Kiegyenlítő számítások MSc 2023/24

7. előadás

Áttekintés

- Bevezetés
- Kálmán szűrő sajátosságai
    - Rekurzív becslés
    - Mérések optimális kombinálása
    - Rekurzív optimális becslés
    - Közvetett állapot becslés
- Kálmán szűrés lineáris esetben
    - statikus Kálmán szűrő egyenletei
    - jármű navigációs példa

Bevezetés

A Kálmán-szűrés olyan algoritmus, amely valamely lineáris dinamikus rendszerben egzakt következtetést tesz lehetővé, amely a rejtett Markov-modellhez hasonló Bayes-féle modell, azonban a rejtett változók állapottere folytonos, és minden rejtett és megfigyelhető változó (gyakran többváltozós) normális eloszlású.

???

„… nagyon sok nehézségünk volt Dr. Kálmán cikkének megértése során a Dr. Kálmán által a vezérlési problémákhoz használt viszonylag új állapottér megközelítés miatt. **A jelöléseket és fogalmakat, amiket Dr. Kálmán használt, a gyakorló mérnökök igen nehezen értették meg.**” McGee és Schmidt (1985)

Eredeti publikáció

- Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering. 82: 35
- Több mint 40 ezer Google Tudós hivatkozás

Solution of the Wiener problem

Let us now define the principal problem of the paper.

Problem I. *Consider the dynamic model*

$$\mathbf{x}(t+1) = \mathbf{\Phi}(t+1;\, t)\mathbf{x}(t) + \mathbf{u}(t) \quad (16)$$

$$\mathbf{y}(t) = \mathbf{M}(t)\mathbf{x}(t) \quad (17)$$

where $\mathbf{u}(t)$ *is an independent gaussian random process of n-vectors with zero mean,* $\mathbf{x}(t)$ *is an n-vector,* $\mathbf{y}(t)$ *is a p-vector* ($p \leq n$), $\mathbf{\Phi}(t+1;\, t)$, $\mathbf{M}(t)$ *are* $n \times n$, *resp.* $p \times n$, *matrices whose elements are nonrandom functions of time.*

Given the observed values of $\mathbf{y}(t_0)$, ..., $\mathbf{y}(t)$ *find an estimate* $\mathbf{x}^*(t_1|t)$ *of* $\mathbf{x}(t_1)$ *which minimizes the expected loss.* (See Fig. 2, where $\mathbf{\Delta}(t) = \mathbf{I}$.)

This problem includes as a special case the problems of filtering, prediction, and data smoothing mentioned earlier. It includes also the problem of reconstructing all the state variables of a linear dynamic system from noisy observations of some of the state variables ($p < n!$).

From Theorem 2-*a* we know that the solution of Problem I is simply the orthogonal projection of $\mathbf{x}(t_1)$ on the linear manifold $\mathcal{Y}(t)$ generated by the observed random variables. As remarked in the Introduction, this is to be accomplished by means of a linear (not necessarily stationary!) dynamic system of the general form (14). With this in mind, we proceed as follows.

Theorem 3. (*Solution of the Wiener Problem*)

Consider Problem I. The optimal estimate $\mathbf{x}^*(t+1|t)$ *of* $\mathbf{x}(t+1)$ *given* $\mathbf{y}(t_0)$, ..., $\mathbf{y}(t)$ is *generated by the linear dynamic system*

$$\mathbf{x}^*(t+1|t) = \mathbf{\Phi}^*(t+1;\, t)\mathbf{x}^*(t|t-1) + \mathbf{\Delta}^*(t)\mathbf{y}(t) \quad (21)$$

The estimation error is given by

$$\tilde{\mathbf{x}}(t+1|t) = \mathbf{\Phi}^*(t+1;\, t)\,\tilde{\mathbf{x}}(t|t-1) + \mathbf{u}(t) \quad (23)$$

The covariance matrix of the estimation error is

$$\operatorname{cov}\, \tilde{\mathbf{x}}(t|t-1) = E\,\tilde{\mathbf{x}}(t|t-1)\,\tilde{\mathbf{x}}'(t|t-1) = \mathbf{P}^*(t) \quad (26)$$

The expected quadratic loss is

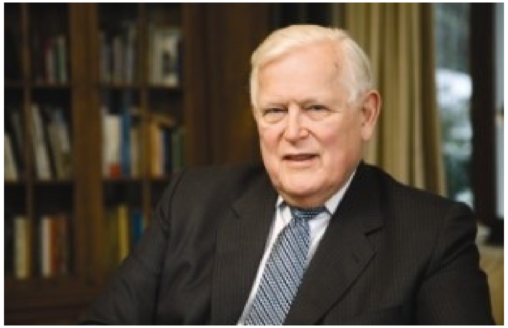
$$\sum_{i=1}^{n} E\tilde{x}_i^2(t\,|\,t-1) = \operatorname{trace} \mathbf{P}^*(t) \quad (27)$$

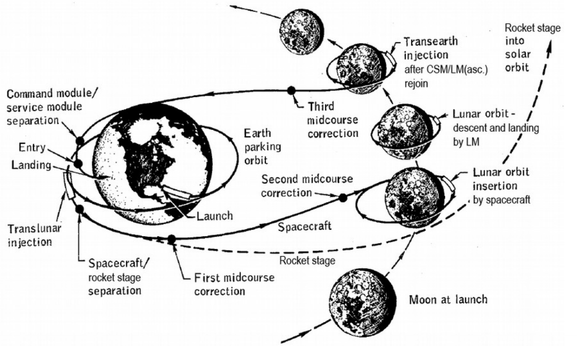
The matrices $\mathbf{\Delta}^*(t)$, $\mathbf{\Phi}^*(t+1;\, t)$, $\mathbf{P}^*(t)$ *are generated by the recursion relations*

$$\left.\begin{aligned}
\mathbf{\Delta}^*(t) &= \mathbf{\Phi}(t+1;\, t)\mathbf{P}^*(t)\mathbf{M}'(t)[\mathbf{M}(t)\mathbf{P}^*(t)\mathbf{M}'(t)]^{-1} && (28)\\
\mathbf{\Phi}^*(t+1;\, t) &= \mathbf{\Phi}(t+1;\, t) - \mathbf{\Delta}^*(t)\mathbf{M}(t) && (29)\\
\mathbf{P}^*(t+1) &= \mathbf{\Phi}^*(t+1;\, t)\mathbf{P}^*(t)\mathbf{\Phi}'(t+1;\, t) + \mathbf{Q}(t) && (30)
\end{aligned}\right\}\; t \geq t_0$$

Története, alkalmazások

- Rudolf Emil Kálmán (1930-2016) magyar származású amerikai villamosmérnök, matematikus publikálja az elméletet (1960)
- Első alkalmazása: ember nélküli amerikai Hold-szonda (1963)
- Navigációs és mérnöki alkalmazások
- Szenzor adatok fúziója (radar, lézerszkenner, kamera, INS,...)
- GNSS vevők, okostelefonok, számítógépek, játékok, stb. elmaradhatatlan része
- Tőzsde előrejelzés, idősor elemzés, függvény approximáció





Apollo fedélzeti számítógép

- Apollo Guidance Computer (AGC)

  2048 szó kapacitású RAM. A "szó" 15 bites adatot jelentett — a RAM csak 3840 bájt volt.

  36 864 szó ROM, vagyis 69 120 bájt.

  legfeljebb 85 000 CPU utasítás másodpercenként

  méretek: 61 cm × 32 cm × 15 cm

  tömeg 30 kg

  áram ellátás: 2.5A áramerősség 28V egyenáram
- Képes volt futtatni a kibővített Kálmán szűrés algoritmusát

A Kálmán szűrő sajátosságai

- Rekurzív becslés
- Mérések optimális kombinálása
- Rekurzív optimális becslés
- Közvetett állapot becslés

Rekurzív becslés

- egyetlen x változóval jellemezhető rendszer állapotát mérjük egymás utáni t_1, t_2, ..., t_n időpontokban, a méréseink: $x_1, x_2, \ldots, x_n$
- az állapot legjobb becslése (Gauss-eloszlást feltételezve) a számtani közép:

$$\mu_n = \frac{1}{n} \sum_{i=1}^{n} x_i .$$

Rekurzív becslés új méréssel

- van egy új mérésünk, x_{n+1}, és a célunk az állapot becslése ezen új mérés felhasználásával
- az állapot legjobb becslése ismét a számtani közép:

$$\mu_{n+1} = \frac{1}{n+1} \sum_{i=1}^{n+1} x_i$$

- felhasználjuk a már kiszámított μ_n átlagot és csak ezt "frissítjük" az új mérés felhasználásával:

$$\mu_{n+1} = \frac{n}{n+1} \left(\frac{1}{n} \sum_{i=1}^{n} x_i + \frac{1}{n} x_{n+1} \right) = \frac{n}{n+1} \mu_n + \frac{1}{n+1} x_{n+1}$$

Rekurzív becslés frissítése

- a rekurzív állapot becslés így írható fel kicsit más alakban:

$$\mu_{n+1} = \frac{n}{n+1}\mu_n + \frac{1}{n+1}x_{n+1} = \mu_n + K\left(x_{n+1} - \mu_n\right)$$

- ahol K az „erősítési tényező”: $K = \frac{1}{n+1}$

- az új információ (innováció): $x_{n+1} - \mu_n$,

Rekurzív átlag becslés

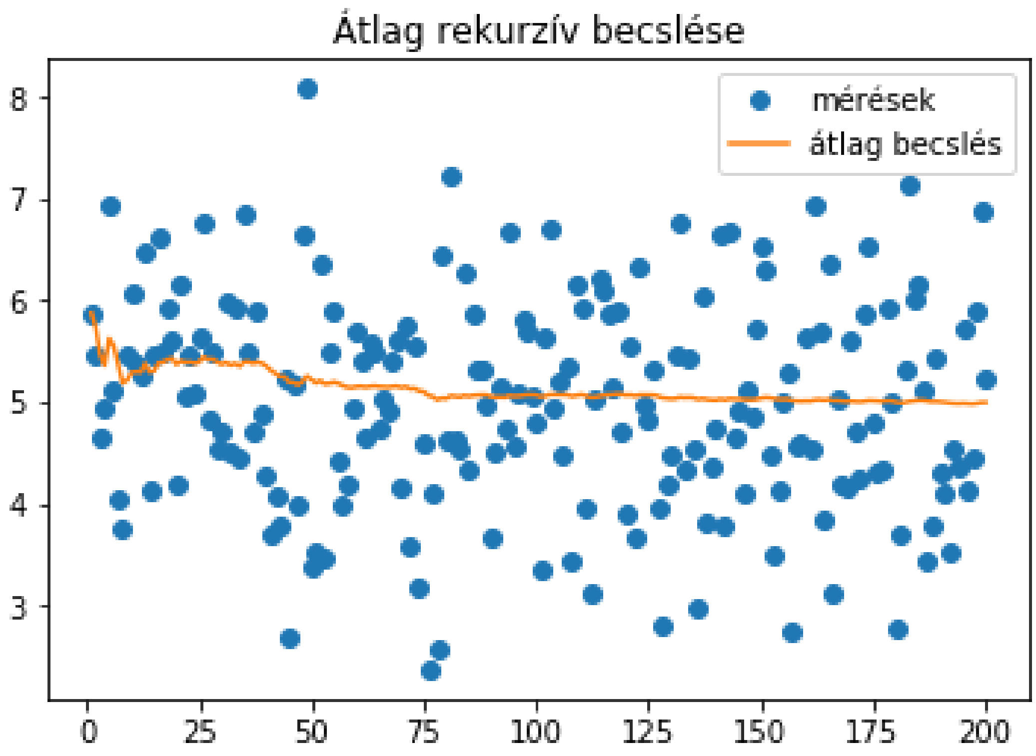
```python
import numpy as np
import matplotlib.pyplot as plt

""" Kálmán-szűrő: rekurzív átlag becslés
"""

npt = 200
x = 5 + np.random.randn(npt)
n = np.arange(1,npt+1)
K = 1.0/(n+1)
mu = np.zeros((npt))
mu[0] = x[0]

for i in range(1,npt):
    mu[i] = mu[i-1] + K[i]*(x[i]-mu[i-1])

plt.plot(n,x,'o',label=u'mérések')
plt.plot(n,mu,label=u'átlag becslés')
plt.legend()
plt.title(u'Átlag rekurzív becslése')
plt.show()
```



Mérések optimális kombinálása

- Tegyük fel, hogy x-et két különböző eszközzel mérjük, az első mérés eredménye x_1, a másodiké x_2
- A méréseink különböző pontosságúak, a szórások σ_1, és σ_2
- A legjobb becslés most a *súlyozott* átlag:

$$\hat{x} = w\,x_1 + (1 - w)\,x_2$$

- Milyen w súly esetén lesz az $\hat{x}$ becslés $\hat{\sigma}$ szórása minimális?

Mérések optimális kombinálása

- Gauss eloszlású v.v-k lineáris kombinációja is Gauss eloszlású, melynek a szórása

$$\hat{\sigma}^2 = w^2\sigma_1^2 + (1-w)^2\sigma_2^2$$

- Az optimális w_{opt} súly ennek a kifejezésnek a minimuma

$$\frac{\partial \hat{\sigma}^2}{\partial w} = 2w_{opt}\sigma_1^2 - 2(1-w_{opt})\sigma_2^2 = 0$$

- Az optimális súly

$$w_{opt} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}$$

Optimális becslés

- x optimális becslése:

$$\hat{x} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} x_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} x_2$$

- Az optimális becslés szórása:

$$\hat{\sigma}^2 = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}$$

- ez a jól ismert súlyozott legkisebb négyzetes becslés

Rekurzív optimális becslés

- Az optimális becslést fogalmazzuk át úgy, hogy az x_1 és szórása becslését frissítjük az új, x_2 mérésből és szórásából számolható korrekcióval:

$$\hat{x} = \hat{x}_2 = \hat{x}_1 + \frac{\hat{\sigma}_1^2}{\hat{\sigma}_1^2 + \sigma_2^2}\left(x_2 - \hat{x}_1\right),$$

$$\hat{\sigma}^2 = \hat{\sigma}_2^2 = \left(1 - \frac{\hat{\sigma}_1^2}{\hat{\sigma}_1^2 + \sigma_2^2}\right)\hat{\sigma}_1^2.$$

Rekurzív optimális becslés

- Elkülönítve ismét a K erősítési tényezőt és az új információt (innováció):

$$\hat{x}_2 = \hat{x}_1 + K\left(x_2 - \hat{x}_1\right),$$

$$\hat{\sigma}_2^2 = \left(1 - K\right)\hat{\sigma}_1^2,$$

$$K = \frac{\hat{\sigma}_1^2}{\hat{\sigma}_1^2 + \sigma_2^2}.$$

Közvetett állapot becslés

- A rendszer $\boldsymbol{x}$ állapota (vektor) helyett csak az állapottól függő $\boldsymbol{z}$ mennyiséget (vektor) tudjuk mérni – („rejtett” állapot, azaz II. kiegyenlítési csoport)
- *Statikus* rendszer: az állapota, x időben nem változik
- *Lineáris* rendszer: az állapot lineáris modellen keresztül kapcsolódik a z mérésekhez

$$\boldsymbol{z} = \boldsymbol{H}\,\boldsymbol{x} + \boldsymbol{r}$$

Közvetett állapot becslés

- A mérési modell mátrixa (alakmátrix): $\boldsymbol{H}$
- Az $\boldsymbol{r}$ mérési zaj Gauss eloszlású, zérus átlagú, $\boldsymbol{R}$ kovariancia mátrixú:

$$\boldsymbol{R} = \mathbb{E}\left[\left(\boldsymbol{r} - \bar{\boldsymbol{r}}\right)\left(\boldsymbol{r} - \bar{\boldsymbol{r}}\right)^T\right]$$

- a várható érték jele: $\mathbb{E}[.]$
- Cél: optimális becslést adni a rendszer állapotára

Statikus Kálmán szűrő egyenletei lineáris esetben

- A levezetés teljesen hasonló a folyamatos csoportképzés közvetítő egyenletekkel esetéhez
  - Detrekői 5.8.2 levezetését követjük
  - A Kálmán szűrés esetében használt jelölést alkalmazzuk
- az eljárás ugyanaz: a meglevő becslést frissítjük az újabb mérésekből számítható korrekcióval
  - rekurzív optimális becslést kapunk

Jelölések megfeleltetése

- alakmátrix: $\boldsymbol{A} \rightarrow \boldsymbol{H}$
- tisztatag - mérési eredmények: $\boldsymbol{l} \rightarrow \boldsymbol{z}$
- mérési javítások: $\boldsymbol{v} \rightarrow -\boldsymbol{r}$
- súlymátrix – kovariancia mátrix: $\boldsymbol{P} \rightarrow \boldsymbol{R}^{-1}$
- paraméterek súlykoefficiens – kovariancia mátrixa: $\boldsymbol{Q}_{xx} \rightarrow \boldsymbol{P}$

Első időpont

- $i = 1$ alkalommal a $\boldsymbol{z}_1$ mérések alapján (n db) meghatározzuk a kiegyenlített $\boldsymbol{x}_1$ paramétereket és azok $\boldsymbol{P}_{11}$ kovariancia mátrixát

normálegyenletek:

$$\boldsymbol{N}_1 \boldsymbol{x}_1 - \boldsymbol{H}_1^T \boldsymbol{R}_{11}^{-1} \boldsymbol{z}_1 = 0$$

$$\boldsymbol{N}_1 = \boldsymbol{H}_1^T \boldsymbol{R}_{11}^{-1} \boldsymbol{H}_1$$

javítási egyenletek:

$$\boldsymbol{r}_1 = \boldsymbol{z}_1 - \boldsymbol{H}_1 \boldsymbol{x}_1$$

becsült paraméterek:

$$\boldsymbol{x}_1 = \boldsymbol{N}_1^{-1} \boldsymbol{H}_1^T \boldsymbol{R}_{11}^{-1} \boldsymbol{z}_1$$

kiegyenlített paraméterek kovariancia mátrixa

$$\boldsymbol{P}_{11} = \boldsymbol{N}_1^{-1} = \left(\boldsymbol{H}_1^T \boldsymbol{R}_{11}^{-1} \boldsymbol{H}_1 \right)^{-1}$$

Második időpont

- $i = 2$ alkalommal a z_2 mérések alapján (s db) meghatározzuk a kiegyenlített $\boldsymbol{x}_2$ paramétereket és azok $\boldsymbol{P}_{22}$ kovariancia mátrixát
- együttes normálegyenlet:

$$\left(\boldsymbol{H}_1^T \boldsymbol{R}_{11}^{-1} \boldsymbol{H}_1 + \boldsymbol{H}_2^T \boldsymbol{R}_{22}^{-1} \boldsymbol{H}_2\right) \boldsymbol{x}_2 - \left(\boldsymbol{H}_1^T \boldsymbol{R}_{11}^{-1} \boldsymbol{z}_1 + \boldsymbol{H}_2^T \boldsymbol{R}_{22}^{-1} \boldsymbol{z}_2\right) = 0$$

- bevezetünk egy új $\boldsymbol{y}$ paraméter vektort (s elemű):

$$\boldsymbol{y} = \boldsymbol{R}_{22}^{-1} \boldsymbol{r}_2 = \boldsymbol{R}_{22}^{-1} \boldsymbol{z}_2 - \boldsymbol{R}_{22}^{-1} \boldsymbol{H}_2 \boldsymbol{x}_2$$

- átalakított normálegyenlet:

$$\left(\boldsymbol{H}_1^T \boldsymbol{R}_{11}^{-1} \boldsymbol{H}_1\right) \boldsymbol{x}_2 - \boldsymbol{H}_2^T \boldsymbol{y} - \boldsymbol{H}_1^T \boldsymbol{R}_{11}^{-1} \boldsymbol{z}_1 = 0$$

Becslés x_2-re

- Megoldandó blokkos egyenletrendszer ($s + n$ méretű):

$$\begin{bmatrix} \boldsymbol{H}_1^T \boldsymbol{R}_{11}^{-1} \boldsymbol{H}_1 & -\boldsymbol{H}_2^T \\ -\boldsymbol{H}_2 & -\boldsymbol{R}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_2 \\ \boldsymbol{y} \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_1^T \boldsymbol{R}_{11}^{-1} \boldsymbol{z}_1 \\ -\boldsymbol{z}_2 \end{bmatrix}$$

- Megoldás Gauss eliminációval a blokkmátrixra

Hipermátrix inverze (lásd: csoportos kiegyenlítés)

közvetlenül számítható Gauss eliminációval a blokkokon:

$$\begin{bmatrix} A & B & E & 0 \\ C & D & 0 & E \end{bmatrix} \rightarrow \begin{bmatrix} E & A^{-1}B & A^{-1} & 0 \\ C & D & 0 & E \end{bmatrix} \rightarrow \begin{bmatrix} E & A^{-1}B & A^{-1} & 0 \\ 0 & D-CA^{-1}B & -CA^{-1} & E \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} E & A^{-1}B & A^{-1} & 0 \\ 0 & E & -\left(D-CA^{-1}B\right)^{-1}CA^{-1} & \left(D-CA^{-1}B\right)^{-1} \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} E & 0 & A^{-1}+A^{-1}B\left(D-CA^{-1}B\right)^{-1}CA^{-1} & -A^{-1}B\left(D-CA^{-1}B\right)^{-1} \\ 0 & E & -\left(D-CA^{-1}B\right)^{-1}CA^{-1} & \left(D-CA^{-1}B\right)^{-1} \end{bmatrix}$$

Becslés x_2-re

- Megoldandó blokkos egyenletrendszer ($s + n$ méretű):

$$\begin{bmatrix} \boldsymbol{H}_1^T \boldsymbol{R}_{11}^{-1} \boldsymbol{H}_1 & -\boldsymbol{H}_2^T \\ -\boldsymbol{H}_2 & -\boldsymbol{R}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_2 \\ \boldsymbol{y} \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_1^T \boldsymbol{R}_{11}^{-1} \boldsymbol{z}_1 \\ -\boldsymbol{z}_2 \end{bmatrix}$$

- Megoldás Gauss-Jordan kiküszöböléssel:

$$\begin{bmatrix} \boldsymbol{x}_2 \\ \boldsymbol{y} \end{bmatrix} = \begin{bmatrix} \boldsymbol{N}_1^{-1} - \boldsymbol{N}_1^{-1} \boldsymbol{H}_2^T \boldsymbol{S} \boldsymbol{H}_2 \boldsymbol{N}_1^{-1} & -\boldsymbol{N}_1^{-1} \boldsymbol{H}_2^T \boldsymbol{S} \\ -\boldsymbol{S} \boldsymbol{H}_2 \boldsymbol{N}_1^{-1} & -\boldsymbol{S} \end{bmatrix} \begin{bmatrix} \boldsymbol{H}_1^T \boldsymbol{R}_{11}^{-1} \boldsymbol{z}_1 \\ -\boldsymbol{z}_2 \end{bmatrix}$$

$$\boldsymbol{S} = \left(\boldsymbol{R}_{22} + \boldsymbol{H}_2 \boldsymbol{N}_1^{-1} \boldsymbol{H}_2^T \right)^{-1}$$

Becslés x_2-re

- Az $\boldsymbol{x}_2$-re vonatkozó megoldás

$$\boldsymbol{x}_2 = \boldsymbol{N}_1^{-1} \boldsymbol{H}_1^T \boldsymbol{R}_{11}^{-1} \boldsymbol{z}_1 - \boldsymbol{K} \boldsymbol{H}_2 \boldsymbol{N}_1^{-1} \boldsymbol{H}_1^T \boldsymbol{R}_{11}^{-1} \boldsymbol{z}_1 + \boldsymbol{K} \boldsymbol{z}_2$$

- felhasználva a $\boldsymbol{K} = \boldsymbol{N}_1^{-1} \boldsymbol{H}_2^T \boldsymbol{S}$ jelölést és azt, hogy

a megoldás: $\boldsymbol{x}_1 = \boldsymbol{N}_1^{-1} \boldsymbol{H}_1^T \boldsymbol{R}_{11}^{-1} \boldsymbol{z}_1$

$$\boldsymbol{x}_2 = \boldsymbol{x}_1 - \boldsymbol{K} \boldsymbol{H}_2 \boldsymbol{x}_1 + \boldsymbol{K} \boldsymbol{z}_2 = \boldsymbol{x}_1 + \boldsymbol{K} \left(\boldsymbol{z}_2 - \boldsymbol{H}_2 \boldsymbol{x}_1 \right)$$

Becslés x_2-re

- Az $\boldsymbol{x}_2$-re vonatkozó megoldás

$$\boldsymbol{x}_2 = \boldsymbol{N}_1^{-1} \boldsymbol{H}_1^T \boldsymbol{R}_{11}^{-1} \boldsymbol{z}_1 - \boldsymbol{K} \boldsymbol{H}_2 \boldsymbol{N}_1^{-1} \boldsymbol{H}_1^T \boldsymbol{R}_{11}^{-1} \boldsymbol{z}_1 + \boldsymbol{K} \boldsymbol{z}_2$$

- felhasználva a $\boldsymbol{K} = \boldsymbol{N}_1^{-1} \boldsymbol{H}_2^T \boldsymbol{S}$ jelölést és azt, hogy

$$\boldsymbol{x}_1 = \boldsymbol{N}_1^{-1} \boldsymbol{H}_1^T \boldsymbol{R}_{11}^{-1} \boldsymbol{z}_1$$

a megoldás:

$$\boldsymbol{x}_2 = \boldsymbol{x}_1 - \boldsymbol{K} \boldsymbol{H}_2 \boldsymbol{x}_1 + \boldsymbol{K} \boldsymbol{z}_2 = \boldsymbol{x}_1 + \boldsymbol{K} \left(\boldsymbol{z}_2 - \boldsymbol{H}_2 \boldsymbol{x}_1 \right)$$

ismerős? $\hat{x}_2 = \hat{x}_1 + K \left(x_2 - \hat{x}_1 \right)$

Erősítési és kovariancia mátrix

- Az $\boldsymbol{S}$ mátrixot visszaírva a $\boldsymbol{K}$ erősítési mátrix:

$$\boldsymbol{K} = \boldsymbol{P}_{11} \boldsymbol{H}_2^T \left(\boldsymbol{R}_{22} + \boldsymbol{H}_2 \boldsymbol{P}_{11} \boldsymbol{H}_2^T \right)^{-1}$$

- A kiegyenlített paraméterek kovariancia mátrixa pedig az invertált hipermátrix bal felső blokkja:

$$\boldsymbol{P}_{22} = \left(\boldsymbol{I} - \boldsymbol{K} \boldsymbol{H}_2 \right) \boldsymbol{P}_{11}$$

- Végeredményben megkaptuk a statikus Kálmán szűrő egyenleteit az 1-es és 2-es epocha között

Erősítési és kovariancia mátrix

- Az $\boldsymbol{S}$ mátrixot visszaírva a $\boldsymbol{K}$ erősítési mátrix:

$$\boldsymbol{K}=\boldsymbol{P}_{11}\boldsymbol{H}_2^T\left(\boldsymbol{R}_{22}+\boldsymbol{H}_2\boldsymbol{P}_{11}\boldsymbol{H}_2^T\right)^{-1}$$

ismerős? $K=\hat{\sigma}_1^2\left(\hat{\sigma}_1^2+\sigma_2^2\right)^{-1}$

- A kiegyenlített paraméterek kovariancia mátrixa pedig az invertált hipermátrix bal felső blokkja:

$$\boldsymbol{P}_{22}=\left(\boldsymbol{I}-\boldsymbol{K}\boldsymbol{H}_2\right)\boldsymbol{P}_{11}$$

- Végeredményben megkaptuk a statikus Kálmán szűrő egyenleteit az 1-es és 2-es epocha között

Dinamikus Kálmán szűrő

- A rendszer $\boldsymbol{x}$ állapota is változik

$$\boldsymbol{x}_k = \boldsymbol{F}\,\boldsymbol{x}_{k-1} + \boldsymbol{B}\,\boldsymbol{u}_{k-1} + \boldsymbol{q}$$

  a $k-1$-edik időpontról a k-adik időpontra, ahol $\boldsymbol{F}$ az állapot átmenet mátrixa, $\boldsymbol{u}$ a rendszer gerjesztése, $\boldsymbol{B}$ a gerjesztés (vezérlés) mátrixa és $\boldsymbol{q}$ a zérus átlagú rendszer zaj, amelynek $\boldsymbol{Q}$ a kovariancia mátrixa

- az állapot átmenet utáni $\boldsymbol{P}^-$ kovariancia mátrixot a kovariancia terjedés adja: $\boldsymbol{P}_k^- = \boldsymbol{F}\,\boldsymbol{P}_{k-1}^-\,\boldsymbol{F}^T + \boldsymbol{Q}$

A dinamikus Kálmán szűrő egyenletei

- A rendszer állapot és kovariancia előrejelzése

$$\hat{\boldsymbol{x}}_k^- = \boldsymbol{F}\,\hat{\boldsymbol{x}}_{k-1} + \boldsymbol{B}\,\boldsymbol{u}_{k-1} + \boldsymbol{q}$$

$$\boldsymbol{P}_k^- = \boldsymbol{F}\,\boldsymbol{P}_{k-1}\,\boldsymbol{F}^T + \boldsymbol{Q}$$

- A mérés utáni állapot és kovariancia frissítés

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

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k\left(\boldsymbol{z}_k - \boldsymbol{H}\,\hat{\boldsymbol{x}}_k^-\right) \qquad \boldsymbol{P}_k = \left(\boldsymbol{I} - \boldsymbol{K}_k\,\boldsymbol{H}\right)\boldsymbol{P}_k^-$$

A Kálmán szűrő működése

előrejelzés

1. állapot előrejelzése

$$\hat{\boldsymbol{x}}_k^- = \boldsymbol{F}\,\hat{\boldsymbol{x}}_{k-1} + \boldsymbol{B}\,\boldsymbol{u}_{k-1} + \boldsymbol{q}$$

2. hiba kovariancia előrejelzése

$$\boldsymbol{P}_k^- = \boldsymbol{F}\,\boldsymbol{P}_{k-1}\,\boldsymbol{F}^T + \boldsymbol{Q}$$

frissítés

1. erősítés számítása

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

2. becslés frissítése a $\boldsymbol{z}_k$ méréssel

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

3. hiba kovariancia frissítése

$$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H})\,\boldsymbol{P}_k^-$$

kiinduló becslések:

$$\hat{\boldsymbol{x}}_{k-1} \quad \boldsymbol{P}_{k-1}$$

K erősítési mátrix szerepe

$$K_k = P_k^- H^T \left(R + H P_k^- H^T \right)^{-1} \qquad \hat{x}_k = \hat{x}_k^- + K_k \left(z_k - H \hat{x}_k^- \right)$$

- Ha a mérési zaj, R nagy lesz, akkor az inverze kicsivé válik és így K_k zérushoz közelít:

  a mérések kis szerepet fognak játszani a frissítésnél

- Ha a mérési zaj, R kicsi lesz, akkor az inverze naggyá válik:

  a mérések nagy szerepet fognak játszani a frissítésnél

A szűrő paraméterezése

- a mérési zaj $\boldsymbol{R}$ kovariancia mátrixa általában előzetesen ismert
- a folyamat zaj $\boldsymbol{Q}$ kovariancia mátrixának felvétele nehezebb: általában nem tudjuk közvetlenül megfigyelni a becslendő állapotot
  - rendszer azonosítás: egy másik Kálmán szűrővel kapjuk meg a $\boldsymbol{Q}, \boldsymbol{R}$ szűrő paramétereket
  - ha $\boldsymbol{Q}, \boldsymbol{R}$ állandó, akkor $\boldsymbol{P}_k$ és $\boldsymbol{K}_k$ gyorsan stabilizálódnak

Kálmán szűrő alkalmazásai

- objektumkövetés, mesterséges látás
- dinamikus helymeghatározás
- gazdaságtan, makroökonómia
- inerciális navigáció
- pályameghatározás
- radar követés
- GNSS
- szeizmológia
- beszédjavítás
- időjárás előrejelzés
- navigációs rendszerek
- 3D modellezés

Példa: járműnavigáció

- egyenes úton haladó jármű
- $\boldsymbol{x}$ állapot vektor: p helyzet és v sebesség
- input vektor $\boldsymbol{u}$: vezérlő gyorsulások (skalár)
- sebesség változás: $v_{k+1} = v_k + \Delta t \cdot u_k + \tilde{v}_k$
- helyzet változás: $p_{k+1} = p_k + \Delta t \cdot v_k + \frac{1}{2} \Delta t^2 u_k + \tilde{p}_k$
- állapot átviteli és mérési egyenlet:

$$x_{k+1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} \Delta t^2/2 \\ \Delta t \end{bmatrix} u_k + w_k \qquad y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} x_k + z_k$$

szimuláció paraméterek

- helyzet bizonytalanság: 10 méter
- véletlenszerű gyorsulások szórása: 0.5 m/s^2
- kezdeti állapot: $\boldsymbol{x} = [\,0, 0]$
- rendszer zaj kovariancia mátrixa (DWNA, diszkrét Wiener fehér zaj gyorsulás modell):

$$Q = \begin{bmatrix} \frac{1}{4}\,\Delta t^4 & \frac{1}{2}\,\Delta t^3 \\ \frac{1}{2}\,\Delta t^3 & \Delta t^2 \end{bmatrix} w_{acc}^2 \,.$$

Python program

```
import numpy as np
import matplotlib.pyplot as plt

def kalman(duration, dt, accelnoise = 0.2):
    ## function kalman(duration, dt) - Kalman filter simulation
    ## duration = length of simulation (seconds)
    ## accelnoise = acceleration noise (m/sec**2)
    ## dt = step size (seconds)

    measnoise  = 3     ## position measurement noise (m)

    F = np.array([[1, dt],[0, 1]])  ## state transition matrix
    H = np.array([[1, 0]])              ## measurement matrix
    x = np.array([[0, 0]]).T    ## initial state vector
    xhat = x     ## initial state estimate

    Q = accelnoise**2 * np.array([[dt**4/4, dt**3/2],[dt**3/2, dt**2]])
    P = Q     ## initial estimation covariance
    R = measnoise**2     ## measurement error covariance

    dimt = int(duration/dt)      ## number of epochs
    pos     = np.zeros(dimt)      ## true position array
    poshat  = np.zeros(dimt)      ## estimated position array
    posmeas = np.zeros(dimt)      ## measured position array
```

```python
    k = 0
    for t in np.arange(0,duration,dt):
        ## Simulate process
        ProcessNoise = accelnoise * np.array([[(dt**2/2)*np.random.normal(), dt*np.random.normal()]]).T
        x = np.dot(F,x) + ProcessNoise
        ## Simulate measurement z at epoch t
        MeasNoise = measnoise * np.random.normal()
        z = np.dot(H,x) + MeasNoise
        ## Innovation: z - H.xh
        Inn = z - np.dot(H,xhat)
        ## Covariance of Innovation: S = H.P.H' + R
        S = np.dot(np.dot(H,P),H.T) + R
        ## Gain matrix K = F.P.H'.inv(S)
        K = np.dot(np.dot(np.dot(F,P),H.T),np.linalg.inv(S))
        ## State estimate: xh = F.xh + K.Inn
        xhat = np.dot(F,xhat) + np.dot(K,Inn)
        ## Covariance of prediction error: P = F.P.F' + Q - K.H.P.F'
        P = np.dot(np.dot(F,P),F.T) + Q - np.dot(np.dot(np.dot(K,H),P),F.T)
        ## Save some parameters in vectors for plotting later
        pos[k]      = x[0]
        posmeas[k] = z
        poshat[k]  = xhat[0]
        k = k + 1

    return pos,posmeas,poshat
```

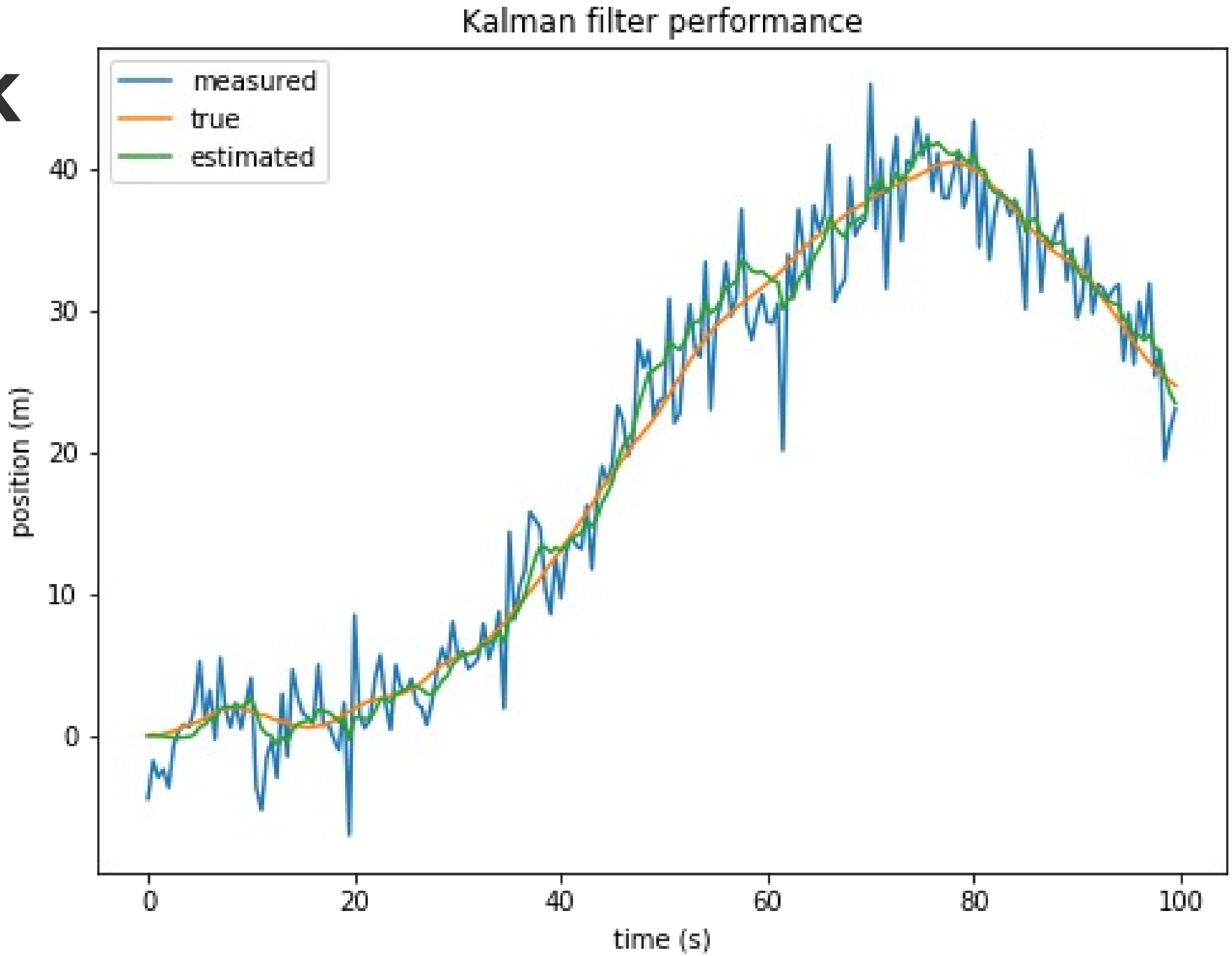
Python program

```
## Simulate

duration = 100
dt = 0.5
p,pm,phat = kalman(duration,dt)

## Plot the results
t = np.arange(0,duration,dt)
plt.figure(figsize=(8,6))
plt.plot(t,pm,label='measured')
plt.plot(t,p,label='true')
plt.plot(t,phat,label='estimated')
plt.legend()
plt.xlabel('time (s)')
plt.ylabel('position (m)')
plt.title('Kalman filter performance')
plt.show()
```

Eredmények


Kalman filter performance
measured
true
estimated
position (m)
time (s)
0
10
20
30
40
0
20
40
60
80
100

Irodalom

- Detrekői 5.8.2: Folyamatos csoportképzés közvetítő egyenletekkel történő kiegyenlítéskor
- Kálmán-szűrés 1.
  Jupyter munkafüzet és Python kód:
  https://github.com/gyulat/Korszeru_matek/blob/master/KF1.ipynb
- Welch G, Bishop G: An introduction to the Kalman Fiter, TR 95-041, UNC, 2006