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 **Wieners Filters & the LMS Algorithm**
- Introduction / General Set-Up
- Applications
- Optimal Filtering: Wiener Filters
- Adaptive Filtering: LMS Algorithm

Chapter-8 **Recursive Least Squares Algorithms**
- Least Squares Estimation
- Recursive Least Squares (RLS)
- Square Root Algorithms

Chapter-9 **Fast Recursive Least Squares Algorithms**

Chapter-10 **Kalman Filters**
- Introduction – Least Squares Parameter Estimation
- Standard Kalman Filter
- Square-Root Kalman Filter

Introduction: Least Squares Parameter Estimation

In Chapter-8, 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 $J_{LS}(\mathbf{w}) = \sum_{l=1}^{k} e_l^2 = \|\mathbf{e}\|_2^2 = \|\mathbf{d} - U\mathbf{w}\|_2^2$

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

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

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 $d_l = \mathbf{u}_l^T . \mathbf{w}^o + e_l$

where $\mathbf{w}^0$ is an unknown parameter vector (='regression coefficients')

and e_l is unknown additive noise

Then…

the aim is to estimate $\mathbf{w}^o$

Least Squares (LS) estimate is (see previous page for definitions of *U* and **d**)

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

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

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

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

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

(*) 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

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

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

- PS1: 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 lower triangular Cholesky factor ('square root'), the Linear MMSE estimator & error covariance matrix are

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

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

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

  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

- PS2: If an initial estimate $\hat{\mathbf{w}}^o$ is available (e.g. from previous observations) with error covariance matrix

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

  where $P^{1/2}$ is the lower triangular Cholesky factor ('square root'), the Linear MMSE estimator & error covariance matrix are

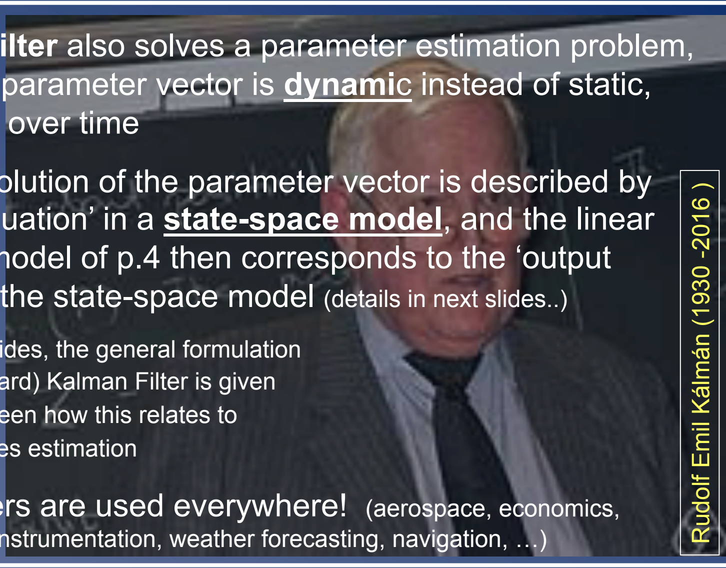
  $$\hat{\mathbf{w}} = (\underbrace{P^{-1} + U^T V^{-1} U}_{U_{\text{EXT}}^T U_{\text{EXT}}})^{-1}.(\underbrace{P^{-1}\hat{\mathbf{w}}^o + U^T V^{-1}\mathbf{d}}_{U_{\text{EXT}}^T \mathbf{d}_{\text{EXT}}}) \qquad E\{(\hat{\mathbf{w}} - \mathbf{w}^o).(\hat{\mathbf{w}} - \mathbf{w}^o)^T\} = (P^{-1} + U^T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

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

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

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

Introduction: Least Squares Parameter Estimation



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

The time-evolution of the parameter vector is described by the 'state equation' in a **<u>state-space model</u>**, 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 formulation of the (Standard) Kalman Filter is given
- In p.16 it is seen how this relates to Least Squares estimation

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

Rudolf Emil Kálmán (1930 -2016)

Standard Kalman Filter

<u>State space model</u>

of a time-varying discrete-time system

state vector — input signal vector — process noise

$$\begin{cases} \mathbf{x}[k+1] &= \mathbf{A}[k].\mathbf{x}[k] + \mathbf{B}[k].\mathbf{u}[k] + \mathbf{v}[k] \\ y[k] &= \mathbf{C}[k].\mathbf{x}[k] + \mathbf{D}[k].\mathbf{u}[k] + w[k] \end{cases}$$

state equation

output equation

output signal — measurement noise

(PS: can also have multiple outputs)

where v[*k*] and w[*k*] are mutually uncorrelated, zero mean, white noises

$$E\left\{\begin{bmatrix} \mathbf{v}[k] \\ w[k] \end{bmatrix} \cdot \begin{bmatrix} \mathbf{v}[l]^H & w[l]^H \end{bmatrix}\right\} = \delta_{kl} \cdot \begin{bmatrix} \mathbf{V}[k] & \mathbf{0} \\ \mathbf{0} & W[k] \end{bmatrix}$$

$$\mathbf{V}[k] = \mathbf{V}[k]^{\frac{1}{2}} . \mathbf{V}[k]^{\frac{T}{2}}$$ = Cholesky/square-root factorization

Standard Kalman Filter

- Example: IIR filter

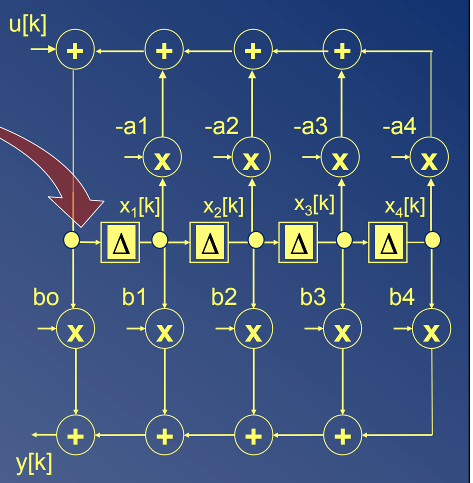
$$y[k] = b_0.u[k] + \ldots + b_4.u[k-L] - a_1.y[k-1] - \ldots - a_4.y[k-L]$$

State space model is

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

$$y[k] = \begin{bmatrix} b_1 - a_1 b_0 & b_2 - a_2 b_0 & b_3 - a_3 b_0 & b_4 - a_4 b_0 \end{bmatrix} . \begin{bmatrix} x_1[k] \\ x_2[k] \\ x_3[k] \\ x_4[k] \end{bmatrix} + b_0.u[k]$$

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



(no process/measurement noise here)

Standard Kalman Filter

State estimation problem

state vector

$$\begin{cases} \mathbf{x}[k+1] = \mathbf{A}[k].\mathbf{x}[k] + \mathbf{B}[k].\mathbf{u}[k] + \mathbf{v}[k] \\ y[k] = \mathbf{C}[k].\mathbf{x}[k] + \mathbf{D}[k].\mathbf{u}[k] + w[k] \end{cases}$$

$$E\{\begin{bmatrix} \mathbf{v}[k]] \\ w[k] \end{bmatrix} . \begin{bmatrix} \mathbf{v}[l]^H & w[l]^H \end{bmatrix}\} = \delta_{kl} . \begin{bmatrix} \mathbf{V}[k] & \mathbf{0} \\ \mathbf{0} & W[k] \end{bmatrix}$$

Given… **A**[k], **B**[k], **C**[k], **D**[k], **V**[k], W[k], k=0,1,2,...
and input/output observations **u**[k],y[k], k=0,1,2,...

Then… estimate the internal states **x**[k], k=0,1,2,...

Standard Kalman Filter

PS: will use shorthand notation $\mathbf{x}_k$, y_k ,.. instead of $\mathbf{x}[k]$, $y[k]$,.. from now on

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 $\hat{\mathbf{x}}_{k|k+n}, n > 0$

Standard Kalman Filter

The 'Standard Kalman Filter' (or 'Conventional Kalman Filter') operation @ time k (k=0,1,2,..) is as follows:

Given a prediction of the state vector @ time k based on previous observations (up to time k-1) $\hat{\mathbf{x}}_{k|k-1}$ with corresponding error covariance matrix $\mathbf{P}_{k|k-1}$

Step-1: Measurement Update

=Compute an improved (filtered) estimate $\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}$ based on 'output equation' @ time k (=observation y[k])

Step-2: Time Update

=Compute a prediction of the next state vector based on 'state equation' $\hat{\mathbf{x}}_{k+1|k}, \mathbf{P}_{k+1|k}$

Standard Kalman Filter

The 'Standard Kalman Filter' formulas are as follows (without proof)

Initalization:

$$E\{\mathbf{x}_0\} = \hat{\mathbf{x}}_{0|-1}$$

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

For k=0,1,2,…

Step-1: Measurement Update

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

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + P_{k|k}C_k^TW_k^{-1}\cdot(y_k - C_k\hat{\mathbf{x}}_{k|k-1} - D_ku_k)$$

← compare to standard RLS! (consider $W_k=1$)

Step-2: Time Update

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

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

← Try to derive this from state equation

Standard Kalman Filter

PS: 'Standard RLS' is a special case of 'Standard KF'

Special case of the state space equations :

$$\mathbf{w}_{k+1} = I\cdot\mathbf{w}_k + 0 + 0$$
$$d_k = \mathbf{u}_k^T\cdot\mathbf{w}_k + 0 + n_k$$

← **Internal state vector is FIR filter coefficients vector, which is assumed to be time-invariant**

with

$$\mathcal{E}\{n_k^2\} = 1.$$

Same substitutions in the conventional KF :

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

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

$$P_{k+1|k} = P_{k|k}$$

$$\hat{\mathbf{w}}_{k+1|k} = \hat{\mathbf{w}}_{k|k}.$$

} 'void'

= **standard RLS algorithm**

Standard Kalman Filter

PS: ‘Standard RLS’ is a special case of ‘Standard KF’

Standard RLS is not numerically stable (see Chapter-8), hence (similarly) the Standard KF is not numerically stable
(i.e. finite precision implementation diverges from infinite precision implementation)

Will therefore again derive an alternative

<u>Square-Root Algorithm</u>

which can be shown to be numerically stable

(i.e. distance between finite precision implementation and infinite precision implementation is bounded)

Square-Root Kalman Filter

The state estimation/prediction @ time k corresponds to a parameter estimation problem in a **<u>linear regression model</u>** (p.4), where the parameter vector contains all previous state vectors…

$$\begin{bmatrix} \hat{\mathbf{x}}_{0|-1} \\ -B_0u_0 \\ y_0-D_0u_0 \\ -B_1u_1 \\ y_1-D_1u_1 \\ \vdots \\ -B_ku_k \\ y_k-D_ku_k \end{bmatrix} = \begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ A_0 & -I & 0 & \cdots & 0 \\ C_0 & 0 & 0 & \cdots & 0 \\ 0 & A_1 & -I & \cdots & 0 \\ 0 & C_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & A_k & -I \\ 0 & 0 & 0 & C_k & 0 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_{k+1} \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{e}_0 \\ \mathbf{v}_0 \\ w_0 \\ \mathbf{v}_1 \\ w_1 \\ \vdots \\ \mathbf{v}_k \\ w_k \end{bmatrix}}_{\mathbf{e}} .$$

Square-Root Kalman Filter

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

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

←compare to p.7

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

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

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 for clarity…
(i.e. stick to previous page)

Square-Root Kalman Filter

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

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

↖explain subscript

Square-Root Kalman Filter

Recursive implementation : is then developed as follows

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

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

→ update = triangularization + backsubstitution

..hence requires only lower-right/lower part ! (explain)

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

Square-Root Kalman Filter

relevant sub-problem is

Propagated from time k-1 to time k

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

i.e.

compare to p.8➔

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

→ update = triangularization + backsubstitution

Square-Root Kalman Filter

Final algorithm is as follows:

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

Propagated from time k to time k+1

Propagated from time k-1 to time k

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

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

←backsubstitution

Square-Root Kalman Filter

Remark QRD- RLS algorithm is a special case of square-root KF

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

See p.16 and 19

With $V_k = 0$ this leads to :

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

=QRD- **RLS**

Standard Kalman Filter (revisited)

Remark : *Conventional Kalman filter*

can be derived from square-root KF equations

Core problem **is** .

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

$n+l$ equations in $\hat{\mathbf{x}}_{k|k}$ **: can be worked into measurement update eq.**

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

[details omitted]