Matrix, K (3)



For multi-epoch least-squares, we want to formulate the problem as

$$
\begin{aligned}
\hat{\mathbf{x}}_1^+ &= \hat{\mathbf{x}}^- + \mathbf{K}_1\left(\tilde{\mathbf{z}}_1 - \mathbf{h}_1\left(\hat{\mathbf{x}}^-\right)\right)\\
\hat{\mathbf{x}}_2^+ &= \hat{\mathbf{x}}_1^+ + \mathbf{K}_2\left(\tilde{\mathbf{z}}_2 - \mathbf{h}_2\left(\hat{\mathbf{x}}_1^+\right)\right)\\
\hat{\mathbf{x}}_3^+ &= \hat{\mathbf{x}}_2^+ + \mathbf{K}_3\left(\tilde{\mathbf{z}}_3 - \mathbf{h}_3\left(\hat{\mathbf{x}}_2^+\right)\right)\\
&\vdots\\
\hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_{k-1}^+ + \mathbf{K}_k\left(\tilde{\mathbf{z}}_k - \mathbf{h}_k\left(\hat{\mathbf{x}}_{k-1}^+\right)\right)
\end{aligned}
$$

How do we do this?

Formulating Sequential Least-Squares

For $k > 1$ (after the first epoch), there are state estimates from the previous epoch: $\hat{\mathbf{x}}_{k-1}^{+}$

Their error covariance is: $\mathbf{P}_{k-1}^{+} = \mathbf{C}_{\mathbf{x},k-1} = \mathrm{E}\left[\left(\hat{\mathbf{x}}_{k-1}^{+} - \mathbf{x}\right)\left(\hat{\mathbf{x}}_{k-1}^{+} - \mathbf{x}\right)^{\mathrm{T}}\right]$

We use these as the predicted states for the current epoch: $\hat{\mathbf{x}}_{k}^{-} = \hat{\mathbf{x}}_{k-1}^{+}$

For which we define an error covariance of:

$$\mathbf{P}_{k}^{-} = \mathrm{E}\left[\left(\hat{\mathbf{x}}_{k}^{-} - \mathbf{x}\right)\left(\hat{\mathbf{x}}_{k}^{-} - \mathbf{x}\right)^{\mathrm{T}}\right] = \mathrm{E}\left[\left(\hat{\mathbf{x}}_{k-1}^{+} - \mathbf{x}\right)\left(\hat{\mathbf{x}}_{k-1}^{+} - \mathbf{x}\right)^{\mathrm{T}}\right] = \mathbf{P}_{k-1}^{+}$$

We also have a set of measurements modelled by:

$$\tilde{\mathbf{z}}_{k} = \mathbf{h}_{k}\left(\mathbf{x}\right) + \mathbf{w}_{m,k} \qquad \mathrm{E}\left(\mathbf{w}_{m,k}\mathbf{w}_{m,k}^{\mathrm{T}}\right) = \mathbf{R}_{k}$$

How do obtain new state estimates, $\hat{\mathbf{x}}_{k}^{+}$, and their error covariance, $\mathbf{P}_{k}^{+}$?

1. Sequential Least-Squares

Sequential Least-Squares Derivation (1)

The new state estimates are a linear combination of the predicted states (= the previous state estimates) and the measurement innovation

$$\hat{\mathbf{x}}_k^+ = \mathbf{L}'_k \hat{\mathbf{x}}_k^- + \mathbf{K}_k \delta\mathbf{z}_k^-$$

Measurement innovation

$$\delta\mathbf{z}_k^- = \tilde{\mathbf{z}}_k - \mathbf{h}_k\left(\hat{\mathbf{x}}_k^-\right)$$

Derivation 4 on Moodle shows that $\mathbf{L}'_k = \mathbf{I}_k$, so

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\tilde{\mathbf{z}}_k - \mathbf{h}_k\left(\hat{\mathbf{x}}_k^-\right)\right)$$

This is the form we want, but how do we obtain $\mathbf{K}_k$?

From before: $\tilde{\mathbf{z}}_k = \mathbf{h}_k\left(\mathbf{x}\right) + \mathbf{w}_{m,k}$

Thus: $\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\mathbf{h}_k\left(\mathbf{x}\right) - \mathbf{h}_k\left(\hat{\mathbf{x}}_k^-\right) + \mathbf{w}_{m,k}\right)$

Sequential Least-Squares Derivation (2)

We have $\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\mathbf{h}_k(\mathbf{x}) - \mathbf{h}_k\left(\hat{\mathbf{x}}_k^-\right) + \mathbf{w}_{m,k}\right)$

where $\mathbf{K}_k$ is to be determined

We make the linearisation approximation

$$\mathbf{h}_k(\mathbf{x}) - \mathbf{h}_k\left(\hat{\mathbf{x}}_k^-\right) \approx \mathbf{H}_k\left(\mathbf{x} - \hat{\mathbf{x}}_k^-\right) \qquad \text{where} \quad \mathbf{H}_k = \left.\frac{\partial \mathbf{h}_k}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k^-}$$

Therefore: $\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left[\mathbf{H}_k\left(\mathbf{x} - \hat{\mathbf{x}}_k^-\right) + \mathbf{w}_{m,k}\right]$

Subtracting the true states, x, from both sides:

$$\begin{aligned}
\hat{\mathbf{x}}_k^+ - \mathbf{x} &= \hat{\mathbf{x}}_k^- - \mathbf{x} + \mathbf{K}_k\left[\mathbf{H}_k\left(\mathbf{x} - \hat{\mathbf{x}}_k^-\right) + \mathbf{w}_{m,k}\right] \\
&= \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\left(\hat{\mathbf{x}}_k^- - \mathbf{x}\right) + \mathbf{K}_k\mathbf{w}_{m,k}
\end{aligned}$$

1. Sequential Least-Squares

Sequential Least-Squares Derivation (3)

We have $\hat{\mathbf{x}}_k^+ - \mathbf{x} = \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k\right)\left(\hat{\mathbf{x}}_k^- - \mathbf{x}\right) + \mathbf{K}_k \mathbf{w}_{m,k}$

where $\mathbf{K}_k$ is to be determined

The state estimation error covariance is defined as

$$\mathbf{P}_k^+ = \mathbf{C}_{\mathbf{x},k} = \mathrm{E}\left[\left(\hat{\mathbf{x}}_k^+ - \mathbf{x}\right)\left(\hat{\mathbf{x}}_k^+ - \mathbf{x}\right)^{\mathrm{T}}\right]$$

Already defined: $\mathrm{E}\left[\left(\hat{\mathbf{x}}_k^- - \mathbf{x}\right)\left(\hat{\mathbf{x}}_k^- - \mathbf{x}\right)^{\mathrm{T}}\right] = \mathbf{P}_k^- \quad \mathrm{E}\left(\mathbf{w}_{m,k} \mathbf{w}_{m,k}^{\mathrm{T}}\right) = \mathbf{R}_k$

As measurement noise is uncorrelated over time,

$$\mathrm{E}\left[\mathbf{w}_{m,k}\left(\hat{\mathbf{x}}_k^- - \mathbf{x}\right)^{\mathrm{T}}\right] = \mathbf{0} \quad \mathrm{E}\left[\left(\hat{\mathbf{x}}_k^- - \mathbf{x}\right)\mathbf{w}_{m,k}^{\mathrm{T}}\right] = \mathbf{0}$$

Therefore, $\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_k^-(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^{\mathrm{T}} + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^{\mathrm{T}}$

see Derivation 4 on Moodle for details

1. Sequential Least-Squares

Sequential Least-Squares Derivation (4)

We have $\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^\mathrm{T} + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^\mathrm{T}$

where $\mathbf{K}_k$ is to be determined

We seek the value of $\mathbf{K}_k$ that minimises the state estimation errors

$$\frac{\partial}{\partial \mathbf{K}_k}\left[\mathrm{Tr}\left(\mathbf{P}_k^+\right)\right] = \mathbf{0}$$

where the trace of a matrix is $\mathrm{Tr}\left(\mathbf{A}\right) = \sum_i A_{ii}$

Following *Derivation 4* on Moodle,

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

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

Sequential Least-Squares Gain Matrix

The **Gain matrix**, $\mathbf{K}_k$, is used to determine the weighting of the measurement vector in updating the state estimates

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

Matrix inversion

Qualitatively...

$$\text{Transformation from measurement domain to state domain} \times \frac{\text{State variance in measurement domain}}{\text{State variance in measurement domain} + \text{Measurement variance}}$$

1. Sequential Least-Squares

Nonlinear Sequential Least-Squares Solution

Predict the states at the current epoch:

$$\boxed{\begin{matrix}\hat{\mathbf{x}}_k^- \\ \mathbf{P}_k^-\end{matrix}} \begin{matrix}= \\ =\end{matrix} \boxed{\begin{matrix}\hat{\mathbf{x}}_{k-1}^+ \\ \mathbf{P}_{k-1}^+\end{matrix}} \qquad \mathbf{P}_{k-1}^+ \equiv \mathbf{C}_{\mathbf{x},k-1}$$

Estimated state vector and error covariance from the previous epoch

Predicted state vector and error covariance for the current epoch

Measurement update:

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

$$\boxed{\hat{\mathbf{x}}_k^+} = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\tilde{\mathbf{z}}_k - \mathbf{h}_k\left(\hat{\mathbf{x}}_k^-\right)\right)$$

$$\boxed{\mathbf{P}_k^+} = \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{P}_k^- \qquad \mathbf{P}_k^+ \equiv \mathbf{C}_{\mathbf{x},k}$$

Updated state vector estimate and error covariance for the current epoch

1. Sequential Least-Squares

Linear Sequential Least-Squares Solution

Predict the states at the current epoch:

$$\boxed{\hat{\mathbf{x}}_k^-} = \boxed{\hat{\mathbf{x}}_{k-1}^+}$$

$$\boxed{\mathbf{P}_k^-} = \boxed{\mathbf{P}_{k-1}^+} \qquad \mathbf{P}_{k-1}^+ \equiv \mathbf{C}_{\mathbf{x},k-1}$$

Estimated state vector and error covariance from the previous epoch

Predicted state vector and error covariance for the current epoch

Measurement update:

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

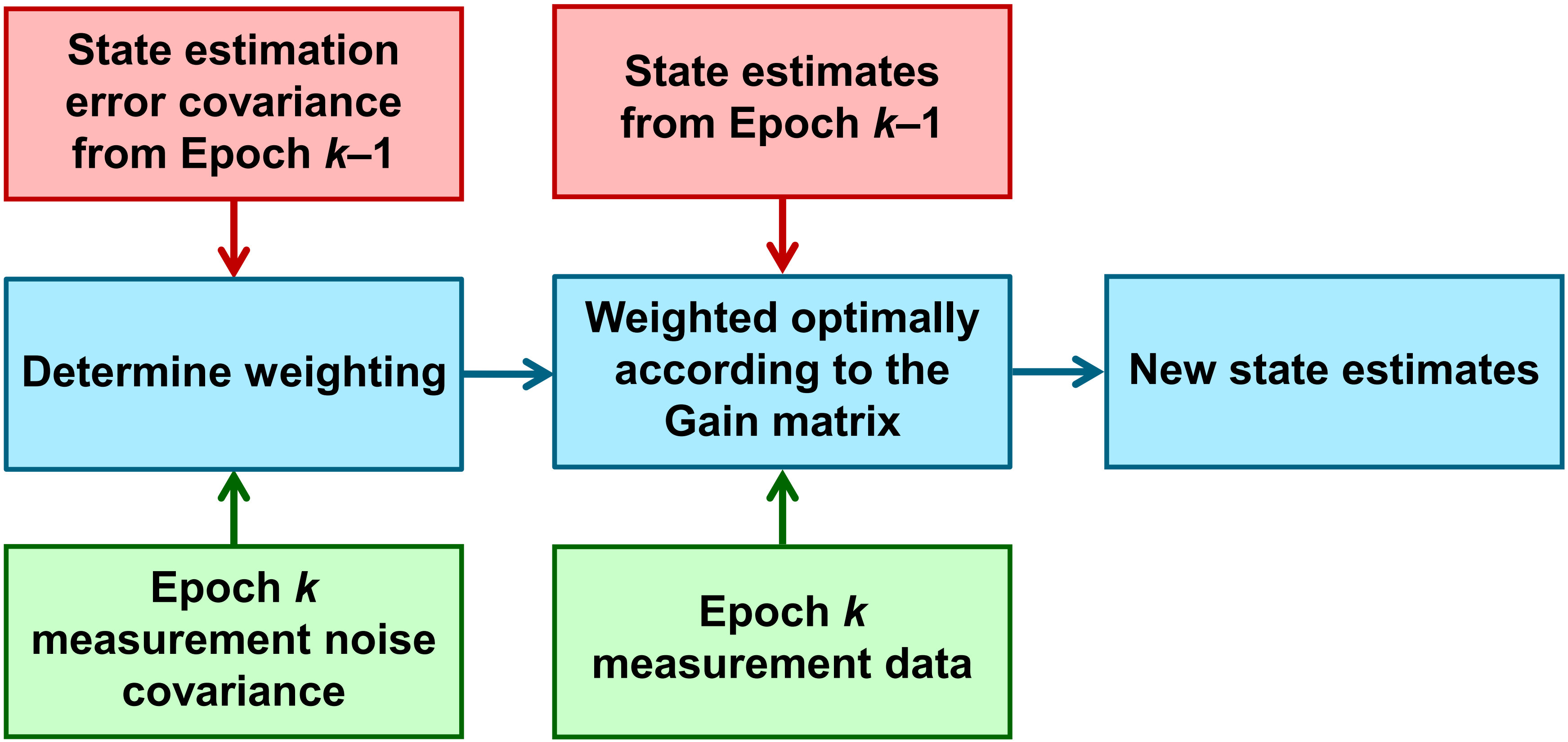
$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\tilde{\mathbf{z}}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-\right)$$

$$\mathbf{P}_k^+ = \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{P}_k^- \qquad \mathbf{P}_k^+ \equiv \mathbf{C}_{\mathbf{x},k}$$

Updated state vector estimate and error covariance for the current epoch

1. Sequential Least-Squares

Sequential Least-Squares Process



1. Sequential Least-Squares

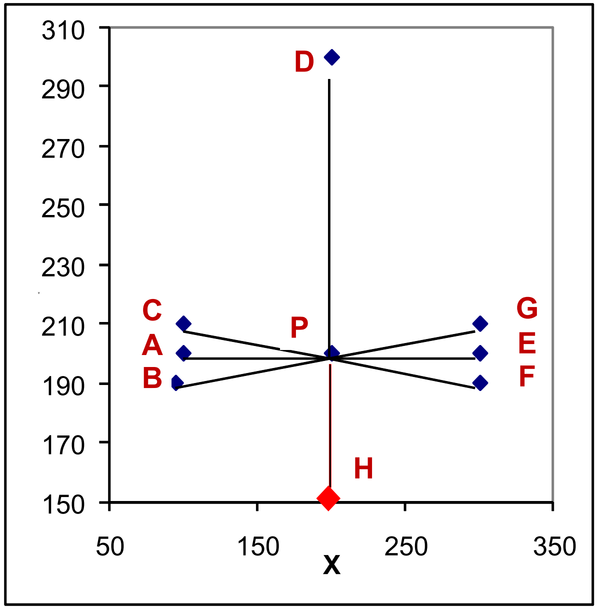
Example 1: Surveying using Ranging

Consider a ranging problem

The position of point **P** has already been estimated by measuring the distance from known points **A** to **G**, but the north-south accuracy is relatively poor

An extra distance measurement from known point **H** becomes available

This is added using sequential least-squares



*Uncertainty of Estimated position of Point **P**:*

	Easting SD	Northing SD
Before	0.005	0.013
After	0.005	0.008

See RVN KF Examples.xlsx on Moodle

Contents

2. Introduction to the Kalman filter

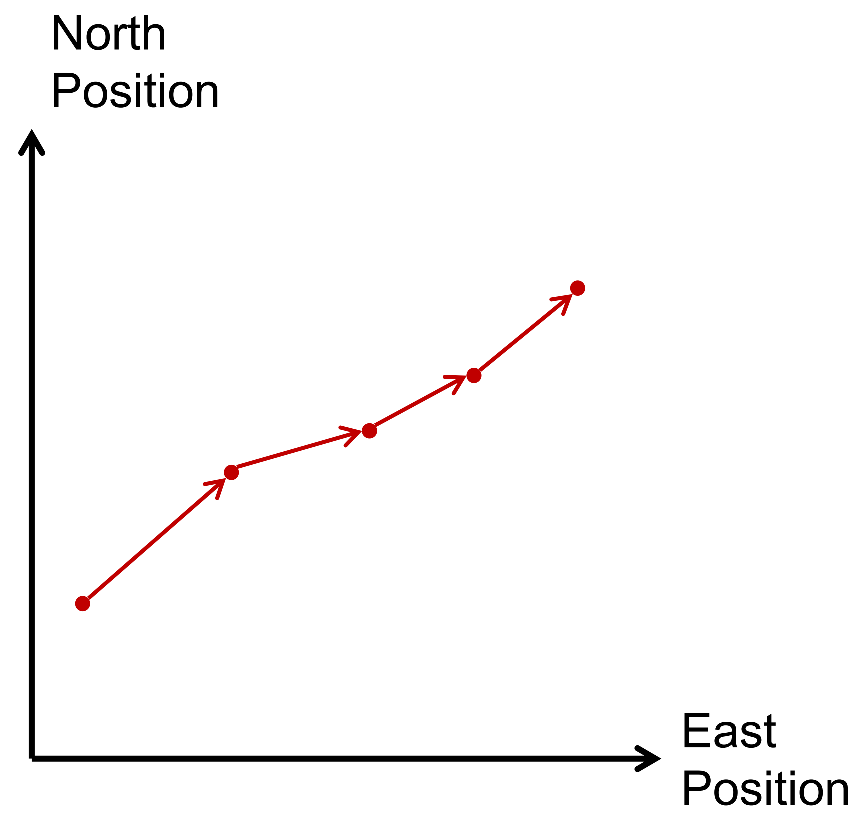
What if the states change with time?

UNTIL NOW We have assumed all states are constant.

Sequential least squares incorporates measurements made at different times

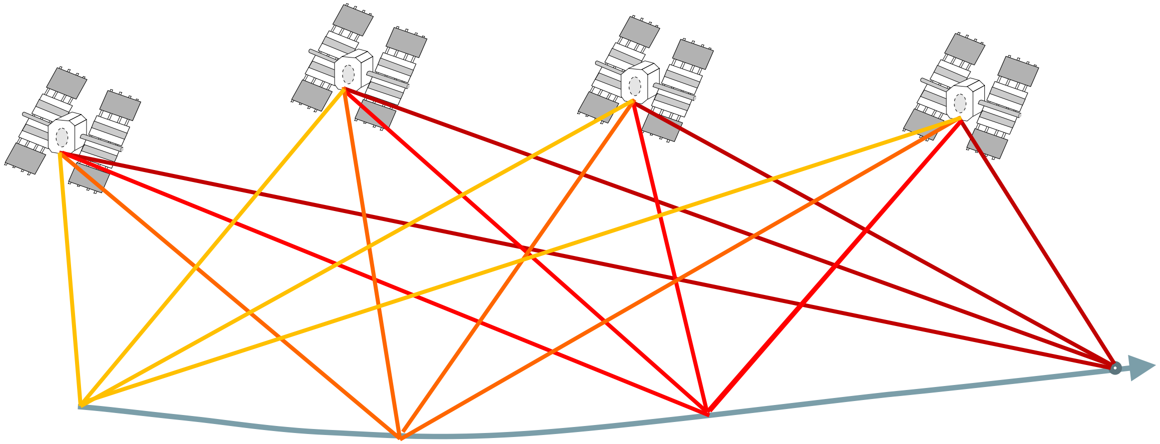
BUT What if the states change with time?

- E.g., the position of a moving object.



2. Introduction to the Kalman filter

Example A: GNSS Positioning



Estimated states

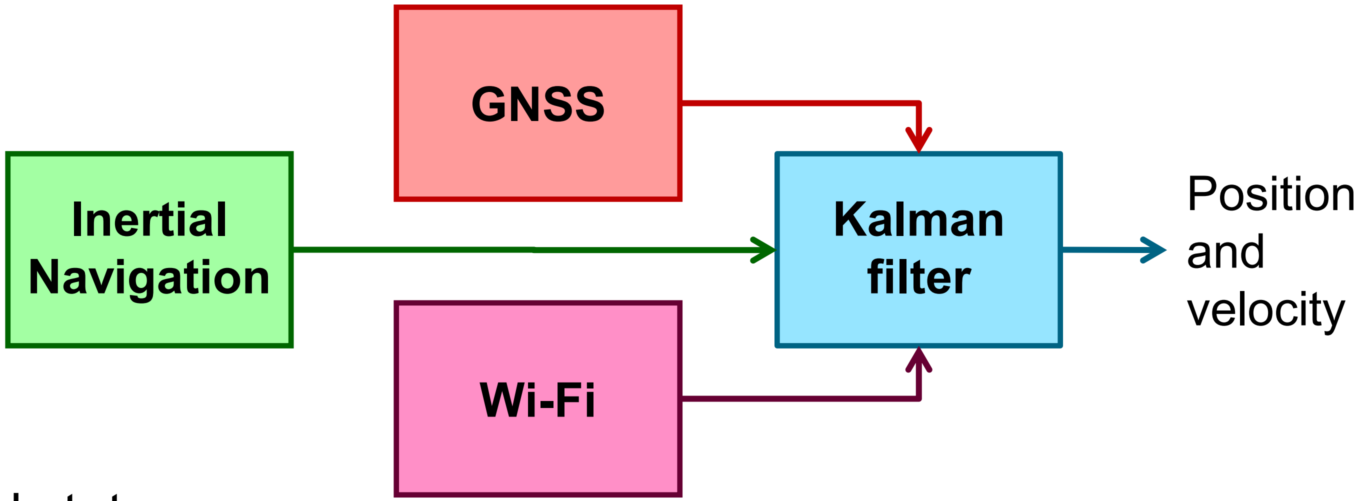
- User antenna position (3) and velocity (3)
- Receiver clock offset and drift

Measurements

- Pseudo-ranges (code)
- Pseudo-range rates (Doppler) or “carrier phase”

2. Introduction to the Kalman filter

Example B: Navigation sensor integration



Estimated states

- Inertial navigation system position error (3), velocity error (3) and attitude error (3)
- Inertial instrument errors

Measurements

- GNSS position and velocity (for a solution-domain approach)
- Wi-Fi position

2. Introduction to the Kalman filter

Further Applications

- Satellite Orbit Determination
- Structural health monitoring
- River flow and flood forecasting
- Finite element analysis
- Traffic management
- Air pollution monitoring
- Guidance and control of aircraft and spacecraft
- Tracking of objects using radar
- Chemical process monitoring
- Economic and financial modelling

What have these problems got in common?

1. We are estimating states from measurements
   - *How about least-squares estimation?*
2. But, measurements are made at multiple epochs
   - *How about sequential least-squares?*
3. But, the states are changing with time

   e.g. the position of moving objects
   - *With least-squares estimation, we would need additional states for every epoch*
   - *The problem would quickly become unmanageable*

We need a new approach

2. Introduction to the Kalman filter

Adapting Sequential Least-Squares

Predict the states at the current epoch:

$$\boxed{\begin{matrix}\hat{\mathbf{x}}_k^- \\ \mathbf{P}_k^-\end{matrix}} \begin{matrix}= \\ =\end{matrix} \boxed{\begin{matrix}\hat{\mathbf{x}}_{k-1}^+ \\ \mathbf{P}_{k-1}^+\end{matrix}}$$

Estimated state vector and error covariance from the previous epoch

Predicted state vector and error covariance for the current epoch

Can we modify the prediction stage to model time variation of the states?

Linear measurement update:

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

$$\boxed{\begin{matrix}\hat{\mathbf{x}}_k^+ \\ \mathbf{P}_k^+\end{matrix}} \begin{matrix}= \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left(\tilde{\mathbf{z}}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^- \right) \\ = \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k \right) \mathbf{P}_k^-\end{matrix}$$

Updated state vector estimate and error covariance for the current epoch

Unknown Change in the States

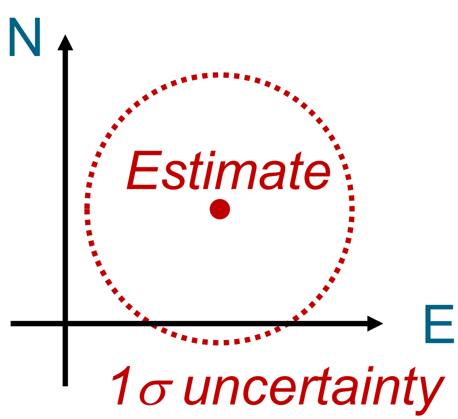
Example: We know that the position of an object has changed, but not by how much.

$\therefore$ We increase the state uncertainty

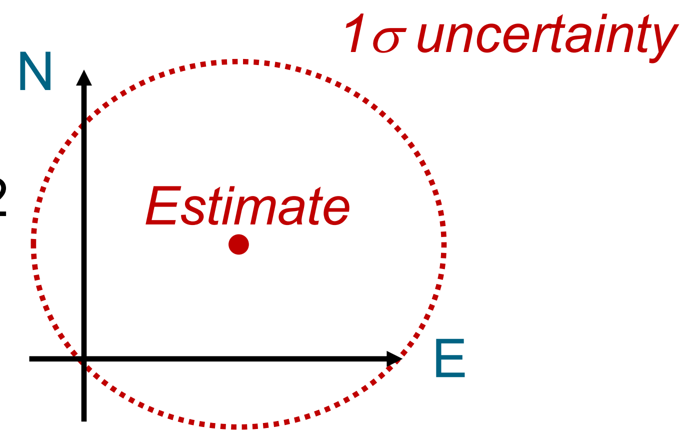
$$\begin{aligned} \hat{\mathbf{x}}_k^- &= \hat{\mathbf{x}}_{k-1}^+ \\ \mathbf{P}_k^- &= \mathbf{P}_{k-1}^+ \end{aligned} \quad \Rightarrow \quad \begin{aligned} \hat{\mathbf{x}}_k^- &= \hat{\mathbf{x}}_{k-1}^+ \\ \mathbf{P}_k^- &= \mathbf{P}_{k-1}^+ + \boxed{\mathbf{Q}_{k-1}} \end{aligned}$$

System Noise Covariance models the additional uncertainty

End of epoch 1



Start of epoch 2



Known Change in the States

Example: We know what the velocity, **v**, is.

∴ We use this to update our position estimate

$$\hat{E}_k^- = \hat{E}_{k-1}^+ + \hat{v}_{E,k-1}^+ \tau_s$$

$$\hat{N}_k^- = \hat{N}_{k-1}^+ + \hat{v}_{N,k-1}^+ \tau_s$$

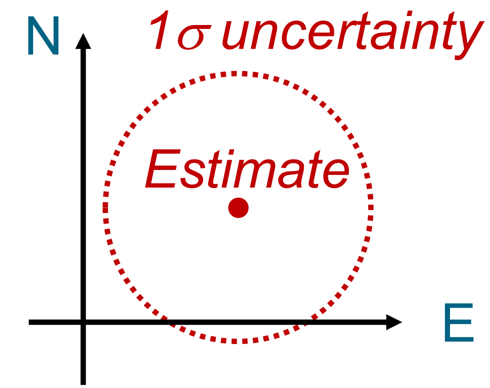
Propagation time interval

$$\Rightarrow \begin{pmatrix} \hat{E}_k^- \\ \hat{N}_k^- \\ \hat{v}_{E,k}^- \\ \hat{v}_{N,k}^- \end{pmatrix} = \begin{pmatrix} 1 & 0 & \tau_s & 0 \\ 0 & 1 & 0 & \tau_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \hat{E}_{k-1}^+ \\ \hat{N}_{k-1}^+ \\ \hat{v}_{E,k-1}^+ \\ \hat{v}_{N,k-1}^+ \end{pmatrix}$$

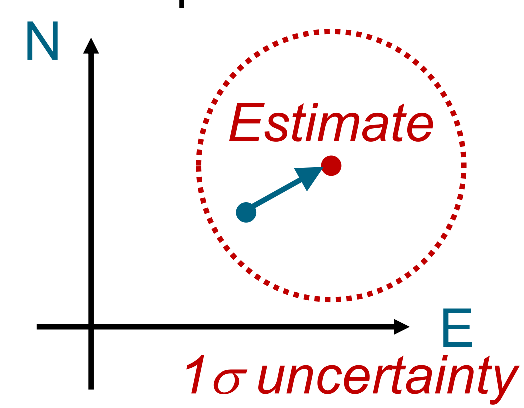
$$\Rightarrow \hat{\mathbf{x}}_k^- = \boxed{\mathbf{\Phi}_{k-1}} \hat{\mathbf{x}}_{k-1}^+$$

Transition matrix

End of epoch 1



Start of epoch 2



Transition Matrix

The **transition matrix**, $\mathbf{\Phi}_{k\text{-}1}$, defines how the state vector changes with time as a function of the dynamics of the system

E.g. position varies as the integral of velocity

Predicted states at epoch *k*

$$\hat{\mathbf{x}}_k^- = \mathbf{\Phi}_{k-1}\hat{\mathbf{x}}_{k-1}^+$$

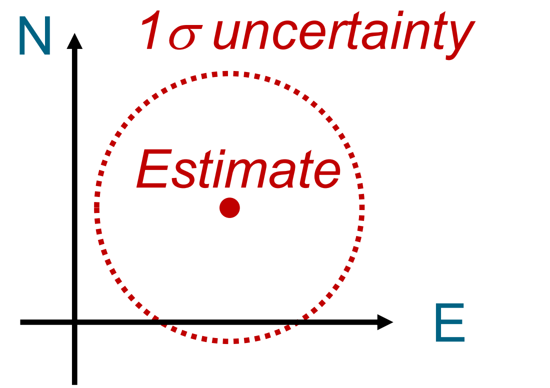
Estimated states at epoch *k*–1

The transition matrix is **always** a function of the time interval

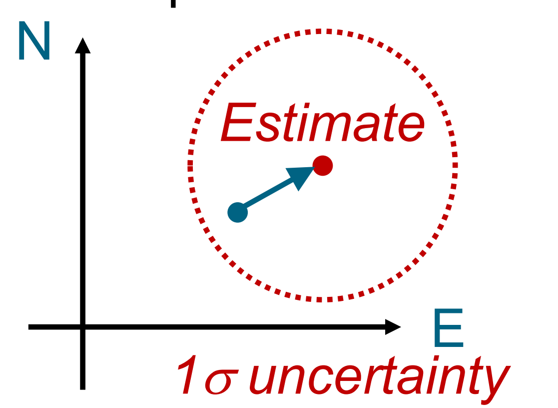
It is **never** a function of the state estimates for a standard Kalman Filter

It may be a function of other parameters, such as kinematics

End of epoch 1



Start of epoch 2



2. Introduction to the Kalman filter

Error Covariance Propagation (Known changes)

The transition matrix is used to propagate the state estimates forward in time.

It also applies to the true states

$$\hat{\mathbf{x}}_k^- = \mathbf{\Phi}_{k-1}\hat{\mathbf{x}}_{k-1}^+ \qquad \mathbf{x}_k = \mathbf{\Phi}_{k-1}\mathbf{x}_{k-1}$$

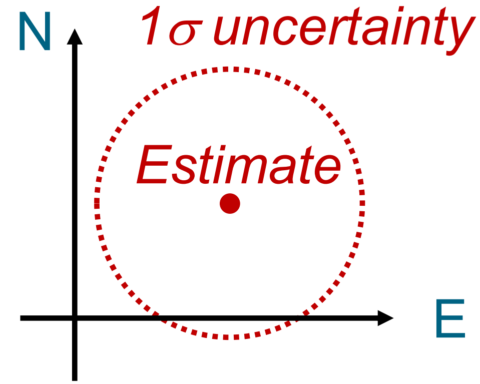
But what about the state **error covariance**?

$$\mathbf{P}_k^- = \mathrm{E}\left[\left(\hat{\mathbf{x}}_k^- - \mathbf{x}_k\right)\left(\hat{\mathbf{x}}_k^- - \mathbf{x}_k\right)^\mathrm{T}\right]$$

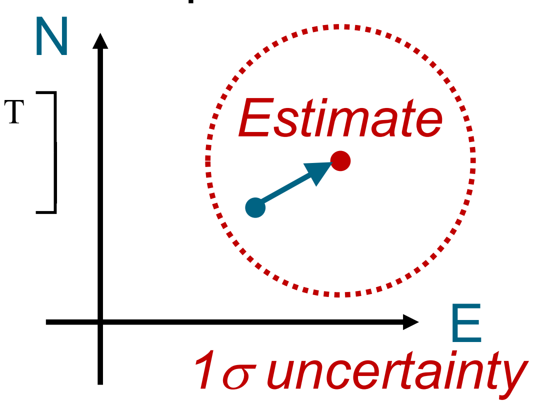
Applying the transition matrix and rearranging:

$$\begin{aligned}\mathbf{P}_k^- &= \mathrm{E}\left[\left(\mathbf{\Phi}_{k-1}\hat{\mathbf{x}}_{k-1}^+ - \mathbf{\Phi}_{k-1}\mathbf{x}_{k-1}\right)\left(\mathbf{\Phi}_{k-1}\hat{\mathbf{x}}_{k-1}^+ - \mathbf{\Phi}_{k-1}\mathbf{x}_{k-1}\right)^\mathrm{T}\right]\\ &= \mathbf{\Phi}_{k-1}\mathrm{E}\left[\left(\hat{\mathbf{x}}_{k-1}^+ - \mathbf{x}_{k-1}\right)\left(\hat{\mathbf{x}}_{k-1}^+ - \mathbf{x}_{k-1}\right)^\mathrm{T}\right]\mathbf{\Phi}_{k-1}^\mathrm{T}\\ &= \mathbf{\Phi}_{k-1}\mathbf{P}_{k-1}^+\mathbf{\Phi}_{k-1}^\mathrm{T}\end{aligned}$$

End of epoch 1



Start of epoch 2



Known and Unknown Changes in the States

Propagating the state estimates forward in time:

$$\hat{\mathbf{x}}_k^- = \boxed{\mathbf{\Phi}_{k-1}\hat{\mathbf{x}}_{k-1}^+}$$
$$\mathbf{P}_k^- = \boxed{\mathbf{\Phi}_{k-1}\mathbf{P}_{k-1}^+\mathbf{\Phi}_{k-1}^{\mathrm{T}}} + \boxed{\mathbf{Q}_{k-1}}$$

Unknown changes

Known changes

Incorporating the measurement:

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^{\mathrm{T}}\left(\mathbf{R}_k + \mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^{\mathrm{T}}\right)^{-1}$$
$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\tilde{\mathbf{z}}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-\right)$$
$$\mathbf{P}_k^+ = \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{P}_k^-$$

The standard Kalman filter applies only to problems where the measurements are a linear function of the states, i.e.

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k) = \mathbf{H}_k\mathbf{x}_k$$

This is the Kalman Filter

2. Introduction to the Kalman filter

Phases of The Kalman Filter

State Propagation

$$\hat{\mathbf{x}}_k^- = \mathbf{\Phi}_{k-1}\hat{\mathbf{x}}_{k-1}^+$$

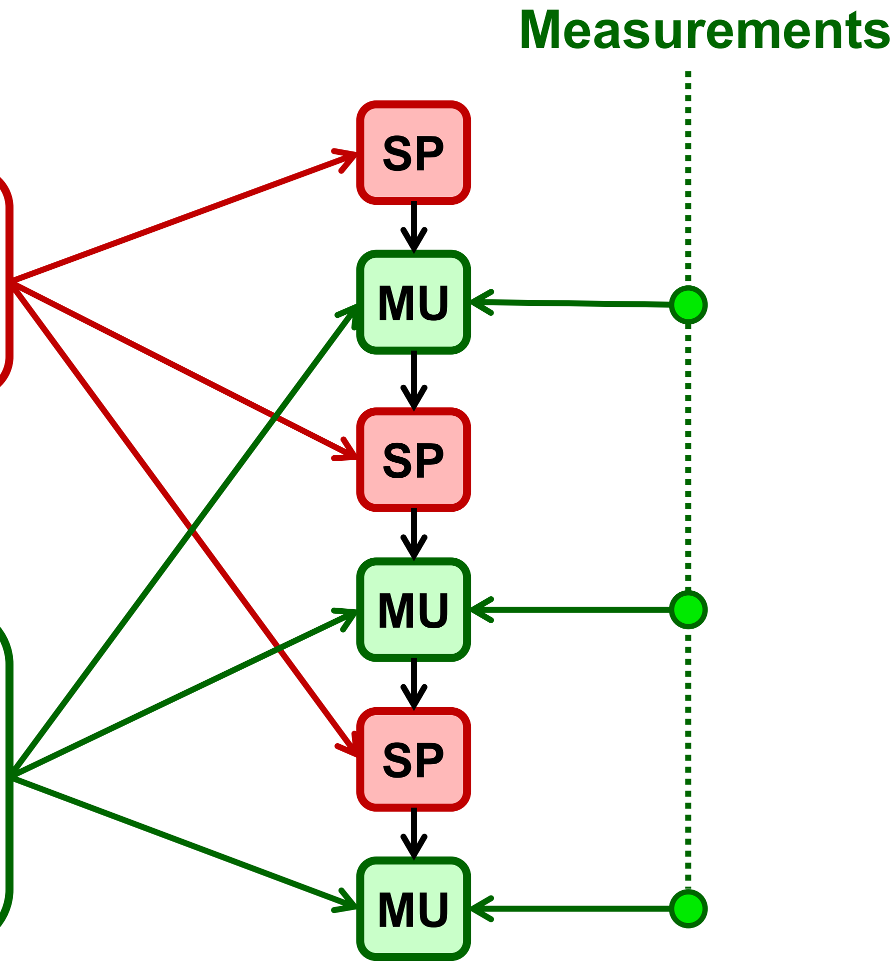
$$\mathbf{P}_k^- = \mathbf{\Phi}_{k-1}\mathbf{P}_{k-1}^+\mathbf{\Phi}_{k-1}^{\mathrm{T}} + \mathbf{Q}_{k-1}$$

Measurement Update

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

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

$$\mathbf{P}_k^+ = \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{P}_k^-$$



2. Introduction to the Kalman filter

Kalman Filter Components: Measurements

Measurement Model

Describes how each measurement varies as a function of the states

E.g. a range measurement is related to the user position by the line-of-sight unit vector

Measurement Data

A set of simultaneous measurements of the system

E.g. ranging measurements or a position estimate

Repeated at regular intervals

System Model

Initial state estimates

Kalman Filter Algorithm

State Estimates *Over Time*

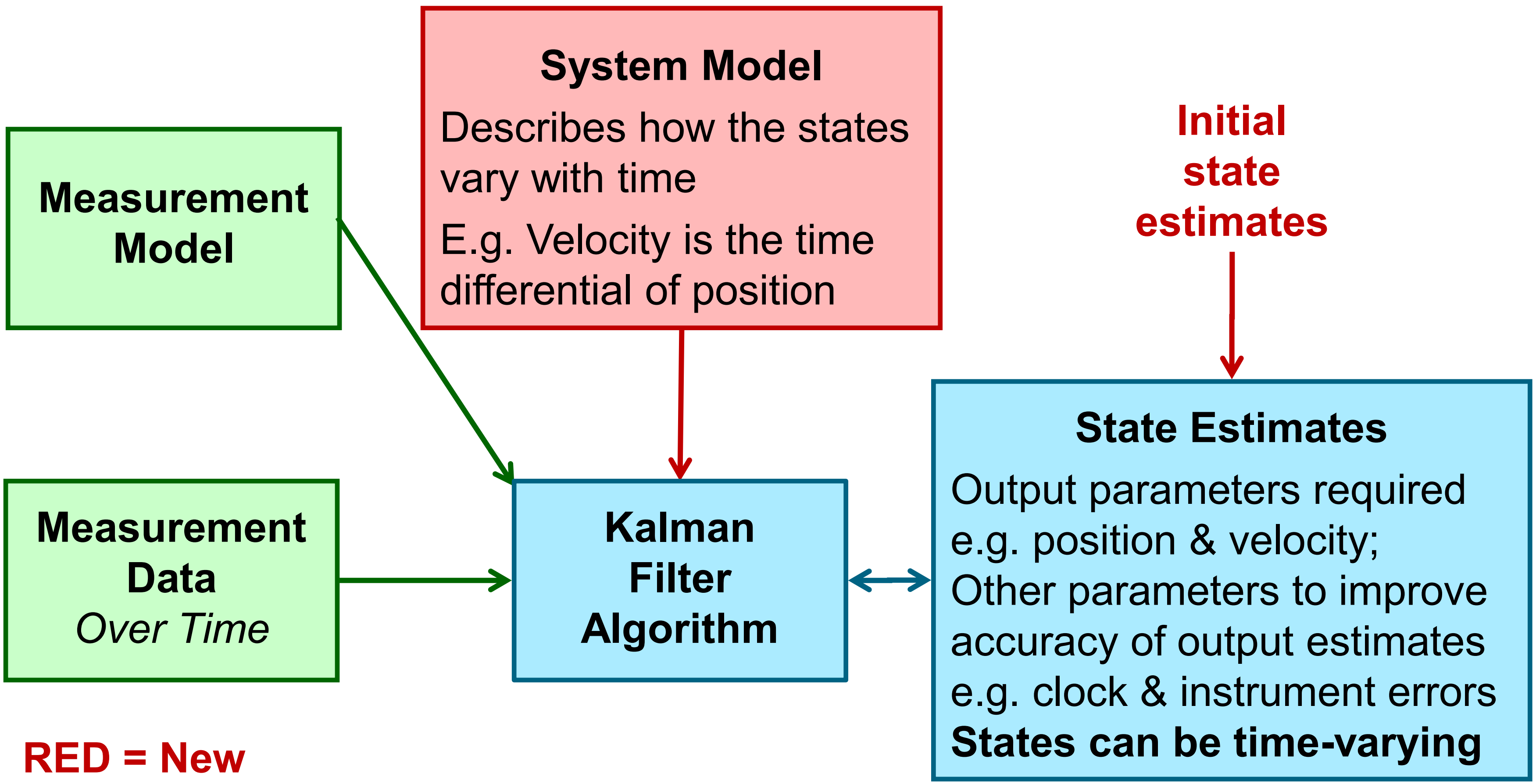
RED = New

Green = As Sequential Least-squares

Blue = Changed

2. Introduction to the Kalman filter

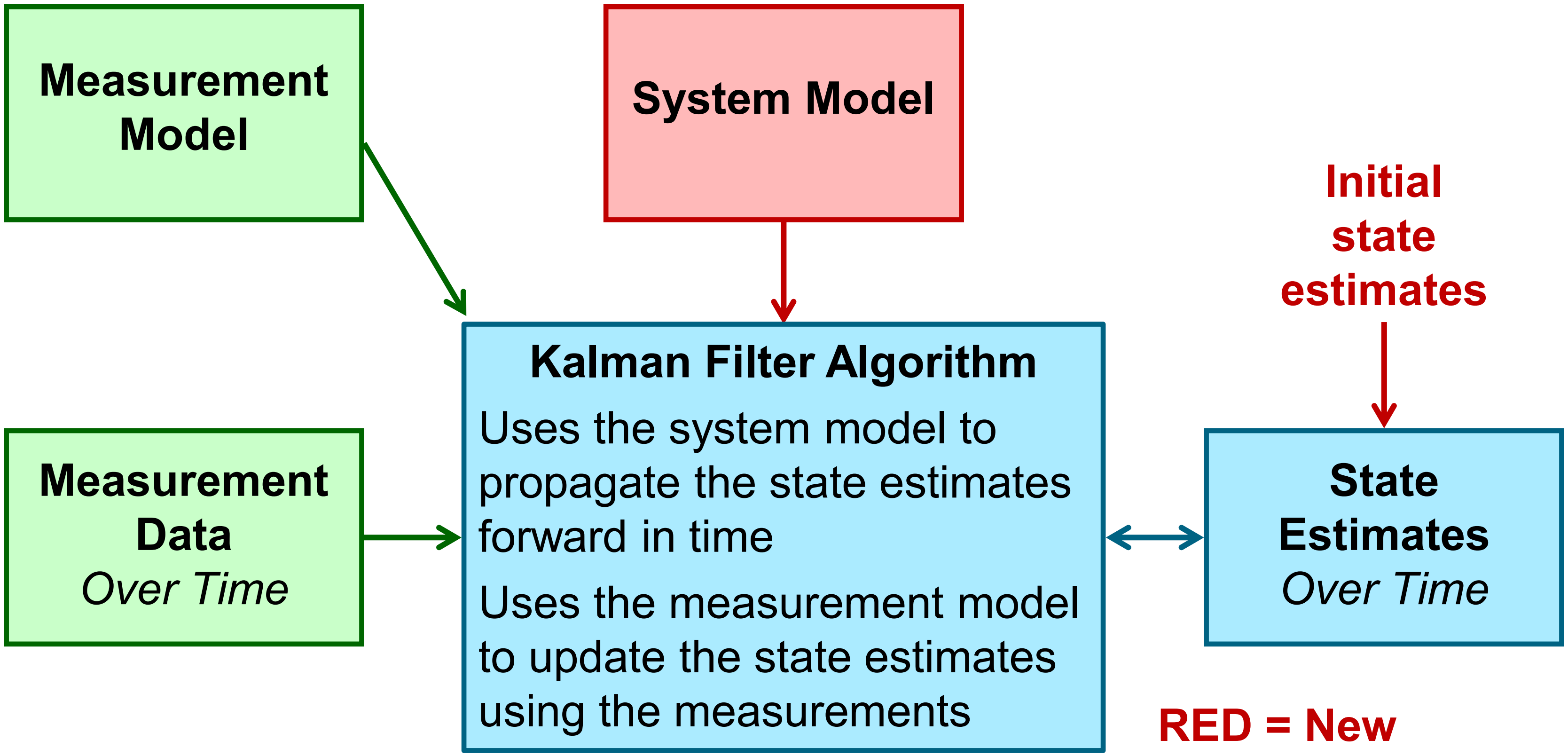
Kalman Filter Components: States



RED = New
Green = As Sequential Least-squares
Blue = Changed

2. Introduction to the Kalman filter

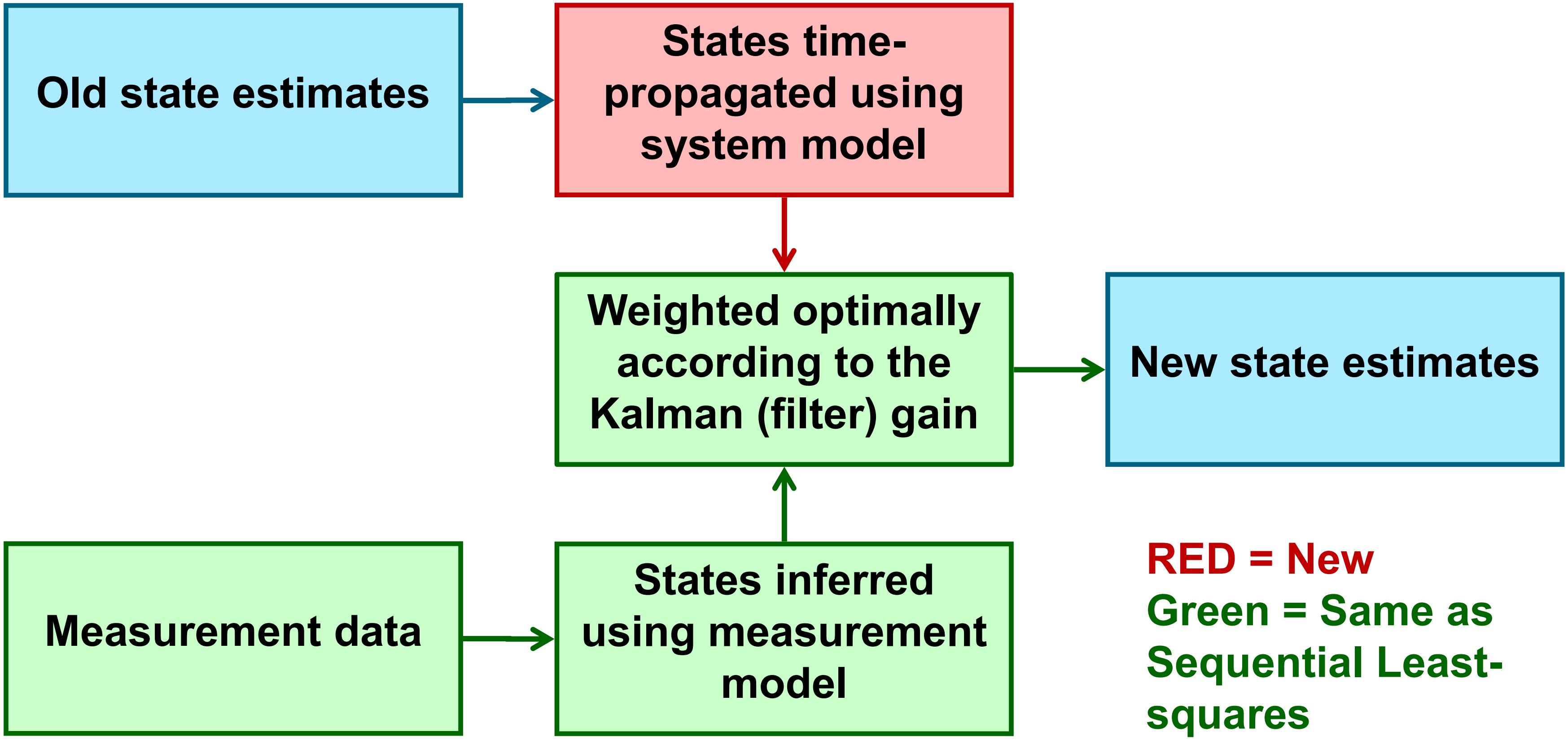
Kalman Filter Components: Algorithm (1)



RED = New
Green = As Sequential Least-squares
Blue = Changed

2. Introduction to the Kalman filter

Kalman Filter Components: Algorithm (2)



RED = New
Green = Same as Sequential Least-squares
Blue = Changed

2. Introduction to the Kalman filter

Kalman Filter Step by Step

State estimates & covariance from the previous epoch: $\hat{\mathbf{x}}_{k-1}^{+}, \mathbf{P}_{k-1}^{+}$

1) Use deterministic system model to calculate transition matrix, $\mathbf{\Phi}_{k-1}$

2) Use stochastic system model to calculate system noise covariance, $\mathbf{Q}_{k-1}$

3) Propagate state estimates $\hat{\mathbf{x}}_{k}^{-} = \mathbf{\Phi}_{k-1}\hat{\mathbf{x}}_{k-1}^{+}$

4) Calculate error covariance $\mathbf{P}_{k}^{-} = \mathbf{\Phi}_{k-1}\mathbf{P}_{k-1}^{+}\mathbf{\Phi}_{k-1}^{\mathrm{T}} + \mathbf{Q}_{k-1}$

5) Calculate the measurement matrix, $\mathbf{H}_{k}$

6) Calculate measurement noise covariance, $\mathbf{R}_{k}$

7) Calculate the Kalman gain $\mathbf{K}_{k} = \mathbf{P}_{k}^{-}\mathbf{H}_{k}^{\mathrm{T}}\left(\mathbf{R}_{k} + \mathbf{H}_{k}\mathbf{P}_{k}^{-}\mathbf{H}_{k}^{\mathrm{T}}\right)^{-1}$

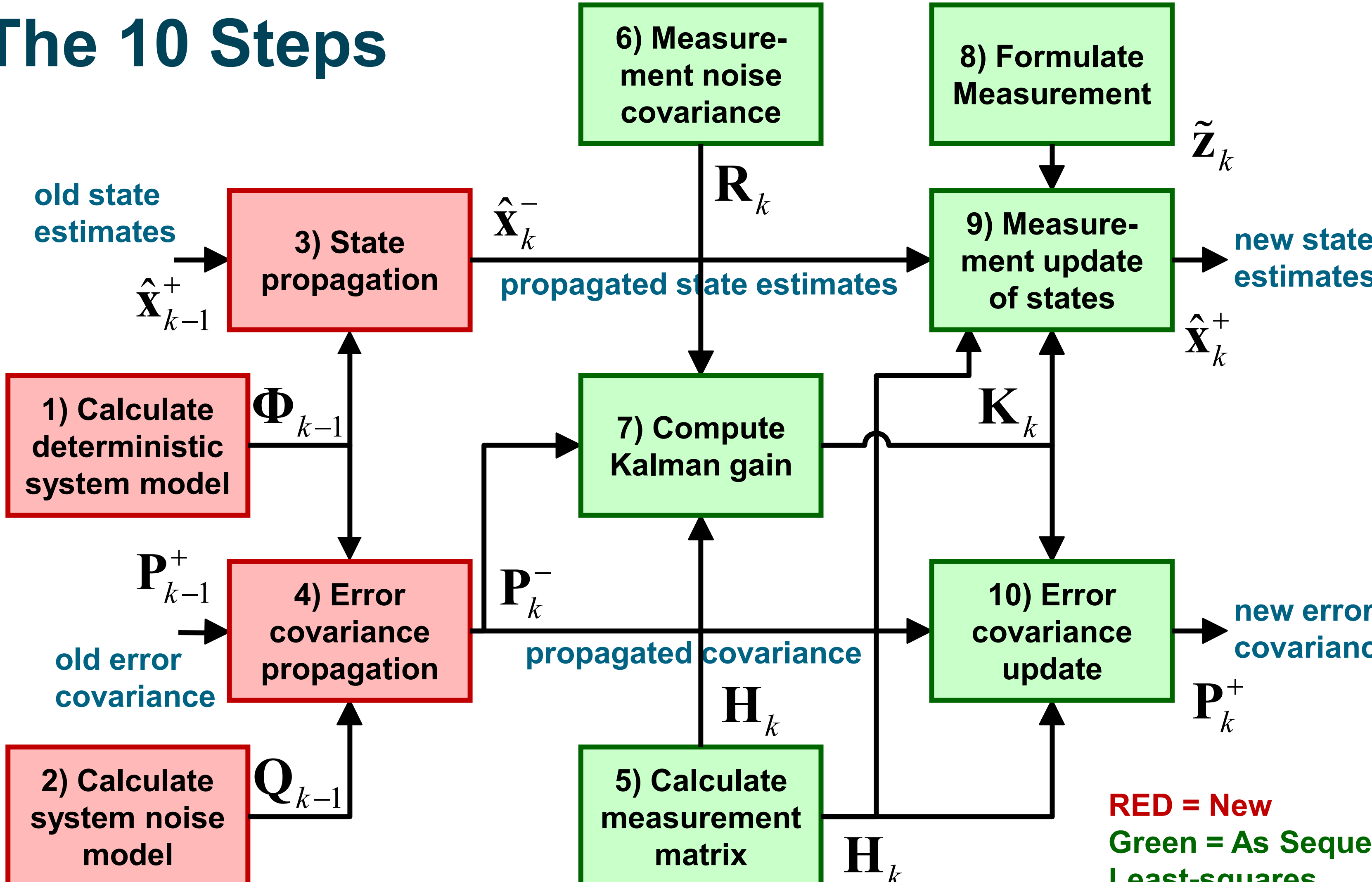
8) Formulate measurements $\tilde{\mathbf{z}}_{k}$

9) Measurement update of state estimates $\hat{\mathbf{x}}_{k}^{+} = \hat{\mathbf{x}}_{k}^{-} + \mathbf{K}_{k}\left(\tilde{\mathbf{z}}_{k} - \mathbf{H}_{k}\hat{\mathbf{x}}_{k}^{-}\right)$

10) Measurement update of error covariance $\mathbf{P}_{k}^{+} = \left(\mathbf{I} - \mathbf{K}_{k}\mathbf{H}_{k}\right)\mathbf{P}_{k}^{-}$

2. Introduction to the Kalman filter

The 10 Steps



RED = New

Green = As Sequential Least-squares

Contents

1. Sequential Least-Squares
2. Introduction to the Kalman Filter
3. Kalman Filter Examples
4. Properties and Extended Kalman filter
5. Filtered GNSS Positioning

3. Kalman filter Examples

Example 2: Position & Velocity in 1 Dimension

A train is an example of a one-dimensional positioning problem

The state vector is defined as

$$\mathbf{x}_k = \begin{pmatrix} r_k \\ v_k \end{pmatrix} \begin{matrix} \leftarrow \text{Position} \\ \leftarrow \text{Velocity} \end{matrix}$$

For this example…

- Initial position estimate is zero with an uncertainty of 1 m
- Initial velocity estimate is 2 m/s with an uncertainty of 0.5 m/s

The initial state estimate and its error covariance are thus

$$\hat{\mathbf{x}}_0^+ = \begin{pmatrix} 0 \\ 2 \end{pmatrix} \quad \mathbf{P}_0^+ = \begin{pmatrix} 1 & 0.1 \\ 0.1 & 0.25 \end{pmatrix}$$

See RVN KF Examples.xlsx on Moodle

Example 2 Step 1: Calculate Transition Matrix

For a two-state Kalman filter estimating 1D position and velocity:

$$\mathbf{x}_k = \begin{pmatrix} r_k \\ v_k \end{pmatrix}$$

The transition matrix is

Column 1: Old position

Column 2: Old velocity

Row 1: New position

Row 2: New velocity

$$\mathbf{\Phi}_{k-1} = \begin{pmatrix} 1 & \tau_s \\ 0 & 1 \end{pmatrix}$$

Change in position is integral of velocity

New state = old state

Velocity does not depend on position

Here, the interval between epochs, $\tau_s = 0.5\text{s}$

See RVN KF Examples.xlsx on Moodle

Example 2 Step 2: System Noise Covariance

The **system noise covariance matrix**, $\mathbf{Q}_{k\text{-}1}$, quantifies the increase in state uncertainties over time due to unknown noise, including motion.

For a two-state filter estimating position and velocity, the system noise is due to acceleration:

$$\mathbf{x}_k = \begin{pmatrix} r_k \\ v_k \end{pmatrix}$$

Short time intervals, τ_s<<*1*

$$\mathbf{Q}_{k-1} \approx \begin{pmatrix} 0 & 0 \\ 0 & S_a \tau_s \end{pmatrix}$$

Acceleration PSD

Longer time intervals

$$\mathbf{Q}_{k-1} = \begin{pmatrix} \tau_s^3/3 & \tau_s^2/2 \\ \tau_s^2/2 & \tau_s \end{pmatrix} S_a$$

Because position noise is the integral of velocity noise

Here, $S_a = 0.2$ and $\tau_s = 0.5$

For more on power spectral density (PSD) see Section B.4 of 'Notes on Statistical Measures and Probability' on Moodle (Session 6 tab)

See RVN KF Examples.xlsx on Moodle

3. Kalman filter Examples

Example 2 Step 3: Propagate State Estimates

Using the transition matrix to propagate the state vector estimate

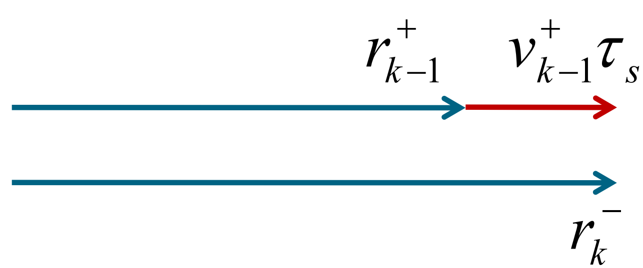
Predicted states at epoch *k*

$$\hat{\mathbf{x}}_k^- = \mathbf{\Phi}_{k-1}\hat{\mathbf{x}}_{k-1}^+$$

Estimated states at epoch *k*–1

$$\hat{\mathbf{x}}_0^+ = \begin{pmatrix} 0 \text{ m} \\ 2 \text{ m/s} \end{pmatrix} \quad \mathbf{\Phi}_0 = \begin{pmatrix} 1 & \tau_s \\ 0 & 1 \end{pmatrix} \quad \tau_s = 0.5 \text{ s} \quad \Rightarrow \quad \hat{\mathbf{x}}_1^- = \begin{pmatrix} 1 \text{ m} \\ 2 \text{ m/s} \end{pmatrix}$$

$$r_k^- = r_{k-1}^+ + v_{k-1}^+ \tau_s$$
$$v_k^- = v_{k-1}^+$$



See RVN KF Examples.xlsx on Moodle

Example 2 Step 4: Propagate Error Covariance

Using the transition matrix and system noise covariance matrix to propagate the state error covariance matrix:

Error covariance of predicted states at epoch *k*

Error covariance of state estimates at epoch *k*–1

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

Add system noise to account for unknown changes in states

Deterministic propagation of P matrix rows

Deterministic propagation of P matrix columns

$$\mathbf{P}_0^+ = \begin{pmatrix} 1 & 0.1 \\ 0.1 & 0.25 \end{pmatrix} \quad \mathbf{\Phi}_0 = \begin{pmatrix} 1 & 0.5 \\ 0 & 1 \end{pmatrix} \quad \mathbf{Q}_0 = \begin{pmatrix} 0.0083 & 0.025 \\ 0.025 & 0.1 \end{pmatrix} \quad \mathbf{P}_1^- = \begin{pmatrix} 1.71 & 0.25 \\ 0.25 & 0.35 \end{pmatrix}$$

Example 2 Step 5: Measurement Matrix

The **measurement matrix**, $\mathbf{H}_k$, defines how the measurement varies with the states

True states

Measurements $\longrightarrow \tilde{\mathbf{z}}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{w}_{m,k} \longleftarrow$ Measurement noise

Here, the measurements comprise position in one dimension

$$\tilde{z}_k = \tilde{r}_k = r_k + w_{m,k} \qquad \mathbf{x}_k = \begin{pmatrix} r_k \\ v_k \end{pmatrix} \qquad \Rightarrow \mathbf{H}_k = \begin{pmatrix} 1 & 0 \end{pmatrix}$$

Step 6: Measurement Noise Covariance

Here, the measurement noise standard deviation is 1.5 m $\therefore\ R_k = 2.25\ \mathrm{m}^2$

See RVN KF Examples.xlsx on Moodle

Step 7: The Kalman Gain Matrix

The **Kalman Gain matrix**, $\mathbf{K}_k$, is used to determine the weighting of the measurement vector in updating the state estimates

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

Matrix inversion

Qualitatively...

$$\text{Transformation from measurement domain to state domain} \times \frac{\text{State variance in measurement domain}}{\text{State variance in measurement domain} + \text{Measurement variance}}$$

3. Kalman filter Examples

Example 2 Step 7: Calculate Gain Matrix

Estimating position & velocity from position measurements

$$\mathbf{x}_k = \begin{pmatrix} r_k \\ v_k \end{pmatrix} \qquad \mathbf{P}_k^- = \begin{pmatrix} \sigma_r^2 & P_{rv} \\ P_{rv} & \sigma_v^2 \end{pmatrix}_k^- \qquad \mathbf{H}_k = \begin{pmatrix} 1 & 0 \end{pmatrix} \qquad \mathbf{R}_k = \sigma_{z,k}^2$$

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

$$\mathbf{P}_k^- \mathbf{H}_k^{\mathrm{T}} = \begin{pmatrix} \sigma_r^2 \\ P_{rv} \end{pmatrix}_k^- \qquad \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^{\mathrm{T}} = \left[\sigma_r^2 \right]_k^- \qquad \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k = \left[\sigma_r^2 \right]_k^- + \sigma_{z,k}^2$$

$$\therefore \mathbf{K}_k = \begin{pmatrix} \sigma_r^2 \\ P_{rv} \end{pmatrix}_k^- \left(\left[\sigma_r^2 \right]_k^- + \sigma_{z,k}^2 \right)^{-1} = \begin{pmatrix} \left[\sigma_r^2 \right]_k^- \Big/ \left(\left[\sigma_r^2 \right]_k^- + \sigma_{z,k}^2 \right) \\ \left[P_{rv} \right]_k^- \Big/ \left(\left[\sigma_r^2 \right]_k^- + \sigma_{z,k}^2 \right) \end{pmatrix} \qquad \mathbf{K}_1 = \begin{pmatrix} 0.342 \\ 0.073 \end{pmatrix}$$

(Cross) covariance of position and velocity errors determines effect of position measurement on velocity estimate

See RVN KF Examples.xlsx on Moodle

Example 2 Step 8: Formulate Measurement

Here, the measurements comprise position in one dimension

- At epoch 1, the measurement is 2m

Measurement innovation

Measurement vector

Measurement Innovation

$$\delta\mathbf{z}_k^- = \tilde{\mathbf{z}}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-$$

Measurement vector predicted from the predicted state vector

$$\hat{r}_1^- = 1\text{m}$$

$$\tilde{z}_1 = 2\text{m}$$

$$\delta z_1^- = 1\text{m}$$

See RVN KF Examples.xlsx on Moodle

3. Kalman filter Examples

Example 2 Step 9: Update State Estimates

Using the measurement innovation and Kalman gain matrix to update the state vector estimate

Estimated states

$$\boxed{\hat{\mathbf{x}}_k^+} = \boxed{\hat{\mathbf{x}}_k^-} + \boxed{\mathbf{K}_k} \left(\tilde{\mathbf{z}}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-\right) = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \delta\mathbf{z}_k^-$$

Measurement innovation

Predicted states

$$\hat{\mathbf{x}}_1^- = \begin{pmatrix} 1 \text{ m} \\ 2 \text{ m/s} \end{pmatrix}$$

Kalman Gain

$$\mathbf{K}_1 = \begin{pmatrix} 0.342 \\ 0.073 \end{pmatrix}$$

$$\hat{r}_1^- = 1\text{m}$$

$$\delta z_1^- = 1\text{m}$$

$$\hat{r}_1^+ = 1.34\text{m}$$

$$\hat{\mathbf{x}}_1^+ = \begin{pmatrix} 1.34 \text{ m} \\ 2.07 \text{ m/s} \end{pmatrix}$$

See RVN KF Examples.xlsx on Moodle

Example 2 Step 10: Update Error Covariance

Using the measurement matrix and Kalman gain matrix to update the error covariance

State estimates error covariance

$$\boxed{\mathbf{P}_k^+} = \left(\mathbf{I} - \boxed{\mathbf{K}_k}\boxed{\mathbf{H}_k}\right)\boxed{\mathbf{P}_k^-}$$

Predicted states error covariance

Kalman Gain

Measurement matrix

$$\mathbf{K}_1 = \begin{pmatrix} 0.342 \\ 0.073 \end{pmatrix} \qquad \mathbf{H}_k = \begin{pmatrix} 1 & 0 \end{pmatrix} \qquad \mathbf{P}_1^- = \begin{pmatrix} 1.71 & 0.25 \\ 0.25 & 0.35 \end{pmatrix}$$

$$\mathbf{P}_1^+ = \begin{pmatrix} 0.77 & 0.16 \\ 0.16 & 0.33 \end{pmatrix}$$

Updated error covariance is always smaller (or the same) as more information has been incorporated into the Kalman filter

See RVN KF Examples.xlsx on Moodle

3. Kalman filter Examples

Example 3: Position Only in 2 Dimensions

This could be used to track a pedestrian in a crowd

The state vector is defined as

$$\mathbf{x}_k = \begin{pmatrix} r_{x,k} \\ r_{y,k} \end{pmatrix} \begin{matrix} \leftarrow \text{Position in x direction} \\ \leftarrow \text{Position in y direction} \end{matrix}$$

For this example, the initial state estimate and its error covariance are

$$\hat{\mathbf{x}}_0^+ = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ m} \quad \mathbf{P}_0^+ = \begin{pmatrix} 0.25 & 0.1 \\ 0.1 & 0.25 \end{pmatrix} \text{ m}^2$$

See RVN KF Examples.xlsx on Moodle

Example 3: State Propagation

Step 1: Calculate Transition Matrix

Change in *x* and *y* position between epochs is unknown $\mathbf{\Phi}_{k-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

Step 2: Calculate System Noise Covariance Matrix

Velocity PSD (x direction), $S_{vx} = 1.8$

Velocity PSD (y direction), $S_{vy} = 2.2$

Interval between epochs, $\tau_s = 0.5\text{s}$

$$\mathbf{Q}_{k-1} = \begin{pmatrix} S_{vx}\tau_s & 0 \\ 0 & S_{vy}\tau_s \end{pmatrix} = \begin{pmatrix} 0.9 & 0 \\ 0 & 1.1 \end{pmatrix}$$

Step 3: Propagate State Estimates

$$\hat{\mathbf{x}}_k^- = \mathbf{\Phi}_{k-1}\,\hat{\mathbf{x}}_{k-1}^+$$

$$\hat{\mathbf{x}}_1^- = \begin{pmatrix} 1 \\ 0 \end{pmatrix}\text{m} \qquad \mathbf{P}_1^- = \begin{pmatrix} 1.15 & 0.1 \\ 0.1 & 1.35 \end{pmatrix}\text{m}^2$$

Step 4: Error Covariance

$$\mathbf{P}_k^- = \mathbf{\Phi}_{k-1}\,\mathbf{P}_{k-1}^+\,\mathbf{\Phi}_{k-1}^{\mathrm{T}} + \mathbf{Q}_{k-1}$$

See RVN KF Examples.xlsx on Moodle

Example 3: Measurement Model

Step 5: Measurement Matrix

Here, the measurements comprise position in two dimensions

$$\tilde{\mathbf{z}}_k = \begin{pmatrix} \tilde{r}_{x,k} \\ \tilde{r}_{y,k} \end{pmatrix} = \begin{pmatrix} r_{x,k} + w_{mx,k} \\ r_{y,k} + w_{my,k} \end{pmatrix} \qquad \mathbf{x}_k = \begin{pmatrix} r_{x,k} \\ r_{y,k} \end{pmatrix} \qquad \Rightarrow \mathbf{H}_k = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Step 6: Measurement Noise Covariance

Here, the measurement noise covariance is:

$$\mathbf{R}_k = \begin{pmatrix} 1 & 0.1 \\ 0.1 & 1 \end{pmatrix} \mathrm{m}^2$$

See RVN KF Examples.xlsx on Moodle

3. Kalman filter Examples

Example 3: Measurement Update

Step 7: Calculate Kalman Gain Matrix

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

Step 8: Formulate Measurement Vector

The measurements at epoch 1 are

$$\tilde{\mathbf{z}}_1 = \begin{pmatrix} \tilde{r}_{x,1} \\ \tilde{r}_{y,1} \end{pmatrix} = \begin{pmatrix} 2 \\ -2 \end{pmatrix} \text{ m}$$

Step 9: Update State Estimates

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

$$\hat{\mathbf{x}}_1^+ = \begin{pmatrix} 1.54 \\ -1.16 \end{pmatrix} \text{ m}$$

Step 10: Update Error Covariance

$$\mathbf{P}_k^+ = \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k \right) \mathbf{P}_k^-$$

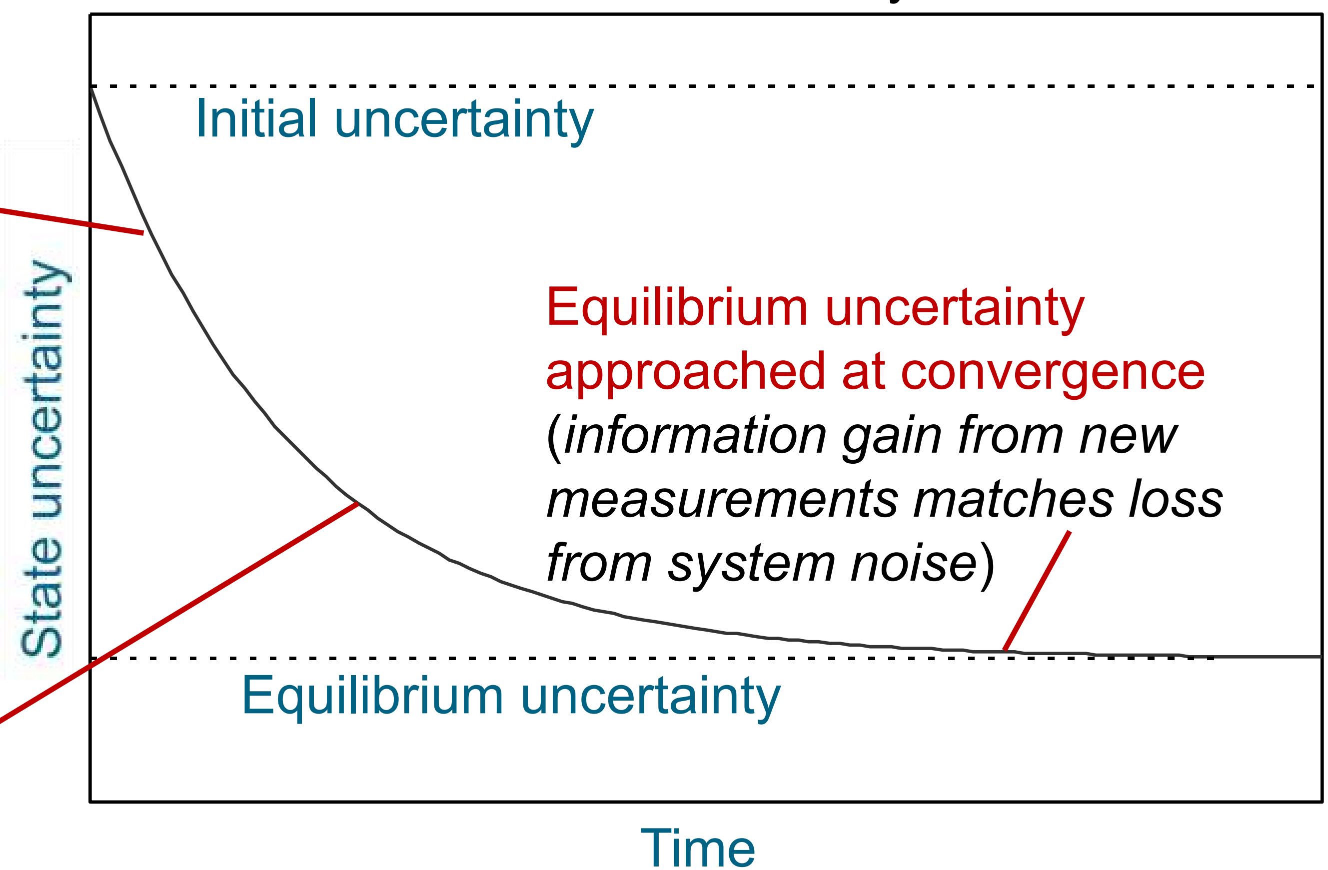
$$\mathbf{P}_1^+ = \begin{pmatrix} 0.53 & 0.05 \\ 0.05 & 0.57 \end{pmatrix} \text{ m}^2$$

See RVN KF Examples.xlsx on Moodle

Contents

Filter Convergence

Variation of state uncertainty with time



Initially, KF uncertainties drop rapidly

Gain drops as estimates improve

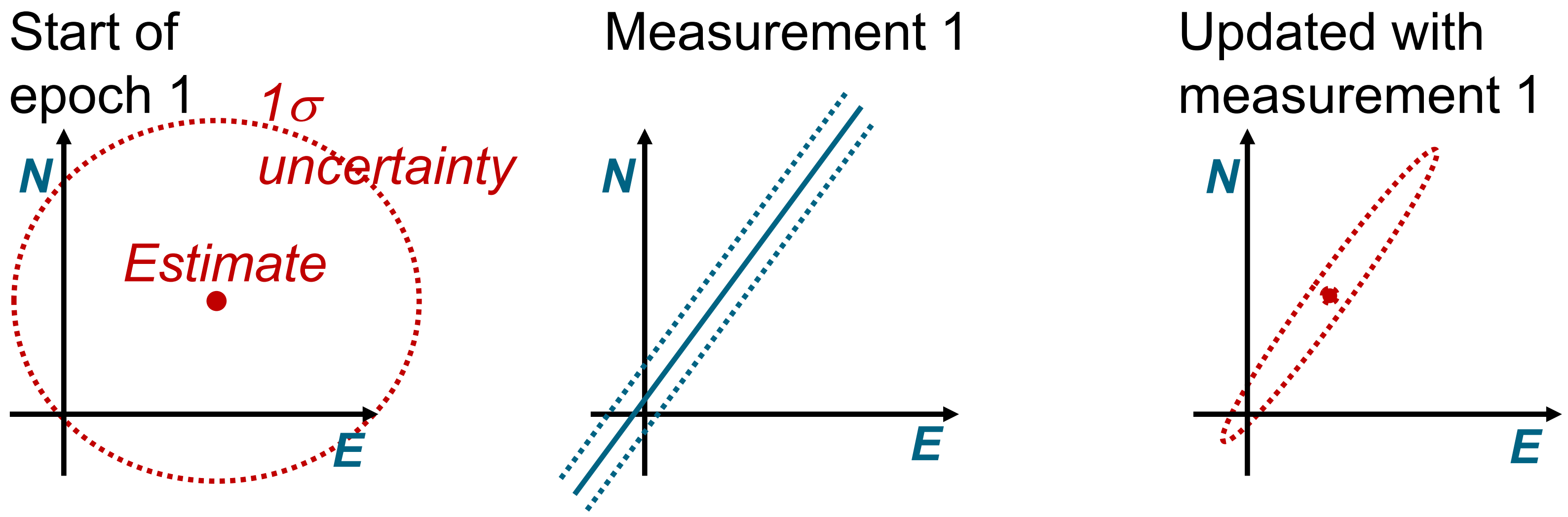
Incomplete measurements (1)

Consider a Kalman filter estimating 2D position: *E* and *N*

Measurement 1 at time epoch 1: $z_1 = aE + bN$

(*a* & *b* are known constants)

E and *N* can not be uniquely estimated using data from this epoch



The KF knows the correlation between the *E* and *N* position errors

Incomplete measurements (2)

Consider a Kalman filter estimating 2D position: *E* and *N*

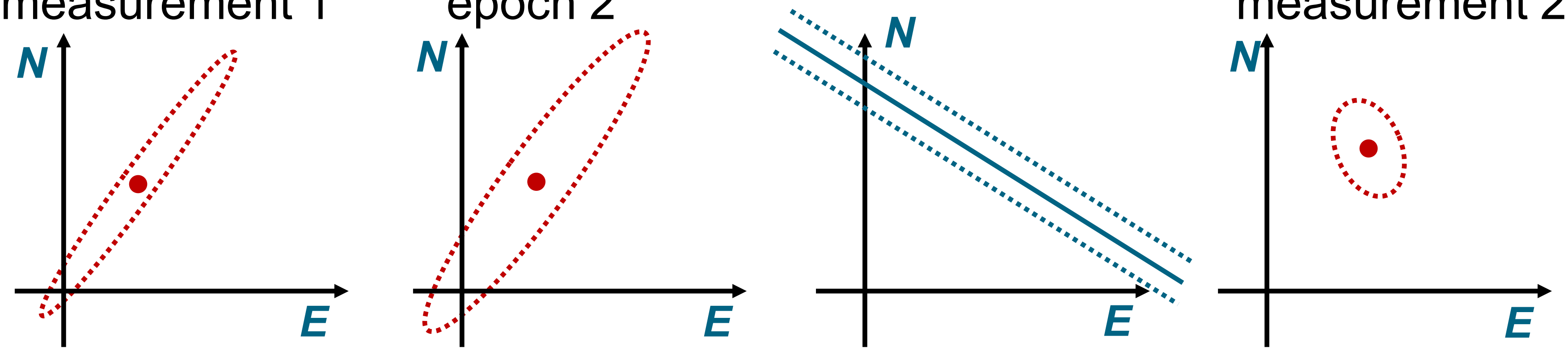
Measurement 2 at time epoch 2: $z_2 = cE + dN$

(*c* & *d* are known constants)

E and *N* can not be uniquely estimated using data from epoch 2 either

They can be estimated using data from both epochs

Updated with measurement 1 | Start of epoch 2 | Measurement 2 | Updated with measurement 2



Kalman Filter Assumptions

- The measurements are a linear function of the states
  i.e., the measurement matrix, $\mathbf{H}$, is not a function of the states, $\mathbf{x}$
- The system model is linear
  i.e., the transition matrix, $\mathbf{\Phi}$, is not a function of the states, $\mathbf{x}$
- All unknowns have Gaussian distributions
- System and measurement noise is not time-correlated
- System and measurement noise covariances are known

Real systems do not obey these rules

4. Properties and Extended Kalman Filter

Real Systems and the Kalman Filter

A Kalman filter assumes systems are linear, Gaussian, and have white noise

Real systems can be nonlinear nonGaussian, and have time-correlated noise

Small deviations from these assumptions can be handled by assuming a larger system noise, $\mathbf{Q}$, and/or measurement noise, $\mathbf{R}$

For larger deviations, modifications to the Kalman filter are required, e.g.

- Extended Kalman Filter
- Unscented Kalman Filter
- Kalman Smoother

4. Properties and Extended Kalman Filter

Nonlinear Measurement Model

- In general, measurements are not linear functions of the states:

Measurement function (nonlinear)

Measurement noise

$$\tilde{\mathbf{z}}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{w}_{m,k}$$

Measurement vector

True state vector

Predicted state vector

Measurement innovation

$$\delta\mathbf{z}_k^- = \tilde{\mathbf{z}}_k - \mathbf{h}_k\left(\hat{\mathbf{x}}_k^-\right)$$

- *Example*: Ranging measurements are not a linear function of the user equipment position
- A standard Kalman filter can not be used for these systems
- But, an **Extended Kalman filter** may be used if the measurement innovation, $\delta\mathbf{z}_k^-$, is a linear function of $\delta\mathbf{x}_k = \hat{\mathbf{x}}_k^+ - \hat{\mathbf{x}}_k^-$

Extended Kalman Filter

An **Extended Kalman filter** may be used with a non-linear measurement model if the measurement innovation can be approximated as a linear function of the state vector innovation

$$\delta \mathbf{z}_k^- \approx \mathbf{H}_k \delta \mathbf{x}_k + \delta \mathbf{z}_k^+$$

Measurement innovation

Measurement matrix

State vector innovation $= \hat{\mathbf{x}}_k^+ - \hat{\mathbf{x}}_k^-$

Measurement residual $= -\mathbf{v}_k$

Then

- Measurement matrix (Step 5) is $\mathbf{H}_k = \left.\dfrac{\partial \mathbf{h}_k(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k^-} = \left.\dfrac{\partial \mathbf{z}_k(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k^-}$
- Measurement update (Step 9) is
$$\begin{aligned}\hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left[\tilde{\mathbf{z}}_k - \mathbf{h}_k\left(\hat{\mathbf{x}}_k^-\right)\right] \\ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k \delta \mathbf{z}_k^-\end{aligned}$$
- *Other steps are the same unless a non-linear system model is used*

Extended Kalman Filter Step by Step

State estimates & covariance from the previous epoch: $\hat{\mathbf{x}}_{k-1}^{+}, \mathbf{P}_{k-1}^{+}$

1) Use deterministic system model to calculate transition matrix, $\mathbf{\Phi}_k$

2) Use stochastic system model to calculate system noise covariance, $\mathbf{Q}_k$

3) Propagate state estimates $\hat{\mathbf{x}}_k^- = \mathbf{\Phi}_{k-1}\hat{\mathbf{x}}_{k-1}^+$ or $\hat{\mathbf{x}}_k^- = \mathbf{f}_{k-1}\left(\hat{\mathbf{x}}_{k-1}^+\right)$

4) Calculate error covariance $\mathbf{P}_k^- = \mathbf{\Phi}_{k-1}\mathbf{P}_{k-1}^+\mathbf{\Phi}_{k-1}^{\mathrm{T}} + \mathbf{Q}_{k-1}$ $\quad \mathbf{\Phi}_{k-1} = \left.\dfrac{\partial \mathbf{f}_{k-1}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1}^+}$

5) Calculate the measurement matrix $\mathbf{H}_k = \partial \mathbf{h}_k / \partial \mathbf{x}\big|_{\mathbf{x}=\hat{\mathbf{x}}_k^-}$

6) Calculate measurement noise covariance, $\mathbf{R}_k$

7) Calculate the Kalman gain $\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^{\mathrm{T}}\left(\mathbf{R}_k + \mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^{\mathrm{T}}\right)^{-1}$

8) Formulate measurements $\tilde{\mathbf{z}}_k$

9) Measurement update of state estimates $\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left[\tilde{\mathbf{z}}_k - \mathbf{h}_k\left(\hat{\mathbf{x}}_k^-\right)\right]$

10) Measurement update of error covariance $\mathbf{P}_k^+ = \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{P}_k^-$

4. Properties and Extended Kalman Filter

Extended Kalman Filter & Nonlinear Least Squares

- An **Extended Kalman filter** is equivalent to nonlinear least squares
- The measurement matrix is a function of the predicted states, $\hat{\mathbf{x}}_k^-$

Extended Kalman Filter
$$\delta\mathbf{z}_k^- = \tilde{\mathbf{z}}_k - \mathbf{h}\left(\hat{\mathbf{x}}_k^-\right) = \mathbf{H}\left(\hat{\mathbf{x}}_k^-\right)\left(\hat{\mathbf{x}}_k^+ - \hat{\mathbf{x}}_k^-\right) + \delta\mathbf{z}_k^+$$

Nonlinear Least Squares
$$\mathbf{b} = \tilde{\mathbf{z}} - \mathbf{h}\left(\hat{\mathbf{x}}^-\right) = \mathbf{H}\left(\hat{\mathbf{x}}^-\right)\left(\hat{\mathbf{x}}^+ - \hat{\mathbf{x}}^-\right) - \mathbf{v}$$

Measurement innovation

Measurement matrix

Updated – predicted

Meas. residual

- *But there is no iteration at a given epoch*
- *Instead, it is assumed that linearisation errors will be reduced over successive epochs, each time new measurements are incorporated*

Contents

5. Filtered GNSS Positioning

Why use a Kalman Filter-Based Approach?

- A single-epoch GNSS navigation solution discards useful information from previous measurements:
  - The previous position and velocity solution provides a good indication of the current position and velocity
  - The previous clock offset and drift solution provides a good indication of the current clock offset and drift
- This information can be used to smooth out noise in GNSS pseudo-range and range-rate (Doppler) measurements
- A filtered navigation solution should always be more accurate than a single-epoch solution
- A (degraded) position solution can be maintained (in the short term) using signals from fewer than 4 GNSS satellites

5. Filtered GNSS Positioning

Clock and Height Coasting

With a Kalman filter-based positioning algorithm…

- Clock offset can be predicted ahead for a few seconds using the clock drift solution
- Height normally changes more slowly than north and east position
- Short periods of 2- or 3-satellite reception can be bridged by assuming constant height and/or clock drift
- A Kalman filter will do this automatically
- Accuracy will degrade until 4-satellite reception returns
  - *Depending on* true height variation
  - *And* receiver oscillator quality

Implementing a GNSS Navigation Filter

Which estimation algorithm?

- A *standard* Kalman filter may be used for the state propagation
- An *extended* Kalman filter *must* be used for the measurement update *because this is a nonlinear*

Which coordinate frames?

- Earth-centred inertial (ECI) is the simplest – *no Sagnac effect*
- Cartesian Earth-centred Earth-fixed (ECEF) is the most common – *It will be used here*
- Latitude, longitude and height with ECEF-referenced velocity resolved about north, east, down – *avoids output conversion*

All three are included in *Principles of GNSS, Inertial and Multisensor Integrated Navigation Systems,* linked to on Moodle

State Selection

Optimum state selection depends on the application

- Receiver clock offset and drift must *always* be estimated
- The choice of kinematic states varies:
  - For static applications, e.g. surveying, only position is needed
  - For low-dynamics applications, e.g. most land and sea navigation, position *and* velocity must be estimated
  - For high-dynamics applications, e.g. air navigation, acceleration must be added, though INS/GNSS is typically used
- Other states may also be estimated (*see Principles of GNSS, Inertial and Multisensor Integrated Navigation Systems*)

5. Filtered GNSS Positioning

State Vector

8 states are estimated in this example:

This is a total-state Kalman filter

The state estimates must be initialised using a single-epoch GNSS position, velocity and clock solution

Initial error covariance must reflect the accuracy of the state vector initialisation

e = ECEF frame

a = user antenna body frame

$$\mathbf{x} = \begin{pmatrix} x^e_{ea} \\ y^e_{ea} \\ z^e_{ea} \\ v^e_{ea,x} \\ v^e_{ea,y} \\ v^e_{ea,z} \\ \delta\rho^a_c \\ \delta\dot{\rho}^a_c \end{pmatrix}$$

$\mathbf{r}^e_{ea}$ Antenna position

$\mathbf{v}^e_{ea}$ Antenna velocity

Receiver clock offset

Receiver clock drift

5. Filtered GNSS Positioning

Deterministic System Model

$\mathbf{r}^e_{ea}(t-\tau_s)$

Position at start of propagation interval

Change in position

$\mathbf{v}^e_{ea}\tau_s$

$\mathbf{r}^e_{ea}(t)$

Position at end of propagation interval

Reference frame origin

Thus

$$\mathbf{r}^e_{ea}(t) = \mathbf{r}^e_{ea}(t-\tau_s) + \mathbf{v}^e_{ea}\tau_s$$

Similarly for the receiver clock:

$$\delta\rho^a_c(t) = \delta\rho^a_c(t-\tau_s) + \delta\dot{\rho}^a_c\tau_s$$

5. Filtered GNSS Positioning

Step 1: Calculate Transition Matrix (1)

The transition matrix relates the states at the current time, t, to their values at the previous time, $t - \tau_s$

$$\mathbf{\Phi}_{k-1} = \left(\begin{array}{ccc|ccc|c|c} 1 & 0 & 0 & \tau_s & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \tau_s & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \tau_s & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 & \tau_s \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right) \qquad \mathbf{x} = \begin{pmatrix} x_{ea}^e \\ y_{ea}^e \\ z_{ea}^e \\ v_{ea,x}^e \\ v_{ea,y}^e \\ v_{ea,z}^e \\ \delta\rho_c^a \\ \delta\dot{\rho}_c^a \end{pmatrix}$$

5. Filtered GNSS Positioning

Step 1: Calculate Transition Matrix (2)

It is convenient to express the KF matrices as arrays of sub matrices corresponding to the vector sub-components

$$
\begin{array}{rc}
 & \begin{array}{cccc} \text{Old position} & \text{Old velocity} & \text{Old clock offset} & \text{Old clock drift} \end{array} \\
\begin{array}{r} \text{New position} \\ \text{New velocity} \\ \text{New clock offset} \\ \text{New clock drift} \end{array} & \mathbf{\Phi}_{k-1} = \begin{pmatrix} \mathbf{I}_3 & \tau_s \mathbf{I}_3 & \mathbf{0}_{3,1} & \mathbf{0}_{3,1} \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_{3,1} & \mathbf{0}_{3,1} \\ \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & 1 & \tau_s \\ \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & 0 & 1 \end{pmatrix}
\end{array}
$$

τ_s = time interval

5. Filtered GNSS Positioning

Step 2: System Noise Covariance Matrix

System noise represents the unknown changes in the states over time

Here, it comprises two main components:

- Random walk of the velocity due to acceleration
- Random walk of the receiver clock drift

These are also integrated onto the position and clock offset through the deterministic system model

$$\mathbf{Q}_{k-1} = \begin{pmatrix} \frac{1}{3} S_a \tau_s^3 \mathbf{I}_3 & \frac{1}{2} S_a \tau_s^2 \mathbf{I}_3 & \mathbf{0}_{3,1} & \mathbf{0}_{3,1} \\ \frac{1}{2} S_a \tau_s^2 \mathbf{I}_3 & S_a \tau_s \mathbf{I}_3 & \mathbf{0}_{3,1} & \mathbf{0}_{3,1} \\ \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & S_{c\phi}^a \tau_s + \frac{1}{3} S_{cf}^a \tau_s^3 & \frac{1}{2} S_{cf}^a \tau_s^2 \\ \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & \frac{1}{2} S_{cf}^a \tau_s^2 & S_{cf}^a \tau_s \end{pmatrix}$$

Acceleration PSD Clock phase drift PSD Clock frequency drift PSD

5. Filtered GNSS Positioning

Steps 3 & 4: Propagate State & Covariance

Propagate state vector estimate:

$$\hat{\mathbf{x}}_k^- = \mathbf{\Phi}_{k-1}\hat{\mathbf{x}}_{k-1}^+$$

Known changes over time

Propagate state estimation error covariance matrix:

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

Unknown changes over time

Using

$$\mathbf{\Phi}_{k-1} = \begin{pmatrix} \mathbf{I}_3 & \tau_s\mathbf{I}_3 & \mathbf{0}_{3,1} & \mathbf{0}_{3,1} \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_{3,1} & \mathbf{0}_{3,1} \\ \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & 1 & \tau_s \\ \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & 0 & 1 \end{pmatrix} \qquad \mathbf{Q}_{k-1} = \begin{pmatrix} \frac{1}{3}S_a\tau_s^3\mathbf{I}_3 & \frac{1}{2}S_a\tau_s^2\mathbf{I}_3 & \mathbf{0}_{3,1} & \mathbf{0}_{3,1} \\ \frac{1}{2}S_a\tau_s^2\mathbf{I}_3 & S_a\tau_s\mathbf{I}_3 & \mathbf{0}_{3,1} & \mathbf{0}_{3,1} \\ \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & S_{c\phi}^a\tau_s + \frac{1}{3}S_{cf}^a\tau_s^3 & \frac{1}{2}S_{cf}^a\tau_s^2 \\ \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & \frac{1}{2}S_{cf}^a\tau_s^2 & S_{cf}^a\tau_s \end{pmatrix}$$

5. Filtered GNSS Positioning

Measurements Used

Pseudo-range

- Range (m) from satellite to user antenna plus clock offset ρ_a^s
- Obtained from code tracking

Pseudo-range rate *or* Doppler shift

- Range rate (m/s) from satellite to user antenna + clock drift $\dot{\rho}_a^s$
- Doppler shift (Hz) = – pseudo-range rate * carrier frequency / speed of light $\Delta f_{ca,a}^s$
- These measurements are obtained from carrier tracking
  - Much smaller errors from noise and multipath interference
  - Improves accuracy of position and velocity solution

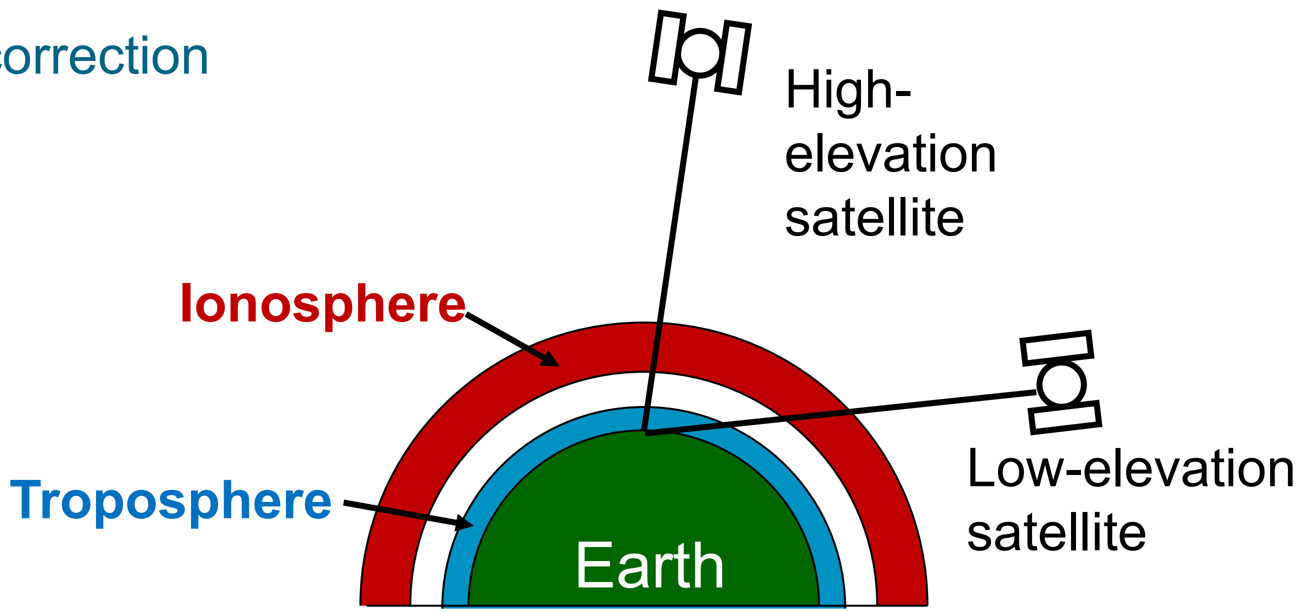
5. Filtered GNSS Positioning

Measurement Correction

Several corrections are applied to the raw GNSS receiver measurements before using them to compute position:

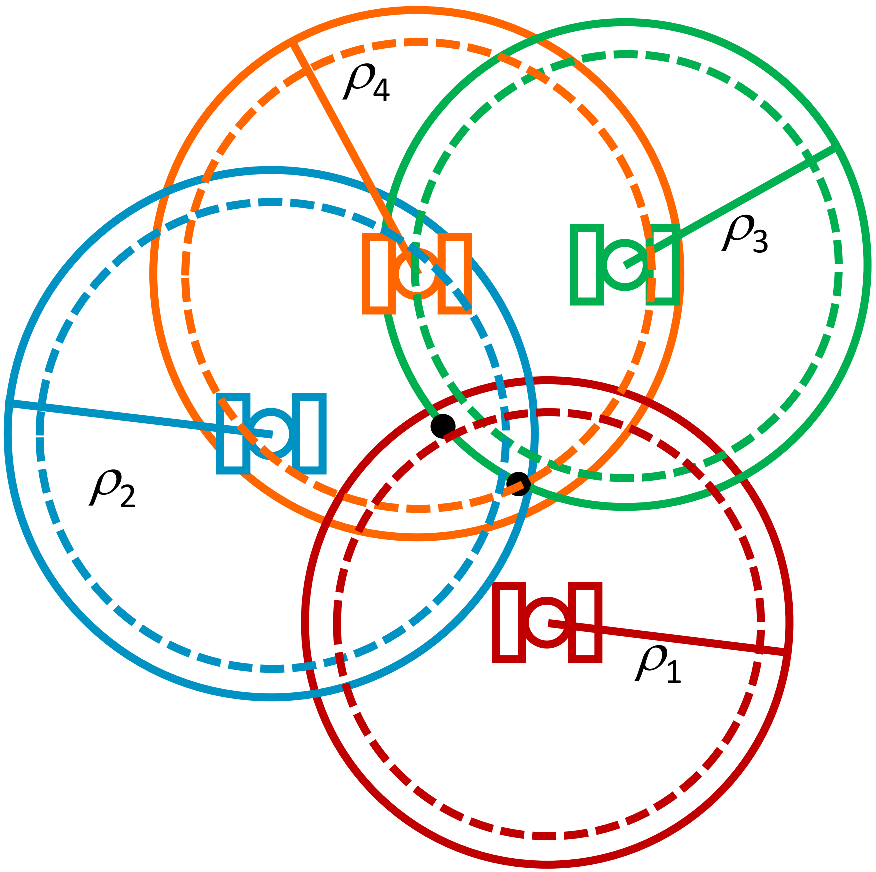
- Satellite clock correction
- Ionosphere correction
- Troposphere correction

See Lecture 2A



5. Filtered GNSS Positioning

Positioning geometry in 3 dimensions



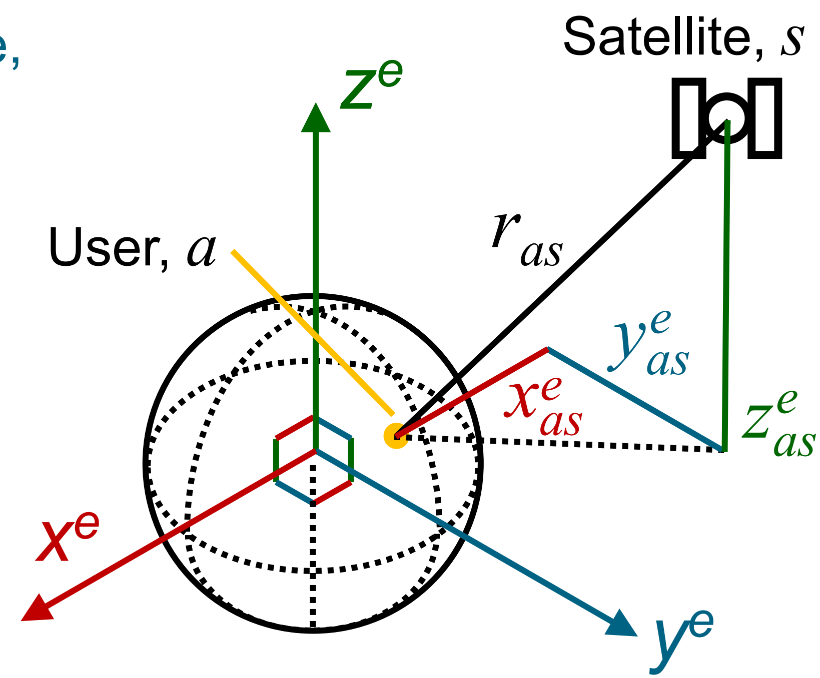
With GNSS *pseudo-*ranges, you need a 4th satellite to resolve the receiver clock error

5. Filtered GNSS Positioning

Ranging Geometry

Applying Pythagorus' Theorem Twice,
The user – satellite distance is

$$r_{as}^{2} = x_{as}^{e\ 2} + y_{as}^{e\ 2} + z_{as}^{e\ 2} = \mathbf{r}_{as}^{e\ \mathrm{T}} \mathbf{r}_{as}^{e}$$



(User to satellite) =
(Earth to satellite) – (Earth to User)

$$\Rightarrow \mathbf{r}_{as}^{e} = \mathbf{r}_{es}^{e} - \mathbf{r}_{ea}^{e} \qquad y_{as}^{e} = y_{es}^{e} - y_{ea}^{e}$$

$$x_{as}^{e} = x_{es}^{e} - x_{ea}^{e} \qquad z_{as}^{e} = z_{es}^{e} - z_{ea}^{e}$$

$$\therefore\ r_{as}^{2} = \left(x_{es}^{e} - x_{ea}^{e}\right)^{2} + \left(y_{es}^{e} - y_{ea}^{e}\right)^{2} + \left(z_{es}^{e} - z_{ea}^{e}\right)^{2}$$

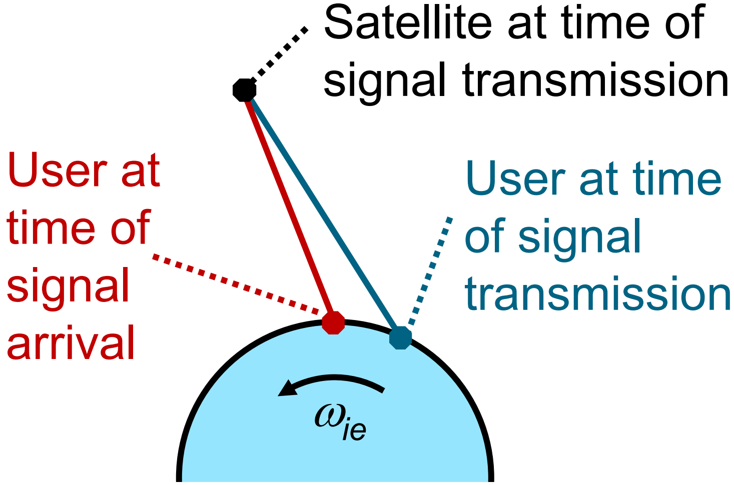
Pseudo-range = range + receiver clock offset
(*where satellite clock offset is corrected*) $\Rightarrow$

$$\rho_{a,C}^{s} = r_{as} + \delta\rho_{c}^{a}$$

Sagnac Effect

A change in the apparent speed of light due to coordinate frame rotation with respect to inertial space:

- Applies to Earth-referenced frames
- Ignoring it results in position errors of up to 40m



A Sagnac correction is typically applied to the satellite positions:

- We replace $\mathbf{r}^e_{es}$ with $\mathbf{r}^I_{Is} = \mathbf{C}^I_e \mathbf{r}^e_{es}$

where

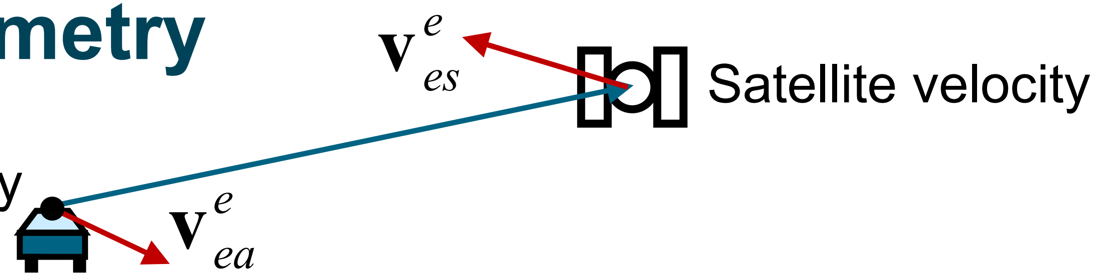
$$\mathbf{C}^I_e \approx \begin{pmatrix} 1 & \omega_{ie} r_{as}/c & 0 \\ -\omega_{ie} r_{as}/c & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

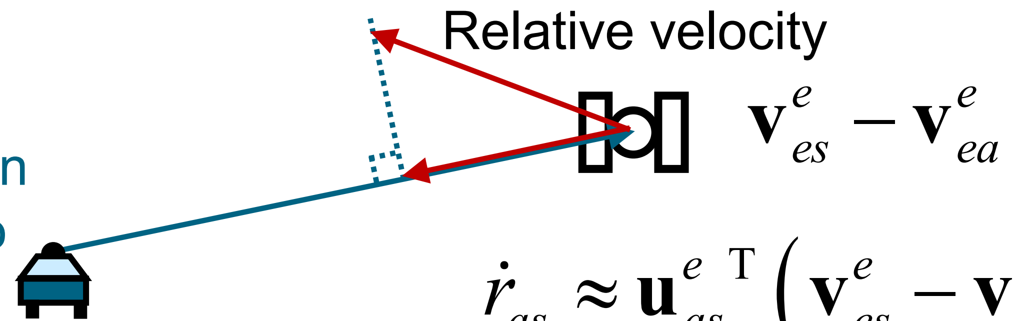
so

$$r_{as} = \sqrt{\left[\mathbf{r}^I_{Is} - \mathbf{r}^e_{ea}\right]^{\mathrm{T}} \left[\mathbf{r}^I_{Is} - \mathbf{r}^e_{ea}\right]}$$

ω_{ie} is Earth rotation rate and c is the speed of light

Range Rate Geometry





Range rate is the projection of the relative velocity onto the line of sight:

$$\dot{r}_{as} \approx \mathbf{u}_{as}^{e\,\mathrm{T}} \left(\mathbf{v}_{es}^{e} - \mathbf{v}_{ea}^{e} \right)$$

Accounting for the Sagnac effect…

$$\dot{r}_{as} = \hat{\mathbf{u}}_{as}^{e\,\mathrm{T}} \left[\mathbf{C}_{e}^{I} \left(\mathbf{v}_{es}^{e} + \mathbf{\Omega}_{ie}^{e} \mathbf{r}_{es}^{e} \right) - \left(\mathbf{v}_{ea}^{e} + \mathbf{\Omega}_{ie}^{e} \mathbf{r}_{ea}^{e} \right) \right]$$

Pseudo-range rate = range rate + receiver clock drift (*where satellite clock offset & drift are corrected*) :

$$\dot{\rho}_{a,C}^{s} = \dot{r}_{as} + \delta\dot{\rho}_{c}^{a}$$

$$\mathbf{\Omega}_{ie}^{e} = \begin{pmatrix} 0 & -\omega_{ie} & 0 \\ \omega_{ie} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

5. Filtered GNSS Positioning

Measurement Vector

Pseudo-ranges — Pseudo-range rates

$$\mathbf{z}_k = \left(\tilde{\rho}^{1}_{a,C},\quad \tilde{\rho}^{2}_{a,C},\quad \cdots \quad \tilde{\rho}^{m}_{a,C},\ \vdots\ \tilde{\dot{\rho}}^{1}_{a,C},\quad \tilde{\dot{\rho}}^{2}_{a,C},\quad \cdots \quad \tilde{\dot{\rho}}^{m}_{a,C}\right)^{\mathrm{T}}_{k}$$

1st satellite

*m*th satellite

C denotes corrected for satellite clock, ionosphere and troposphere errors

5. Filtered GNSS Positioning

Measurement Model

The predicted pseudo-ranges and pseudo-range rates are

Pseudo-ranges | Pseudo-range rates

$$\mathbf{h}\left(\hat{\mathbf{x}}_k^-\right) = \left(\hat{\rho}_{a,C}^{1-}, \quad \hat{\rho}_{a,C}^{2-}, \quad \cdots \quad \hat{\rho}_{a,C}^{m-}, \;\vdots\; \hat{\dot{\rho}}_{a,C}^{1-}, \quad \hat{\dot{\rho}}_{a,C}^{2-}, \quad \cdots \quad \hat{\dot{\rho}}_{a,C}^{m-}\right)_k^{\mathrm{T}}$$

1st satellite

mth satellite

Corrected for satellite clock, ionosphere and troposphere errors

$$\hat{\rho}_{a,C,k}^{s-} = \sqrt{\left[\mathbf{C}_e^I \hat{\mathbf{r}}_{es}^e - \hat{\mathbf{r}}_{ea,k}^{e-}\right]^{\mathrm{T}} \left[\mathbf{C}_e^I \hat{\mathbf{r}}_{es}^e - \hat{\mathbf{r}}_{ea,k}^{e-}\right]} + \delta\hat{\rho}_{c,k}^{a-}$$

$$\hat{\dot{\rho}}_{a,C,k}^{s-} = \hat{\mathbf{u}}_{as,k}^{e-\,\mathrm{T}} \left[\mathbf{C}_e^I \left(\hat{\mathbf{v}}_{es}^e + \mathbf{\Omega}_{ie}^e \hat{\mathbf{r}}_{es}^e\right) - \left(\hat{\mathbf{v}}_{ea}^{e-} + \mathbf{\Omega}_{ie}^e \hat{\mathbf{r}}_{ea,k}^{e-}\right)\right] + \delta\hat{\dot{\rho}}_{c,k}^{a-}$$

Line-of-sight

5. Filtered GNSS Positioning

Step 5: Calculate Measurement Matrix

$$\mathbf{H}_k = \left.\frac{\partial \mathbf{h}(\mathbf{x}, t_k)}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k^-}$$

Line-of-sight unit vector

$$\mathbf{u}_{as,k}^e \approx \frac{\hat{\mathbf{r}}_{es}^e(\tilde{t}_{st,a,k}^s) - \hat{\mathbf{r}}_{ea,k}^{e-}}{\left|\hat{\mathbf{r}}_{es}^e(\tilde{t}_{st,a,k}^s) - \hat{\mathbf{r}}_{ea,k}^{e-}\right|}$$

$$\mathbf{H}_k \approx \left.\begin{pmatrix} -u_{a1,x}^e & -u_{a1,y}^e & -u_{a1,z}^e & 0 & 0 & 0 & 1 & 0 \\ -u_{a2,x}^e & -u_{a2,y}^e & -u_{a2,z}^e & 0 & 0 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -u_{am,x}^e & -u_{am,y}^e & -u_{am,z}^e & 0 & 0 & 0 & 1 & 0 \\ \hdashline 0 & 0 & 0 & -u_{a1,x}^e & -u_{as,1,y}^e & -u_{a1,z}^e & 0 & 1 \\ 0 & 0 & 0 & -u_{a2,x}^e & -u_{a2,y}^e & -u_{a2,z}^e & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & -u_{am,x}^e & -u_{am,y}^e & -u_{am,z}^e & 0 & 1 \end{pmatrix}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k^-}$$

As least-squares $\mathbf{H}$ matrix

5. Filtered GNSS Positioning

Step 6: Measurement Noise Covariance

Variance of 1st satellite pseudo-range

$$\mathbf{R}_k = \begin{pmatrix} \sigma^2_{\rho 1,k} & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & \sigma^2_{\rho 2,k} & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma^2_{\rho m,k} & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \sigma^2_{r1,k} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \sigma^2_{r2,k} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & \sigma^2_{rm,k} \end{pmatrix}$$

Variance of m^{th} satellite pseudo-range rate

R matrix commonly modelled as diagonal and constant:

Accounts for noise-like errors only, not biases

- Benefit in modelling **R** as a function of
  - Signal to noise level of each signal
  - Dynamics, i.e. line-of-sight acceleration for each signal
- Pseudo-range and pseudo-range rate measurement noise may be correlated if pseudo-range is carrier-smoothed

Steps 7 to 10: Measurement Update

7: Calculate Kalman gain matrix

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

8: Formulate measurement innovation

$$\delta \mathbf{z}_k^- = \tilde{\mathbf{z}}_k - \mathbf{h}\left(\hat{\mathbf{x}}_k^- \right)$$

9: Update state vector estimate

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \delta \mathbf{z}_k^-$$

10: Update error covariance matrix

$$\mathbf{P}_k^+ = \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k \right) \mathbf{P}_k^-$$

5. Filtered GNSS Positioning

Using Delta Range Measurements

Carrier phase is measured by comparing incoming signal with reference (code must be demodulated first)

Much smaller errors due to

- Signal tracking
- Multipath

Pseudo-range from carrier phase subject to

- One wavelength ambiguity
- Receiver & satellite phase biases

Time differencing cancels out these errors

These **delta range** measurements may be used in an EKF *instead of pseudo-range rate*

- More precise; less resilient