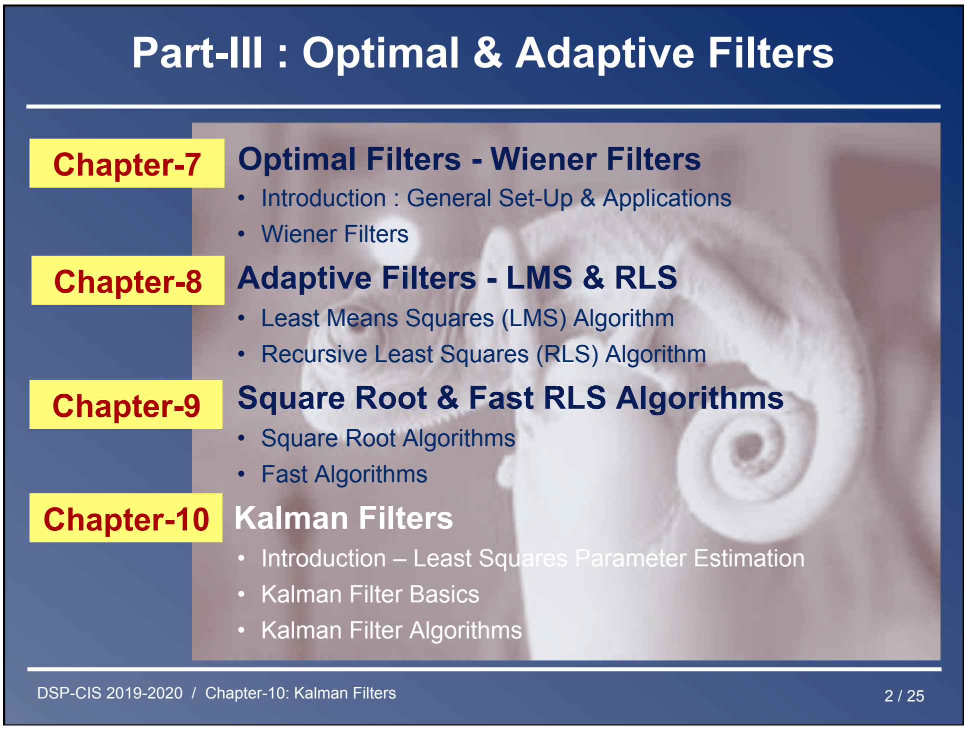
DSP-CIS

Part-III : Optimal & Adaptive Filters

Chapter-10 : Kalman Filters

Marc Moonen
Dept. E.E./ESAT-STADIUS, KU Leuven
marc.moonen@kuleuven.be
www.esat.kuleuven.be/stadius/



Part-III : Optimal & Adaptive Filters

Chapter-7 **Optimal Filters - Wiener Filters**

- Introduction : General Set-Up & Applications
- Wiener Filters

Chapter-8 **Adaptive Filters - LMS & RLS**

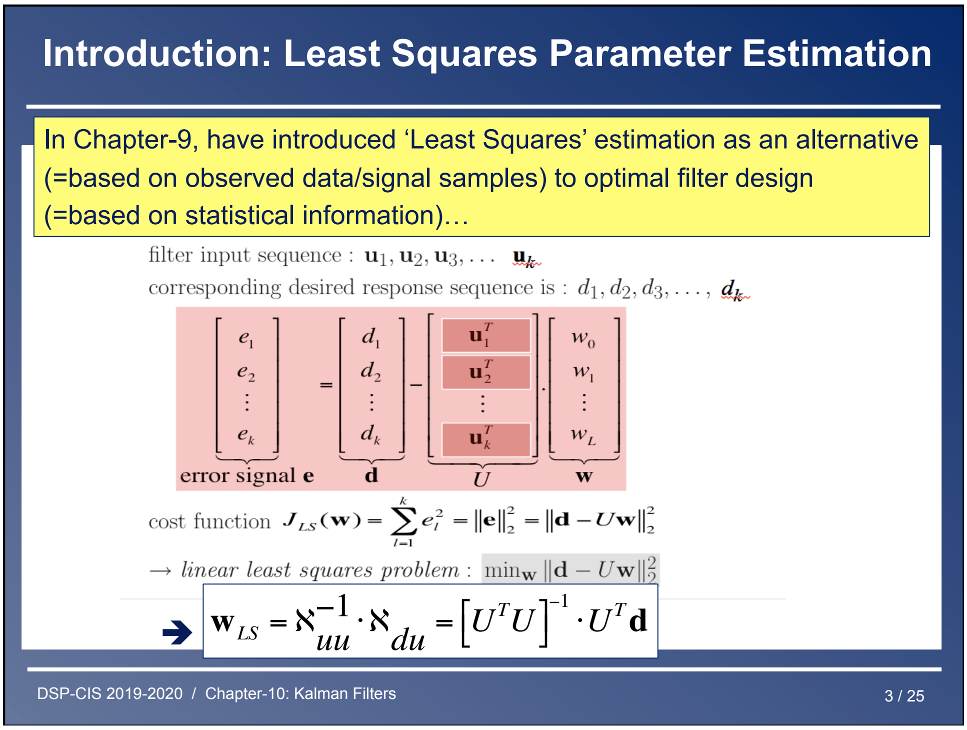
- Least Means Squares (LMS) Algorithm
- Recursive Least Squares (RLS) Algorithm

Chapter-9 **Square Root & Fast RLS Algorithms**

- Square Root Algorithms
- Fast Algorithms

Chapter-10 **Kalman Filters**

- Introduction – Least Squares Parameter Estimation
- Kalman Filter Basics
- Kalman Filter Algorithms



Introduction: Least Squares Parameter Estimation

In Chapter-9, have introduced ‘Least Squares’ estimation as an alternative (=based on observed data/signal samples) to optimal filter design (=based on statistical information)…

filter input sequence : $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \ldots \ \mathbf{u}_k$

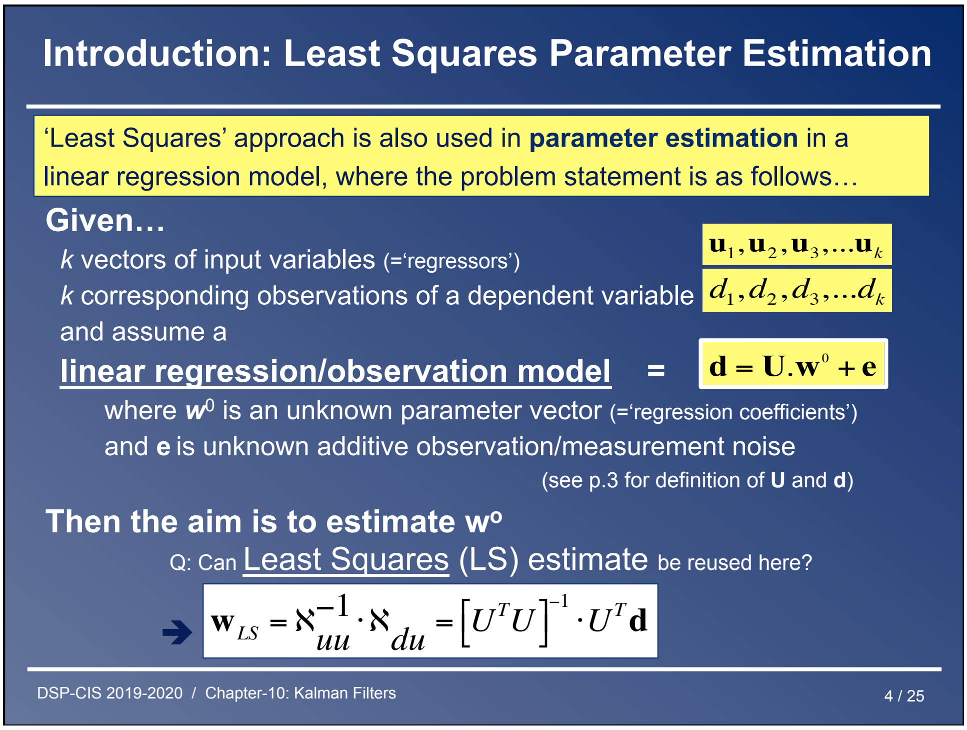
corresponding desired response sequence is : $d_1, d_2, d_3, \ldots, \ d_k$

$$\underbrace{\begin{bmatrix} e_1\\ e_2\\ \vdots\\ e_k\end{bmatrix}}_{\text{error signal } \mathbf{e}} = \underbrace{\begin{bmatrix} d_1\\ d_2\\ \vdots\\ d_k\end{bmatrix}}_{\mathbf{d}} - \underbrace{\begin{bmatrix}\mathbf{u}_1^T\\ \mathbf{u}_2^T\\ \vdots\\ \mathbf{u}_k^T\end{bmatrix}}_{U} \cdot \underbrace{\begin{bmatrix} w_0\\ w_1\\ \vdots\\ w_L\end{bmatrix}}_{\mathbf{w}}$$

cost function $J_{LS}(\mathbf{w}) = \sum_{i=1}^{k} e_i^2 = \|\mathbf{e}\|_2^2 = \|\mathbf{d} - U\mathbf{w}\|_2^2$

→ *linear least squares problem* : $\min_{\mathbf{w}} \|\mathbf{d} - U\mathbf{w}\|_2^2$

$$\rightarrow \boxed{\mathbf{w}_{LS} = \aleph_{uu}^{-1} \cdot \aleph_{du} = \left[U^T U\right]^{-1} \cdot U^T \mathbf{d}}$$



Introduction: Least Squares Parameter Estimation

‘Least Squares’ approach is also used in **parameter estimation** in a linear regression model, where the problem statement is as follows…

Given…

k vectors of input variables (=‘regressors’) $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \ldots \mathbf{u}_k$

k corresponding observations of a dependent variable $d_1, d_2, d_3, \ldots d_k$

and assume a

linear regression/observation model = $\mathbf{d} = \mathbf{U}.\mathbf{w}^o + \mathbf{e}$

where $\boldsymbol{w}^0$ is an unknown parameter vector (=‘regression coefficients’)

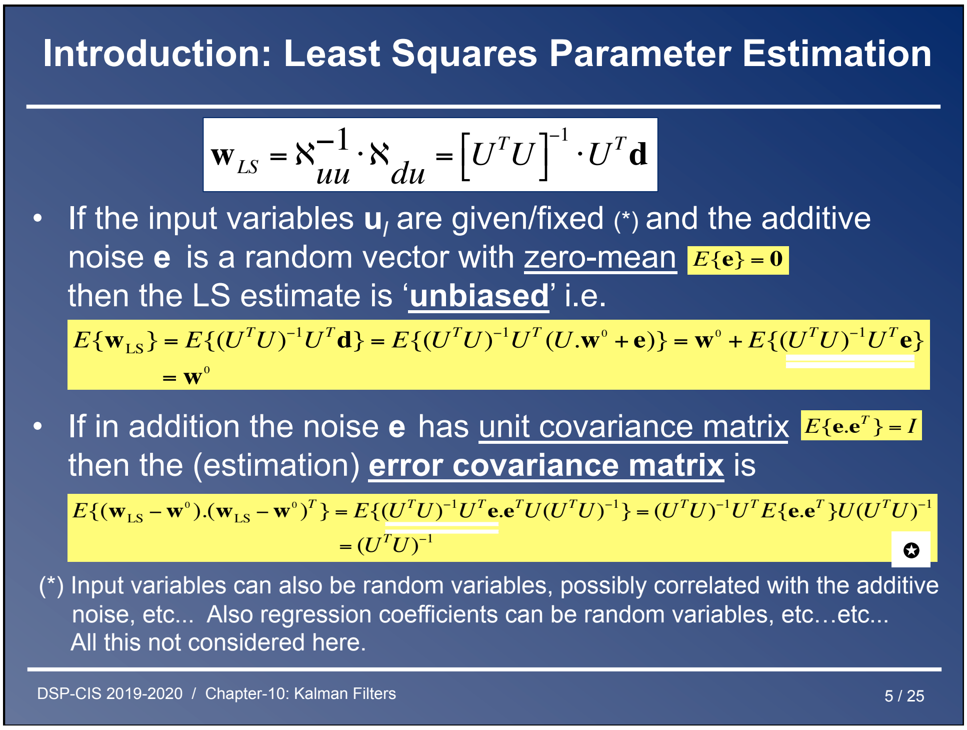
and **e** is unknown additive observation/measurement noise

(see p.3 for definition of **U** and **d**)

Then the aim is to estimate wᵒ

Q: Can Least Squares (LS) estimate be reused here?

$$\rightarrow \boxed{\mathbf{w}_{LS} = \aleph_{uu}^{-1} \cdot \aleph_{du} = \left[U^T U\right]^{-1} \cdot U^T \mathbf{d}}$$



Introduction: Least Squares Parameter Estimation

$$\mathbf{w}_{LS} = \aleph_{uu}^{-1} \cdot \aleph_{du} = \left[U^T U\right]^{-1} \cdot U^T \mathbf{d}$$

- If the input variables $\mathbf{u}_l$ are given/fixed (*) and the additive noise **e** is a random vector with zero-mean $E\{\mathbf{e}\} = \mathbf{0}$ then the LS estimate is '**unbiased**' i.e.

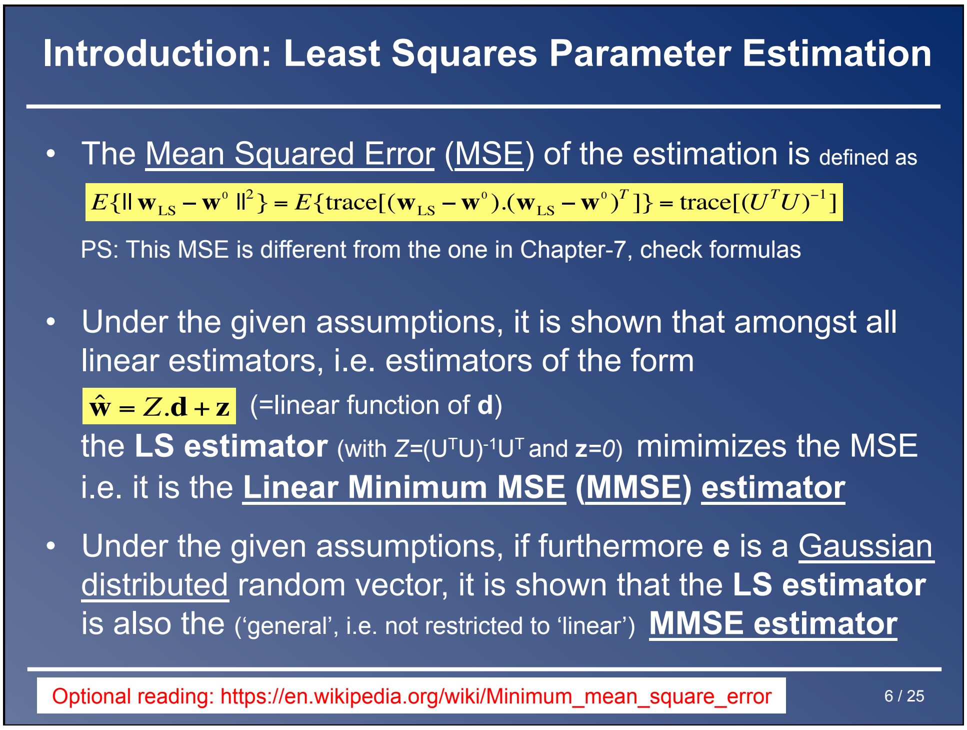
$$E\{\mathbf{w}_{\mathrm{LS}}\} = E\{(U^T U)^{-1} U^T \mathbf{d}\} = E\{(U^T U)^{-1} U^T (U.\mathbf{w}^o + \mathbf{e})\} = \mathbf{w}^o + E\{\underline{\underline{(U^T U)^{-1} U^T}}\mathbf{e}\}$$
$$= \mathbf{w}^0$$

- If in addition the noise **e** has unit covariance matrix $E\{\mathbf{e}.\mathbf{e}^T\} = I$ then the (estimation) **error covariance matrix** is

$$E\{(\mathbf{w}_{\mathrm{LS}} - \mathbf{w}^o).(\mathbf{w}_{\mathrm{LS}} - \mathbf{w}^o)^T\} = E\{\underline{\underline{(U^T U)^{-1} U^T}}\mathbf{e}.\mathbf{e}^T U (U^T U)^{-1}\} = (U^T U)^{-1} U^T E\{\mathbf{e}.\mathbf{e}^T\} U (U^T U)^{-1}$$
$$= (U^T U)^{-1}$$

✪

(*) Input variables can also be random variables, possibly correlated with the additive noise, etc... Also regression coefficients can be random variables, etc…etc... All this not considered here.



Introduction: Least Squares Parameter Estimation

- The Mean Squared Error (MSE) of the estimation is defined as

$$E\{\|\mathbf{w}_{\mathrm{LS}} - \mathbf{w}^o\|^2\} = E\{\mathrm{trace}[(\mathbf{w}_{\mathrm{LS}} - \mathbf{w}^o).(\mathbf{w}_{\mathrm{LS}} - \mathbf{w}^o)^T]\} = \mathrm{trace}[(U^T U)^{-1}]$$

PS: This MSE is different from the one in Chapter-7, check formulas

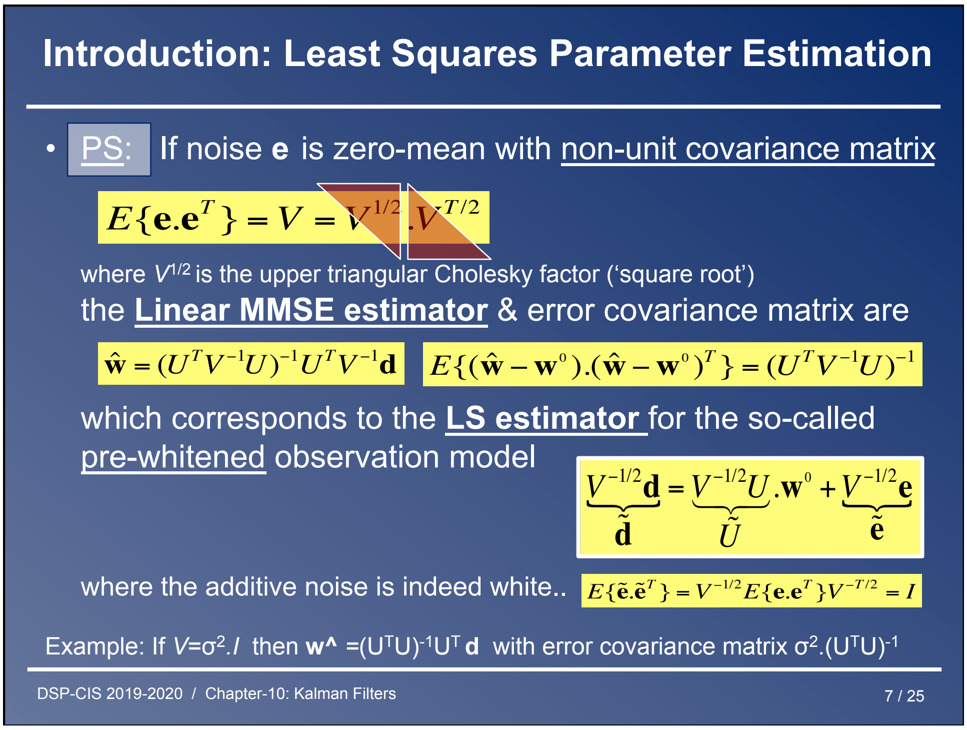
- Under the given assumptions, it is shown that amongst all linear estimators, i.e. estimators of the form

$\hat{\mathbf{w}} = Z.\mathbf{d} + \mathbf{z}$ (=linear function of **d**)

the **LS estimator** (with Z=$(U^T U)^{-1} U^T$ and $\mathbf{z}$=0) mimimizes the MSE i.e. it is the **Linear Minimum MSE (MMSE) estimator**

- Under the given assumptions, if furthermore **e** is a Gaussian distributed random vector, it is shown that the **LS estimator** is also the ('general', i.e. not restricted to 'linear') **MMSE estimator**

Optional reading: https://en.wikipedia.org/wiki/Minimum_mean_square_error



Introduction: Least Squares Parameter Estimation

- PS: If noise **e** is zero-mean with non-unit covariance matrix

$$E\{\mathbf{e}.\mathbf{e}^T\} = V = V^{1/2}.V^{T/2}$$

where $V^{1/2}$ is the upper triangular Cholesky factor ('square root')

the **Linear MMSE estimator** & error covariance matrix are

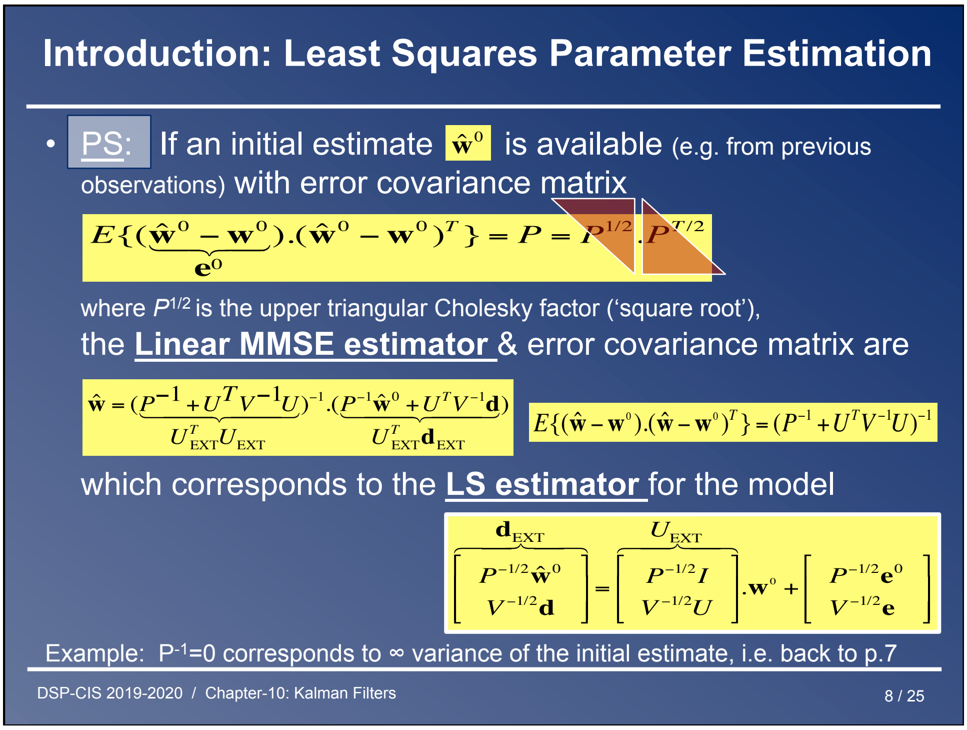
$$\hat{\mathbf{w}} = (U^T V^{-1} U)^{-1} U^T V^{-1} \mathbf{d} \qquad E\{(\hat{\mathbf{w}} - \mathbf{w}^0).(\hat{\mathbf{w}} - \mathbf{w}^0)^T\} = (U^T V^{-1} U)^{-1}$$

which corresponds to the **LS estimator** for the so-called pre-whitened observation model

$$\underbrace{V^{-1/2}\mathbf{d}}_{\tilde{\mathbf{d}}} = \underbrace{V^{-1/2}U}_{\tilde{U}}.\mathbf{w}^0 + \underbrace{V^{-1/2}\mathbf{e}}_{\tilde{\mathbf{e}}}$$

where the additive noise is indeed white.. $E\{\tilde{\mathbf{e}}.\tilde{\mathbf{e}}^T\} = V^{-1/2} E\{\mathbf{e}.\mathbf{e}^T\} V^{-T/2} = I$

Example: If $V=\sigma^2.I$ then **w^** $=(U^TU)^{-1}U^T$ **d** with error covariance matrix $\sigma^2.(U^TU)^{-1}$



Introduction: Least Squares Parameter Estimation

- PS: If an initial estimate $\hat{\mathbf{w}}^0$ is available (e.g. from previous observations) with error covariance matrix

$$E\{(\underbrace{\hat{\mathbf{w}}^0 - \mathbf{w}^0}_{\mathbf{e}^0}).(\hat{\mathbf{w}}^0 - \mathbf{w}^0)^T\} = P = P^{1/2}.P^{T/2}$$

where $P^{1/2}$ is the upper triangular Cholesky factor ('square root'),

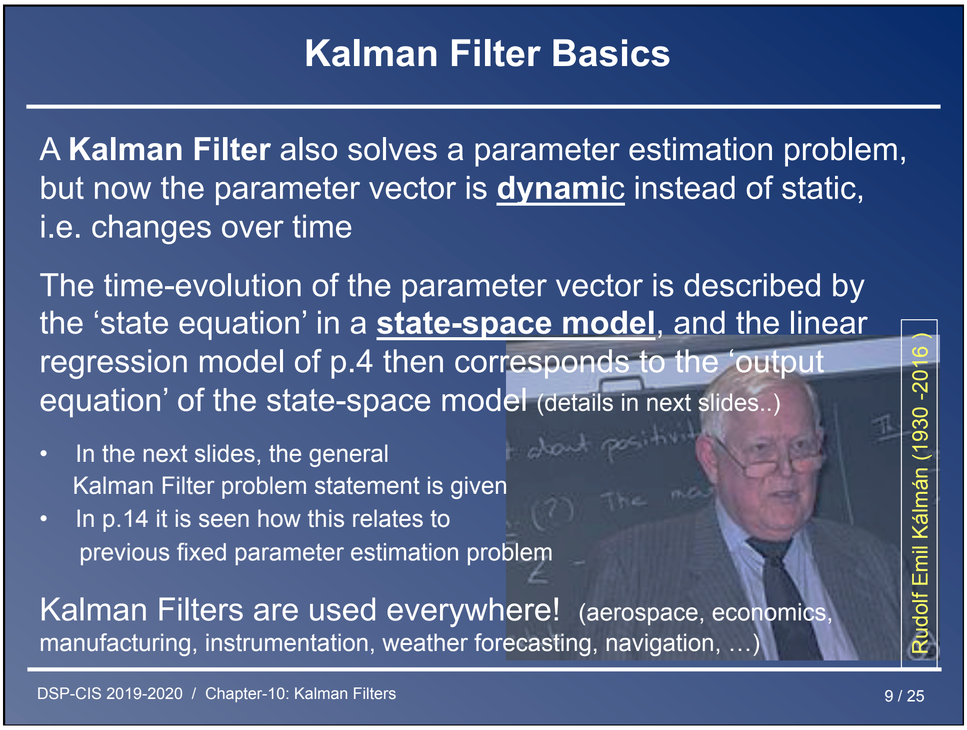
the **Linear MMSE estimator** & error covariance matrix are

$$\hat{\mathbf{w}} = (\underbrace{P^{-1} + U^T V^{-1} U}_{U_{\mathrm{EXT}}^T U_{\mathrm{EXT}}})^{-1}.(\underbrace{P^{-1}\hat{\mathbf{w}}^0 + U^T V^{-1}\mathbf{d}}_{U_{\mathrm{EXT}}^T \mathbf{d}_{\mathrm{EXT}}}) \qquad E\{(\hat{\mathbf{w}} - \mathbf{w}^0).(\hat{\mathbf{w}} - \mathbf{w}^0)^T\} = (P^{-1} + U^T V^{-1} U)^{-1}$$

which corresponds to the **LS estimator** for the model

$$\overbrace{\begin{bmatrix} P^{-1/2}\hat{\mathbf{w}}^0 \\ V^{-1/2}\mathbf{d} \end{bmatrix}}^{\mathbf{d}_{\mathrm{EXT}}} = \overbrace{\begin{bmatrix} P^{-1/2} I \\ V^{-1/2} U \end{bmatrix}}^{U_{\mathrm{EXT}}}.\mathbf{w}^0 + \begin{bmatrix} P^{-1/2}\mathbf{e}^0 \\ V^{-1/2}\mathbf{e} \end{bmatrix}$$

Example: $P^{-1}=0$ corresponds to ∞ variance of the initial estimate, i.e. back to p.7



Kalman Filter Basics

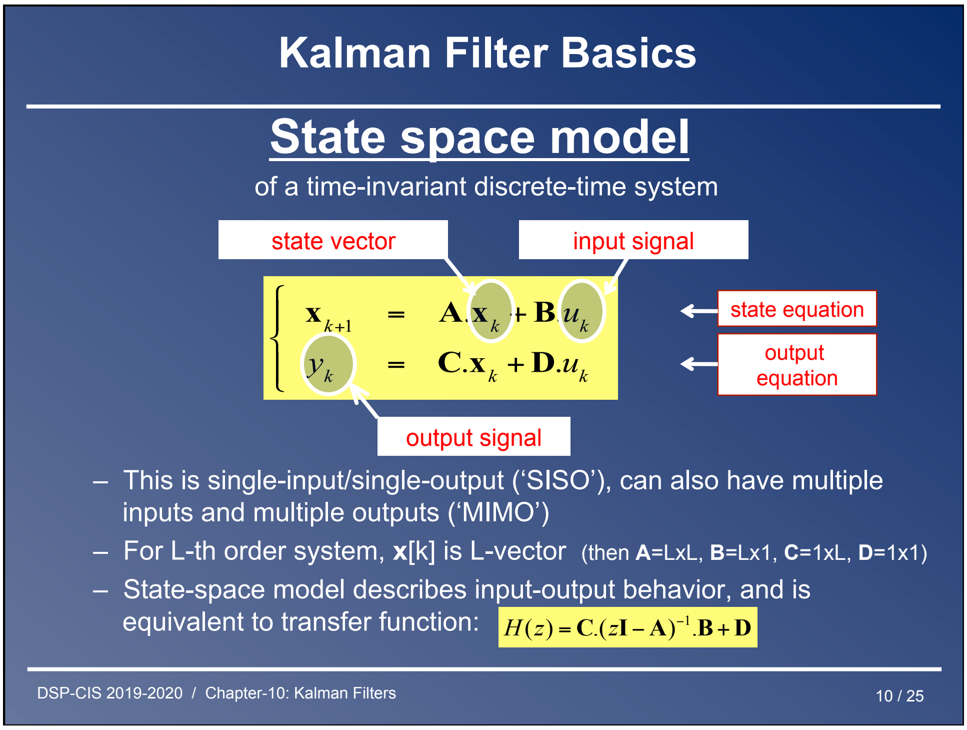
A **Kalman Filter** also solves a parameter estimation problem, but now the parameter vector is **dynamic** instead of static, i.e. changes over time

The time-evolution of the parameter vector is described by the 'state equation' in a **state-space model**, and the linear regression model of p.4 then corresponds to the 'output equation' of the state-space model (details in next slides..)

- In the next slides, the general Kalman Filter problem statement is given
- In p.14 it is seen how this relates to previous fixed parameter estimation problem

Kalman Filters are used everywhere! (aerospace, economics, manufacturing, instrumentation, weather forecasting, navigation, …)

Rudolf Emil Kálmán (1930 -2016)



Kalman Filter Basics

State space model

of a time-invariant discrete-time system

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{A}.\mathbf{x}_k + \mathbf{B}.u_k \\ y_k = \mathbf{C}.\mathbf{x}_k + \mathbf{D}.u_k \end{cases}$$

state vector
input signal
state equation
output equation
output signal

- This is single-input/single-output ('SISO'), can also have multiple inputs and multiple outputs ('MIMO')
- For L-th order system, **x**[k] is L-vector (then **A**=LxL, **B**=Lx1, **C**=1xL, **D**=1x1)
- State-space model describes input-output behavior, and is equivalent to transfer function: $H(z) = \mathbf{C}.(z\mathbf{I}-\mathbf{A})^{-1}.\mathbf{B}+\mathbf{D}$

Kalman Filter Basics

- Example: IIR filter

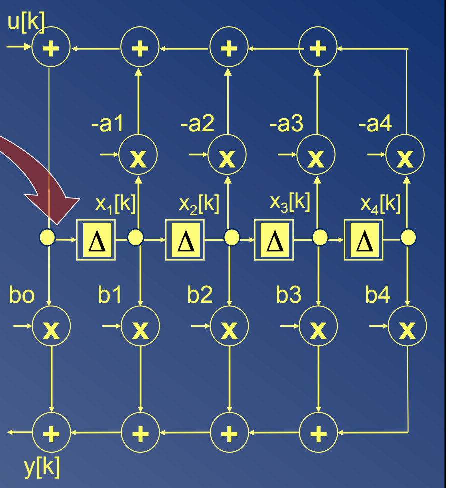
$$y[k]=b_0.u[k]+...+b_4.u[k-L]$$
$$-a_1.y[k-1]-...-a_4.y[k-L]$$

State space model is

$$\begin{bmatrix} x_1[k+1] \\ x_2[k+1] \\ x_3[k+1] \\ x_4[k+1] \end{bmatrix} = \begin{bmatrix} -a_1 & -a_2 & -a_3 & -a_4 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} . \begin{bmatrix} x_1[k] \\ x_2[k] \\ x_3[k] \\ x_4[k] \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} .u[k]$$

$$y[k] = \begin{bmatrix} b_1-a_1b_0 & b_2-a_2b_0 & b_3-a_3b_0 & b_4-a_4b_0 \end{bmatrix} . \begin{bmatrix} x_1[k] \\ x_2[k] \\ x_3[k] \\ x_4[k] \end{bmatrix} + b_0.u[k]$$

$$H(z)=\mathbf{C}.(z\mathbf{I}-\mathbf{A})^{-1}.\mathbf{B}+\mathbf{D}=...=\frac{B(z)}{A(z)}$$



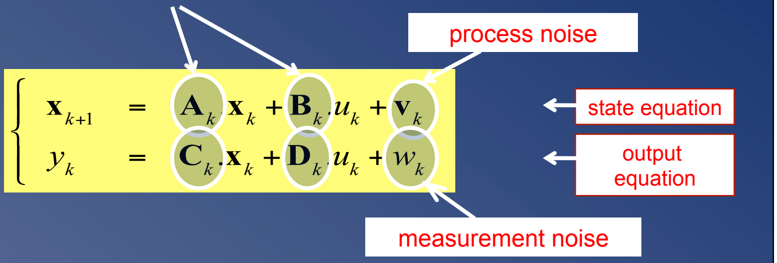
PS: Remember we mostly use shorthand notation, i.e. $\mathbf{x}_k$, y_k ,.. instead of **x**[k], y[k],..

Kalman Filter Basics

Will consider a more general

State space model

of a time-varying discrete-time system + noise



process noise

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{A}_k.\mathbf{x}_k + \mathbf{B}_k.u_k + \mathbf{v}_k \\ y_k = \mathbf{C}_k.\mathbf{x}_k + \mathbf{D}_k.u_k + w_k \end{cases}$$

state equation

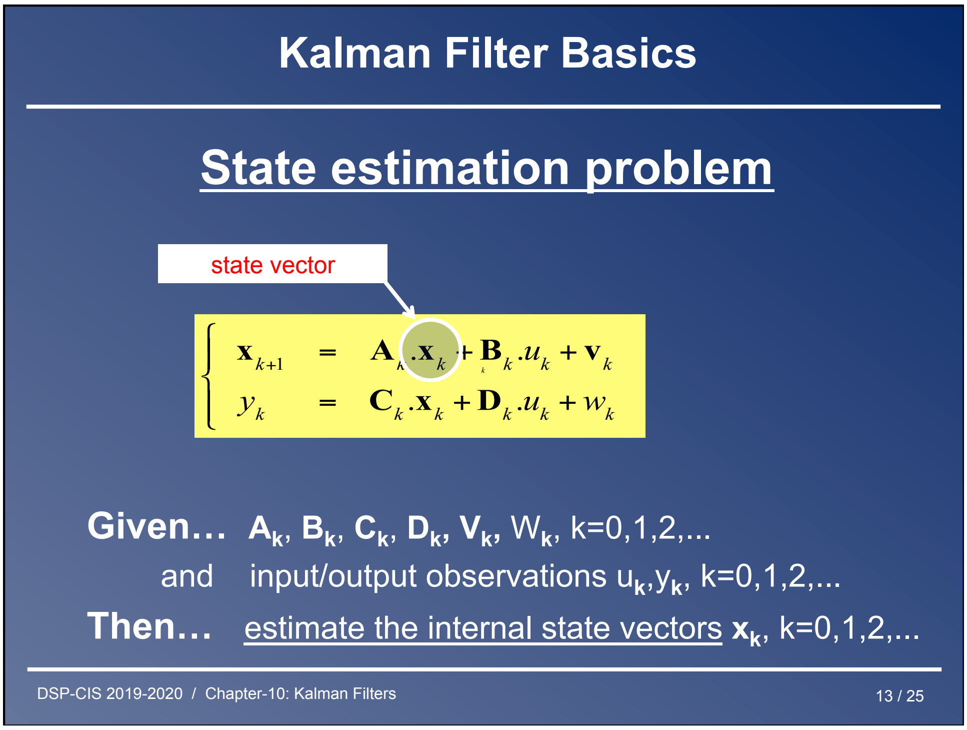
output equation

measurement noise

where $\mathbf{v}_k$ and w_k are mutually uncorrelated, zero mean, white noises

$$E\{\begin{bmatrix} \mathbf{v}_k \\ w_k \end{bmatrix}.\begin{bmatrix} \mathbf{v}_l^T & w_l^T \end{bmatrix}\} = \delta_{kl}.\begin{bmatrix} \mathbf{V}_k & \mathbf{0} \\ \mathbf{0} & W_k \end{bmatrix}$$

$$\mathbf{V}_k = \mathbf{V}_k^{\frac{1}{2}}.\mathbf{V}_k^{\frac{T}{2}}$$
$$= \text{Cholesky/square-root factorization}$$



Kalman Filter Basics

State estimation problem

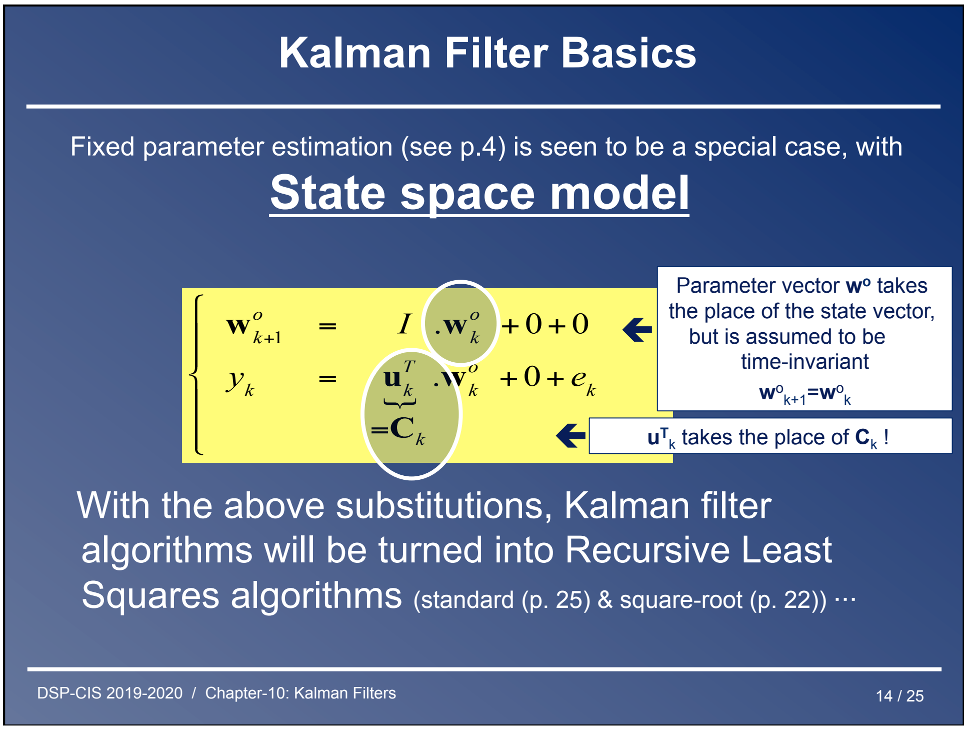
state vector

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{A}_k.\mathbf{x}_k + \mathbf{B}_k.u_k + \mathbf{v}_k \\ y_k = \mathbf{C}_k.\mathbf{x}_k + \mathbf{D}_k.u_k + w_k \end{cases}$$

Given… $\mathbf{A}_k$, $\mathbf{B}_k$, $\mathbf{C}_k$, $\mathbf{D}_k$, $\mathbf{V}_k$, W_k, k=0,1,2,...

and input/output observations u_k,y_k, k=0,1,2,...

Then… estimate the internal state vectors $\mathbf{x}_k$, k=0,1,2,...



Kalman Filter Basics

Fixed parameter estimation (see p.4) is seen to be a special case, with

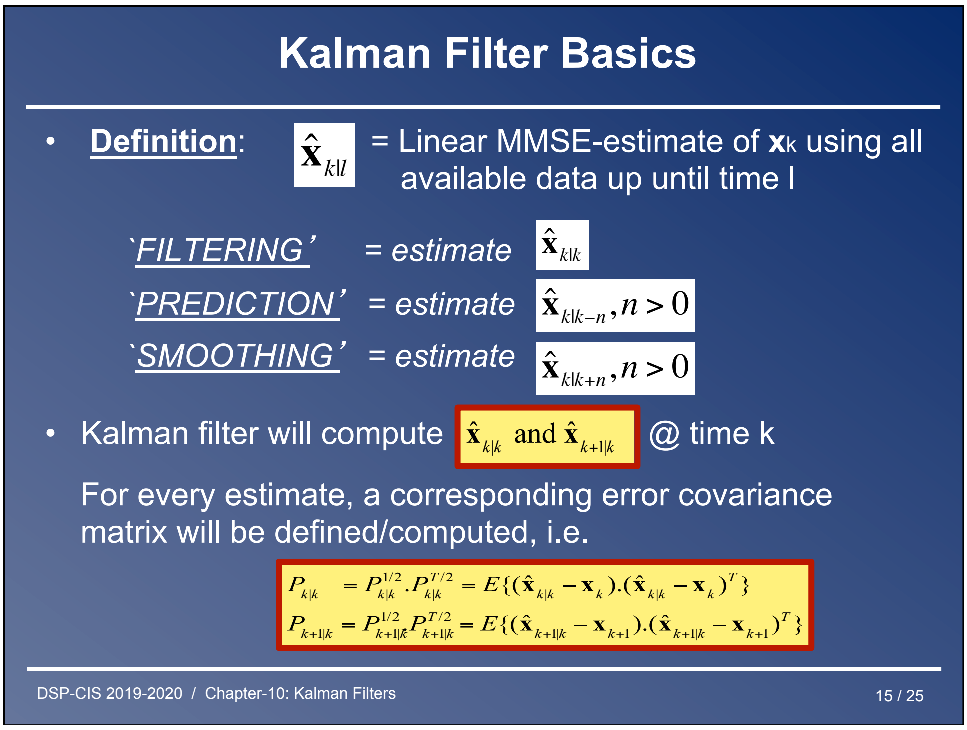
State space model

$$\begin{cases} \mathbf{w}^o_{k+1} = I\,.\mathbf{w}^o_k + 0 + 0 \\ y_k = \underbrace{\mathbf{u}^T_k}_{=\mathbf{C}_k}.\mathbf{w}^o_k + 0 + e_k \end{cases}$$

Parameter vector **w°** takes the place of the state vector, but is assumed to be time-invariant **$\mathbf{w}^o_{k+1}=\mathbf{w}^o_k$**

$\mathbf{u}^T_k$ takes the place of **$\mathbf{C}_k$** !

With the above substitutions, Kalman filter algorithms will be turned into Recursive Least Squares algorithms (standard (p. 25) & square-root (p. 22)) …



Kalman Filter Basics

- **Definition**: $\hat{\mathbf{x}}_{k|l}$ = Linear MMSE-estimate of $\mathbf{x}_k$ using all available data up until time l

  `FILTERING' = *estimate* $\hat{\mathbf{x}}_{k|k}$

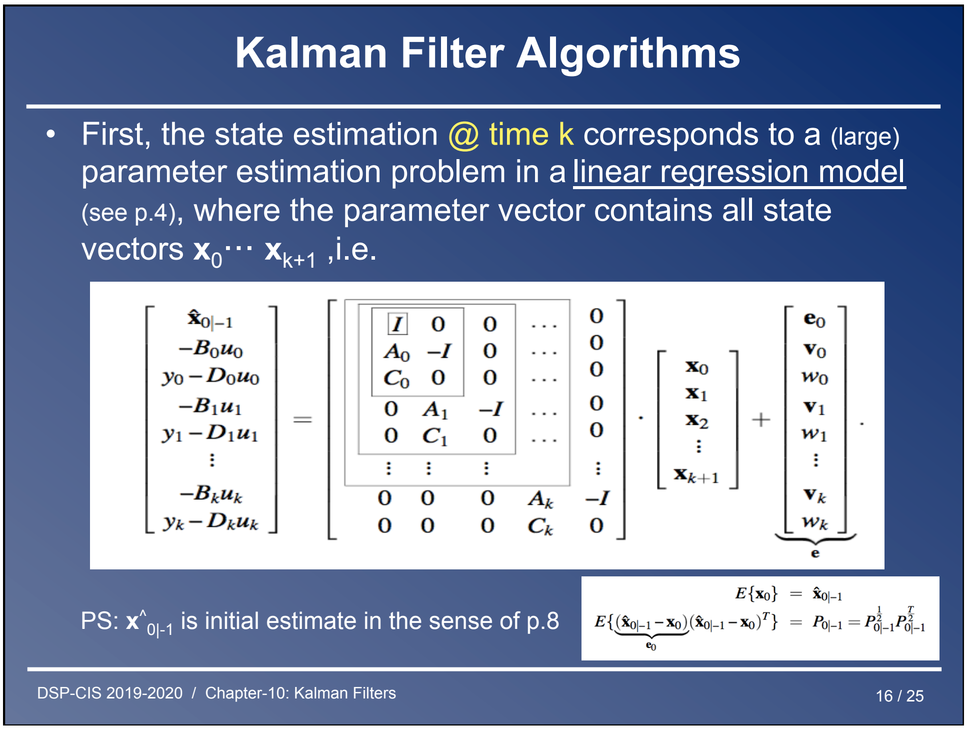
  `PREDICTION' = *estimate* $\hat{\mathbf{x}}_{k|k-n}, n > 0$

  `SMOOTHING' = *estimate* $\hat{\mathbf{x}}_{k|k+n}, n > 0$

- Kalman filter will compute $\hat{\mathbf{x}}_{k|k}$ and $\hat{\mathbf{x}}_{k+1|k}$ @ time k

  For every estimate, a corresponding error covariance matrix will be defined/computed, i.e.

$$\begin{aligned} P_{k|k} &= P_{k|k}^{1/2}.P_{k|k}^{T/2} = E\{(\hat{\mathbf{x}}_{k|k}-\mathbf{x}_k).(\hat{\mathbf{x}}_{k|k}-\mathbf{x}_k)^T\} \\ P_{k+1|k} &= P_{k+1|k}^{1/2}.P_{k+1|k}^{T/2} = E\{(\hat{\mathbf{x}}_{k+1|k}-\mathbf{x}_{k+1}).(\hat{\mathbf{x}}_{k+1|k}-\mathbf{x}_{k+1})^T\} \end{aligned}$$



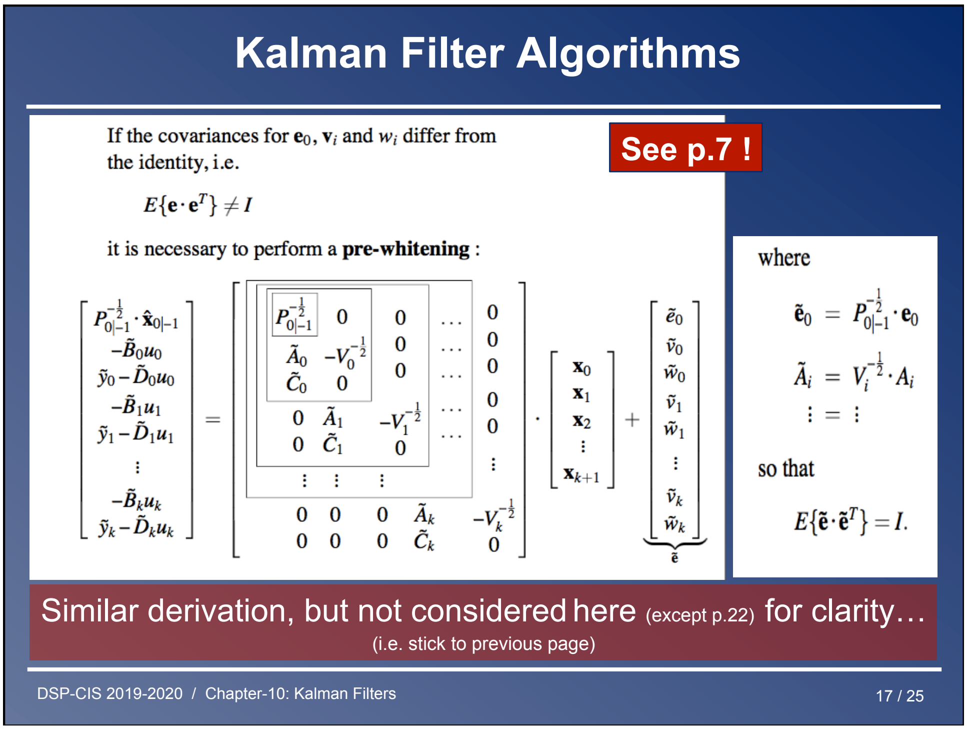
Kalman Filter Algorithms

- First, the state estimation @ time k corresponds to a (large) parameter estimation problem in a linear regression model (see p.4), where the parameter vector contains all state vectors $\mathbf{x}_0 \cdots \mathbf{x}_{k+1}$,i.e.

$$\begin{bmatrix} \hat{\mathbf{x}}_{0|-1} \\ -B_0 u_0 \\ y_0 - D_0 u_0 \\ -B_1 u_1 \\ y_1 - D_1 u_1 \\ \vdots \\ -B_k u_k \\ y_k - D_k u_k \end{bmatrix} = \begin{bmatrix} \boxed{I} & 0 & 0 & \cdots & 0 \\ A_0 & -I & 0 & \cdots & 0 \\ C_0 & 0 & 0 & \cdots & 0 \\ 0 & A_1 & -I & \cdots & 0 \\ 0 & C_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & A_k & -I \\ 0 & 0 & 0 & C_k & 0 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_{k+1} \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{e}_0 \\ \mathbf{v}_0 \\ w_0 \\ \mathbf{v}_1 \\ w_1 \\ \vdots \\ \mathbf{v}_k \\ w_k \end{bmatrix}}_{\mathbf{e}} .$$

PS: **x**^$_{0|-1}$ is initial estimate in the sense of p.8

$$\begin{aligned} E\{\mathbf{x}_0\} &= \hat{\mathbf{x}}_{0|-1} \\ E\{\underbrace{(\hat{\mathbf{x}}_{0|-1}-\mathbf{x}_0)}_{\mathbf{e}_0}(\hat{\mathbf{x}}_{0|-1}-\mathbf{x}_0)^T\} &= P_{0|-1} = P_{0|-1}^{\frac{1}{2}} P_{0|-1}^{\frac{T}{2}} \end{aligned}$$



Kalman Filter Algorithms

If the covariances for $\mathbf{e}_0$, $\mathbf{v}_i$ and w_i differ from the identity, i.e.

See p.7 !

$$E\{\mathbf{e}\cdot\mathbf{e}^T\} \neq I$$

it is necessary to perform a **pre-whitening** :

$$\begin{bmatrix} P_{0|-1}^{-\frac{1}{2}}\cdot\hat{\mathbf{x}}_{0|-1} \\ -\tilde{B}_0 u_0 \\ \tilde{y}_0-\tilde{D}_0 u_0 \\ -\tilde{B}_1 u_1 \\ \tilde{y}_1-\tilde{D}_1 u_1 \\ \vdots \\ -\tilde{B}_k u_k \\ \tilde{y}_k-\tilde{D}_k u_k \end{bmatrix} = \begin{bmatrix} \boxed{P_{0|-1}^{-\frac{1}{2}}} & 0 & 0 & \cdots & 0 \\ \tilde{A}_0 & -V_0^{-\frac{1}{2}} & 0 & \cdots & 0 \\ \tilde{C}_0 & 0 & 0 & \cdots & 0 \\ 0 & \tilde{A}_1 & -V_1^{-\frac{1}{2}} & \cdots & 0 \\ 0 & \tilde{C}_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \tilde{A}_k & -V_k^{-\frac{1}{2}} \\ 0 & 0 & 0 & \tilde{C}_k & 0 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_{k+1} \end{bmatrix} + \underbrace{\begin{bmatrix} \tilde{e}_0 \\ \tilde{v}_0 \\ \tilde{w}_0 \\ \tilde{v}_1 \\ \tilde{w}_1 \\ \vdots \\ \tilde{v}_k \\ \tilde{w}_k \end{bmatrix}}_{\tilde{\mathbf{e}}}$$

where

$$\tilde{\mathbf{e}}_0 = P_{0|-1}^{-\frac{1}{2}}\cdot\mathbf{e}_0$$

$$\tilde{A}_i = V_i^{-\frac{1}{2}}\cdot A_i$$

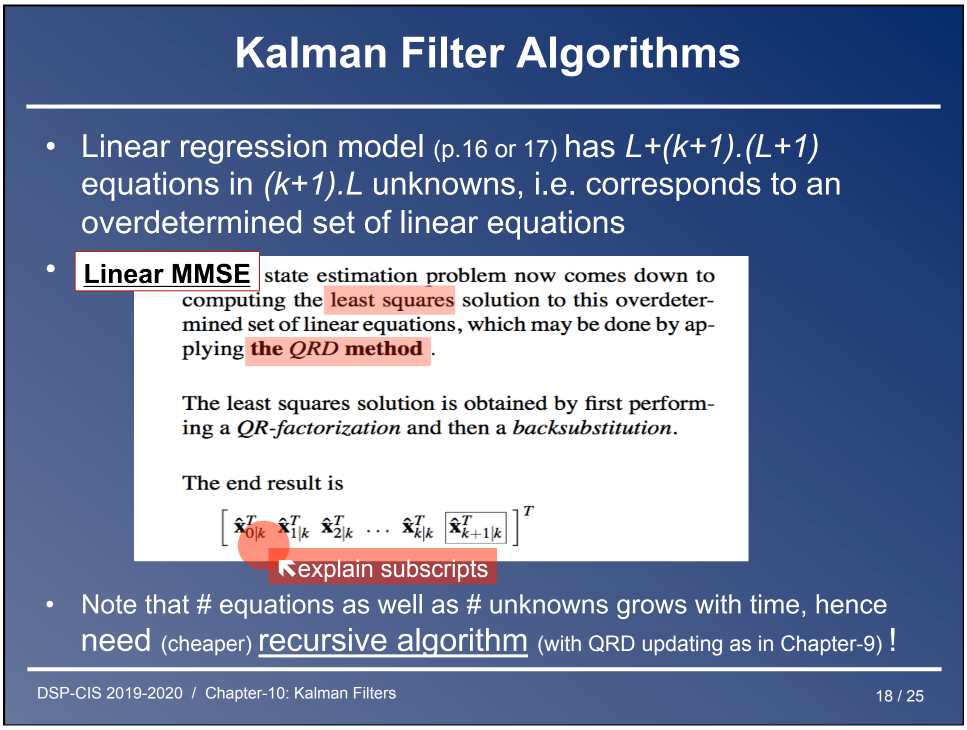
$$\vdots = \vdots$$

so that

$$E\{\tilde{\mathbf{e}}\cdot\tilde{\mathbf{e}}^T\} = I.$$

Similar derivation, but not considered here (except p.22) for clarity…

(i.e. stick to previous page)



Kalman Filter Algorithms

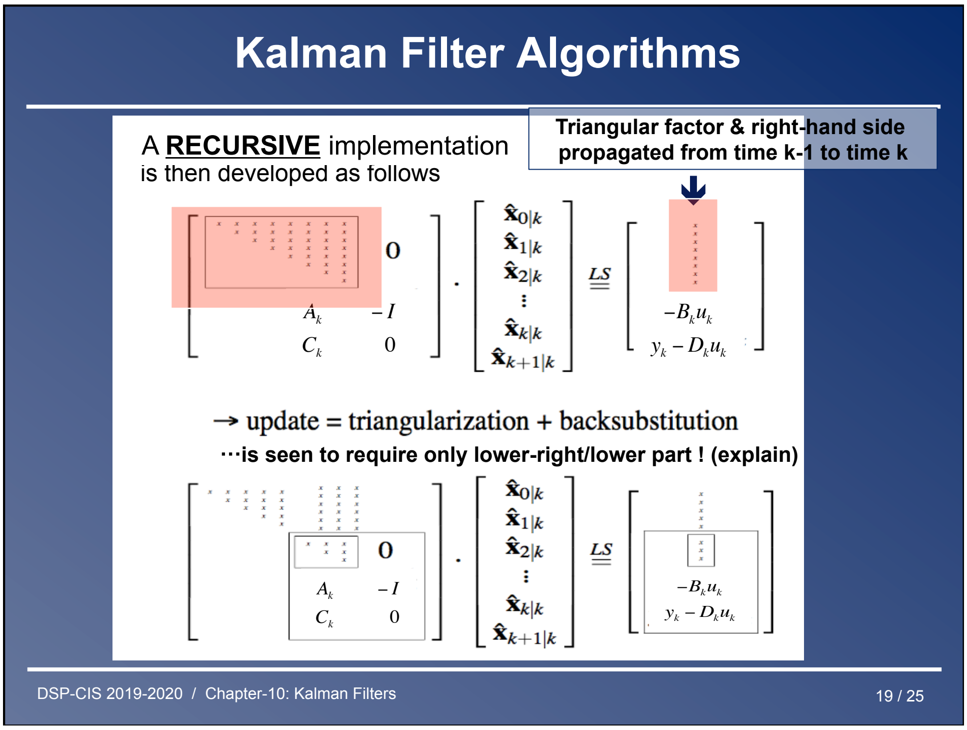
- Linear regression model (p.16 or 17) has *L+(k+1).(L+1)* equations in *(k+1).L* unknowns, i.e. corresponds to an overdetermined set of linear equations
- **Linear MMSE** state estimation problem now comes down to computing the least squares solution to this overdetermined set of linear equations, which may be done by applying **the *QRD* method** .

  The least squares solution is obtained by first performing a *QR-factorization* and then a *backsubstitution*.

  The end result is

  $$\begin{bmatrix} \hat{\mathbf{x}}_{0|k}^T & \hat{\mathbf{x}}_{1|k}^T & \hat{\mathbf{x}}_{2|k}^T & \cdots & \hat{\mathbf{x}}_{k|k}^T & \boxed{\hat{\mathbf{x}}_{k+1|k}^T} \end{bmatrix}^T$$

  explain subscripts
- Note that # equations as well as # unknowns grows with time, hence need (cheaper) recursive algorithm (with QRD updating as in Chapter-9) !



Kalman Filter Algorithms

A **RECURSIVE** implementation is then developed as follows

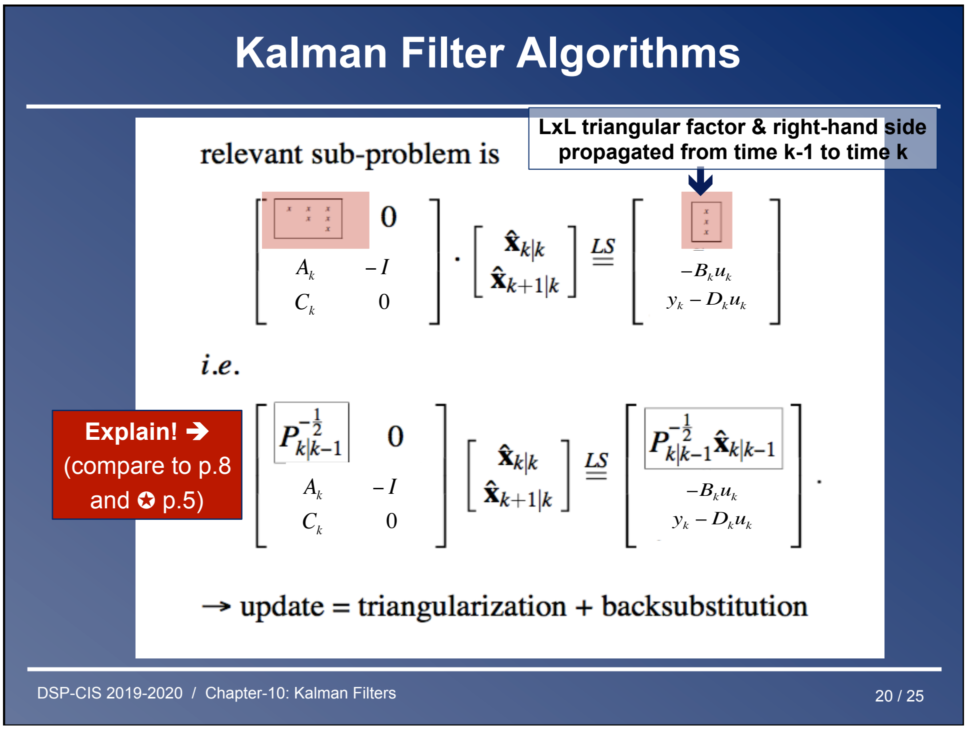
Triangular factor & right-hand side propagated from time k-1 to time k

$$\begin{bmatrix} \begin{matrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \ddots & \times & \times \\ & & & \times & \times \\ & & & & \times \end{matrix} & 0 \\ A_k & -I \\ C_k & 0 \end{bmatrix} \cdot \begin{bmatrix} \hat{\mathbf{x}}_{0|k} \\ \hat{\mathbf{x}}_{1|k} \\ \hat{\mathbf{x}}_{2|k} \\ \vdots \\ \hat{\mathbf{x}}_{k|k} \\ \hat{\mathbf{x}}_{k+1|k} \end{bmatrix} \underset{=}{\underline{LS}} \begin{bmatrix} \times \\ \vdots \\ \times \\ -B_k u_k \\ y_k - D_k u_k \end{bmatrix}$$

$\rightarrow$ update = triangularization + backsubstitution

…is seen to require only lower-right/lower part ! (explain)

$$\begin{bmatrix} \begin{matrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \ddots & \times & \times \end{matrix} & \\ & \boxed{\begin{matrix} \begin{matrix} \times & \times \\ & \times \end{matrix} & 0 \\ A_k & -I \\ C_k & 0 \end{matrix}} \end{bmatrix} \cdot \begin{bmatrix} \hat{\mathbf{x}}_{0|k} \\ \hat{\mathbf{x}}_{1|k} \\ \hat{\mathbf{x}}_{2|k} \\ \vdots \\ \hat{\mathbf{x}}_{k|k} \\ \hat{\mathbf{x}}_{k+1|k} \end{bmatrix} \underset{=}{\underline{LS}} \begin{bmatrix} \times \\ \vdots \\ \boxed{\begin{matrix} \times \\ \times \\ -B_k u_k \\ y_k - D_k u_k \end{matrix}} \end{bmatrix}$$



Kalman Filter Algorithms

relevant sub-problem is

LxL triangular factor & right-hand side propagated from time k-1 to time k

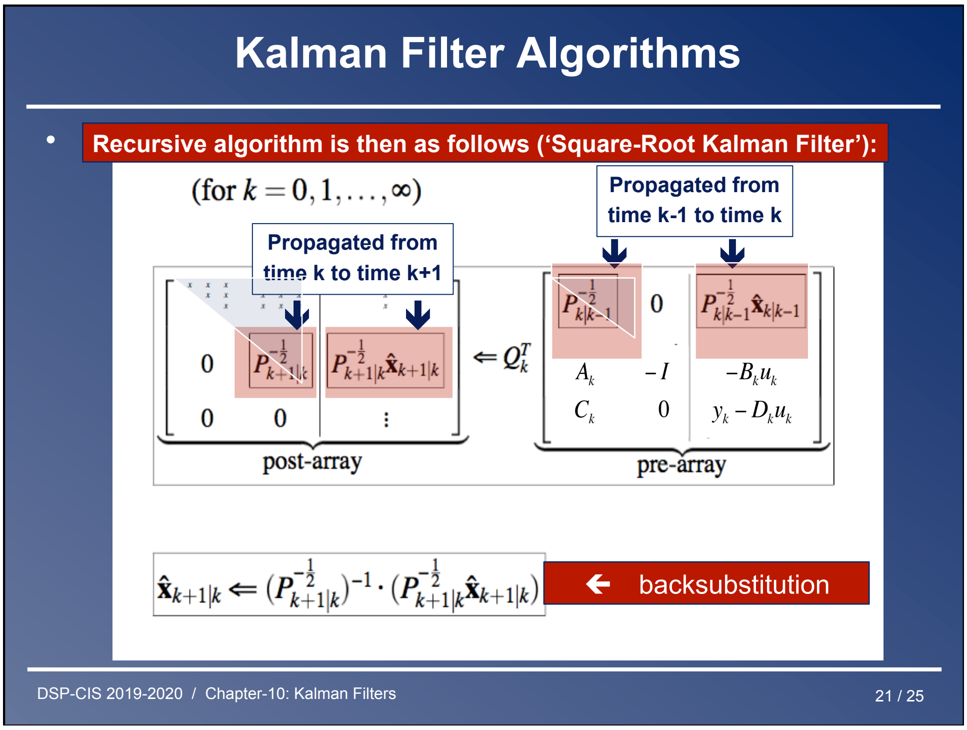
$$\begin{bmatrix} \begin{matrix} \times & \times & \times \\ & \times & \times \\ & & \times \end{matrix} & 0 \\ A_k & -I \\ C_k & 0 \end{bmatrix} \cdot \begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \hat{\mathbf{x}}_{k+1|k} \end{bmatrix} \underset{=}{\underline{LS}} \begin{bmatrix} \times \\ \times \\ \times \\ -B_k u_k \\ y_k - D_k u_k \end{bmatrix}$$

i.e.

$$\begin{bmatrix} P_{k|k-1}^{-\frac{1}{2}} & 0 \\ A_k & -I \\ C_k & 0 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \hat{\mathbf{x}}_{k+1|k} \end{bmatrix} \underset{=}{\underline{LS}} \begin{bmatrix} P_{k|k-1}^{-\frac{1}{2}} \hat{\mathbf{x}}_{k|k-1} \\ -B_k u_k \\ y_k - D_k u_k \end{bmatrix}.$$

Explain! ➔ (compare to p.8 and ✪ p.5)

$\rightarrow$ update = triangularization + backsubstitution



Kalman Filter Algorithms

- **Recursive algorithm is then as follows ('Square-Root Kalman Filter'):**

(for $k=0,1,\ldots,\infty$)

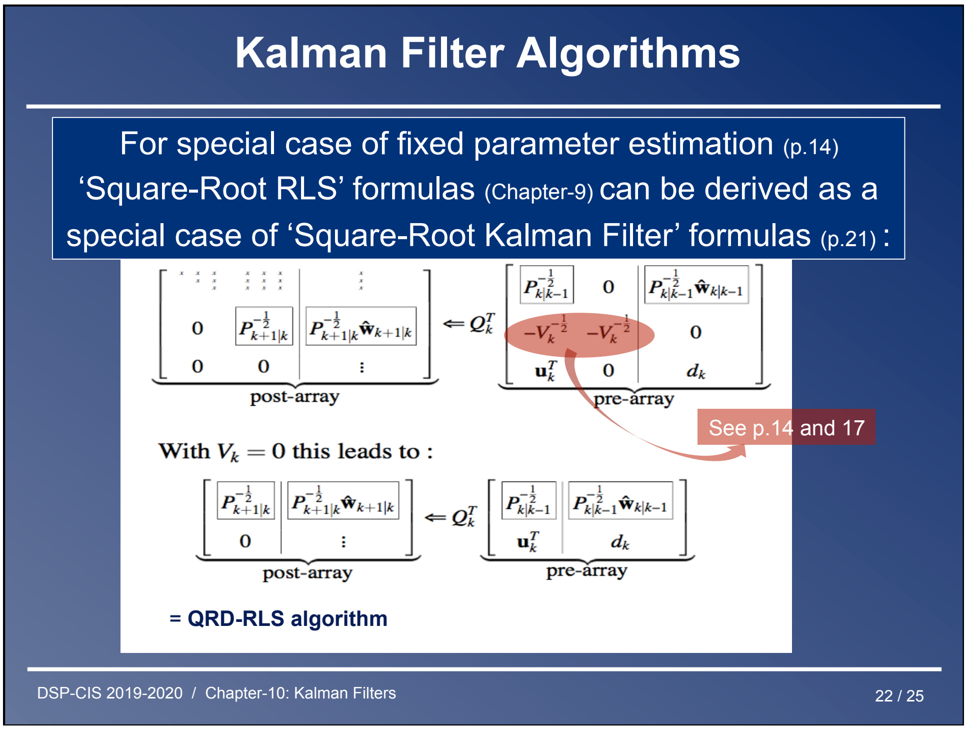
Propagated from time k to time k+1

Propagated from time k-1 to time k

$$\underbrace{\begin{bmatrix} \times & \times & \times \\ 0 & P_{k+1|k}^{-\frac{1}{2}} & P_{k+1|k}^{-\frac{1}{2}}\hat{\mathbf{x}}_{k+1|k} \\ 0 & 0 & \vdots \end{bmatrix}}_{\text{post-array}} \Leftarrow Q_k^T \cdot \underbrace{\begin{bmatrix} P_{k|k-1}^{-\frac{1}{2}} & 0 & P_{k|k-1}^{-\frac{1}{2}}\hat{\mathbf{x}}_{k|k-1} \\ A_k & -I & -B_k u_k \\ C_k & 0 & y_k - D_k u_k \end{bmatrix}}_{\text{pre-array}}$$

$$\hat{\mathbf{x}}_{k+1|k} \Leftarrow (P_{k+1|k}^{-\frac{1}{2}})^{-1} \cdot (P_{k+1|k}^{-\frac{1}{2}}\hat{\mathbf{x}}_{k+1|k})$$

← backsubstitution



Kalman Filter Algorithms

For special case of fixed parameter estimation (p.14) 'Square-Root RLS' formulas (Chapter-9) can be derived as a special case of 'Square-Root Kalman Filter' formulas (p.21) :

$$\underbrace{\begin{bmatrix} \times & \times & \times \\ 0 & P_{k+1|k}^{-\frac{1}{2}} & P_{k+1|k}^{-\frac{1}{2}}\hat{\mathbf{w}}_{k+1|k} \\ 0 & 0 & \vdots \end{bmatrix}}_{\text{post-array}} \Leftarrow Q_k^T \cdot \underbrace{\begin{bmatrix} P_{k|k-1}^{-\frac{1}{2}} & 0 & P_{k|k-1}^{-\frac{1}{2}}\hat{\mathbf{w}}_{k|k-1} \\ -V_k^{-\frac{1}{2}} & -V_k^{-\frac{1}{2}} & 0 \\ \mathbf{u}_k^T & 0 & d_k \end{bmatrix}}_{\text{pre-array}}$$

See p.14 and 17

With $V_k = 0$ this leads to :

$$\underbrace{\begin{bmatrix} P_{k+1|k}^{-\frac{1}{2}} & P_{k+1|k}^{-\frac{1}{2}}\hat{\mathbf{w}}_{k+1|k} \\ 0 & \vdots \end{bmatrix}}_{\text{post-array}} \Leftarrow Q_k^T \cdot \underbrace{\begin{bmatrix} P_{k|k-1}^{-\frac{1}{2}} & P_{k|k-1}^{-\frac{1}{2}}\hat{\mathbf{w}}_{k|k-1} \\ \mathbf{u}_k^T & d_k \end{bmatrix}}_{\text{pre-array}}$$

= QRD-RLS algorithm

Kalman Filter Algorithms

- In textbooks, mostly 'Standard (a.k.a. conventional) Kalman Filter' formulas are given, i.e.

Initialization:

$$\begin{aligned} E\{\mathbf{x}_0\} &= \hat{\mathbf{x}}_{0|-1} \\ E\{\underbrace{(\hat{\mathbf{x}}_{0|-1}-\mathbf{x}_0)}_{\mathbf{e}_0}(\hat{\mathbf{x}}_{0|-1}-\mathbf{x}_0)^T\} &= P_{0|-1} = P_{0|-1}^{\frac{1}{2}} P_{0|-1}^{\frac{T}{2}} \end{aligned}$$

For k=0,1,2,…

Step-1: Measurement Update (corresponding to output equation)

$$\boxed{P_{k|k} = P_{k|k-1} - P_{k|k-1}C_k^T(W_k + C_k P_{k|k-1} C_k^T)^{-1} C_k P_{k|k-1}}$$

$$\boxed{\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + P_{k|k} C_k^T W_k^{-1} \cdot (y_k - C_k \hat{\mathbf{x}}_{k|k-1} - D_k u_k)}$$

Step-2: Time Update (corresponding to state equation)

$$\boxed{P_{k+1|k} = A_k P_{k|k} A_k^T + V_k}$$

$$\boxed{\hat{\mathbf{x}}_{k+1|k} = A_k \cdot \hat{\mathbf{x}}_{k|k} + B_k \cdot u_k}$$

read on ➔

Kalman Filter Algorithms

'Standard Kalman Filter' formulas (p.23) are straightforwardly derived from 'Square-Root Kalman Filter' formulas (p.21) :

Core problem is .

$$\begin{bmatrix} \boxed{P_{k|k-1}^{-\frac{1}{2}}} & 0 \\ A_k & -I \\ C_k & 0 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \hat{\mathbf{x}}_{k+1|k} \end{bmatrix} \overset{LS}{=} \begin{bmatrix} \boxed{P_{k|k-1}^{-\frac{1}{2}} \hat{\mathbf{x}}_{k|k-1}} \\ -B_k u_k \\ y_k - D_k u_k \end{bmatrix} .$$

L+1 equations in $\hat{\mathbf{x}}_{k|}$ can be worked into measurement update eq.

L equations in $\hat{\mathbf{x}}_{k+1|k}$ can be worked into state update eq.

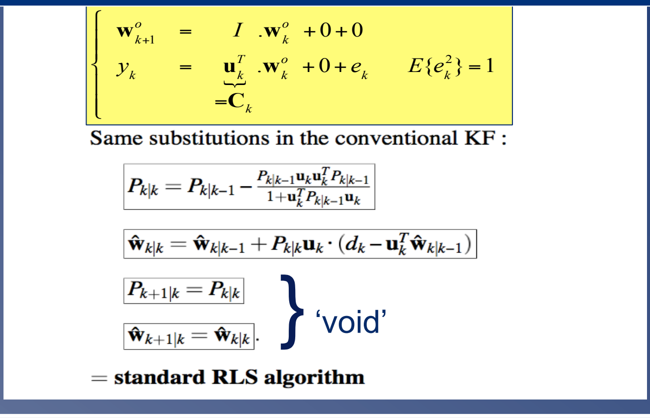
[details omitted]

In infinite precision, algorithms are equivalent

In finite precision, square-root algorithm is preferred

Kalman Filter Algorithms

For special case of fixed parameter estimation (p.14) 'Standard RLS' formulas (Chapter-8) are straightforwardly derived as a special case of 'Standard Kalman Filter' formulas (p.23) :



$$\begin{cases} \mathbf{w}^o_{k+1} = I \,.\mathbf{w}^o_k + 0 + 0 & \\ y_k = \underbrace{\mathbf{u}^T_k}_{=\mathbf{C}_k} .\mathbf{w}^o_k + 0 + e_k & E\{e^2_k\} = 1 \end{cases}$$

Same substitutions in the conventional KF :

$$\boxed{P_{k|k} = P_{k|k-1} - \frac{P_{k|k-1}\mathbf{u}_k\mathbf{u}_k^T P_{k|k-1}}{1+\mathbf{u}_k^T P_{k|k-1}\mathbf{u}_k}}$$

$$\boxed{\hat{\mathbf{w}}_{k|k} = \hat{\mathbf{w}}_{k|k-1} + P_{k|k}\mathbf{u}_k \cdot (d_k - \mathbf{u}_k^T \hat{\mathbf{w}}_{k|k-1})}$$

$$\left.\begin{array}{l} \boxed{P_{k+1|k} = P_{k|k}} \\ \boxed{\hat{\mathbf{w}}_{k+1|k} = \hat{\mathbf{w}}_{k|k}}. \end{array}\right\} \text{'void'}$$

$=$ **standard RLS algorithm**