



7. előadás

Kiegyenlítő számítások MSc

2018/19

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

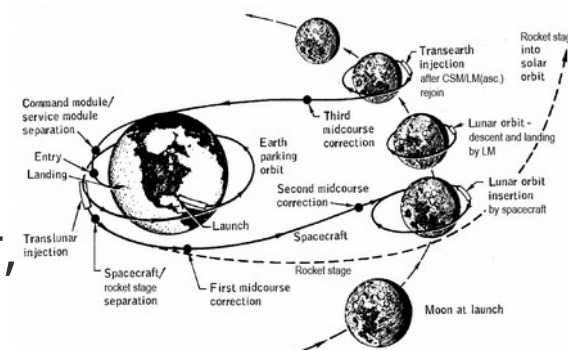
Bevezetés

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.

???

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, lézershkenner, 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: 24"×12.5"×6"

tömeg 30 kg

áram ellátás: 2.5A áramerősség 28V egyenáram

- Képes volt futtatni a Kálmán szűrés algoritmusát

Eredeti publikáció

- Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering. 82: 35
- Több mint 29 ezer Google Scholar hivatkozás

Solution of the Wiener problem

Let us now define the principal problem of the paper.

Problem I. Consider the dynamic model

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

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

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 ($p \leq n$), $\Phi(t+1; t)$, $\mathbf{M}(t)$ are $n \times n$, resp. $p \times n$, matrices whose elements are nonrandom functions of time.

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

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 $\mathbf{y}(t_0), \dots, \mathbf{y}(t)$ is generated by the linear dynamic system

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

The estimation error is given by

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

The covariance matrix of the estimation error is

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

The expected quadratic loss is

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

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

$$\Delta^*(t) = \Phi(t+1; t)\mathbf{P}^*(t)\mathbf{M}'(t)[\mathbf{M}(t)\mathbf{P}^*(t)\mathbf{M}'(t)]^{-1} \quad (28)$$

$$\Phi^*(t+1; t) = \Phi(t+1; t) - \Delta^*(t)\mathbf{M}(t) \quad (29)$$

$$\mathbf{P}^*(t+1) = \Phi^*(t+1; t)\mathbf{P}^*(t)\Phi^*(t+1; t) + \mathbf{Q}(t) \quad (30)$$

$t \geq t_0$

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, \dots, t_n időpontokban, a méréseink: x_1, x_2, \dots, 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:

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

- ahol K az „erősítési tényező”:
$$K = \frac{1}{n+1}$$
- az új információ (innováció):
$$x_{n+1} - \mu_n,$$

Rekurzív átlag becslés

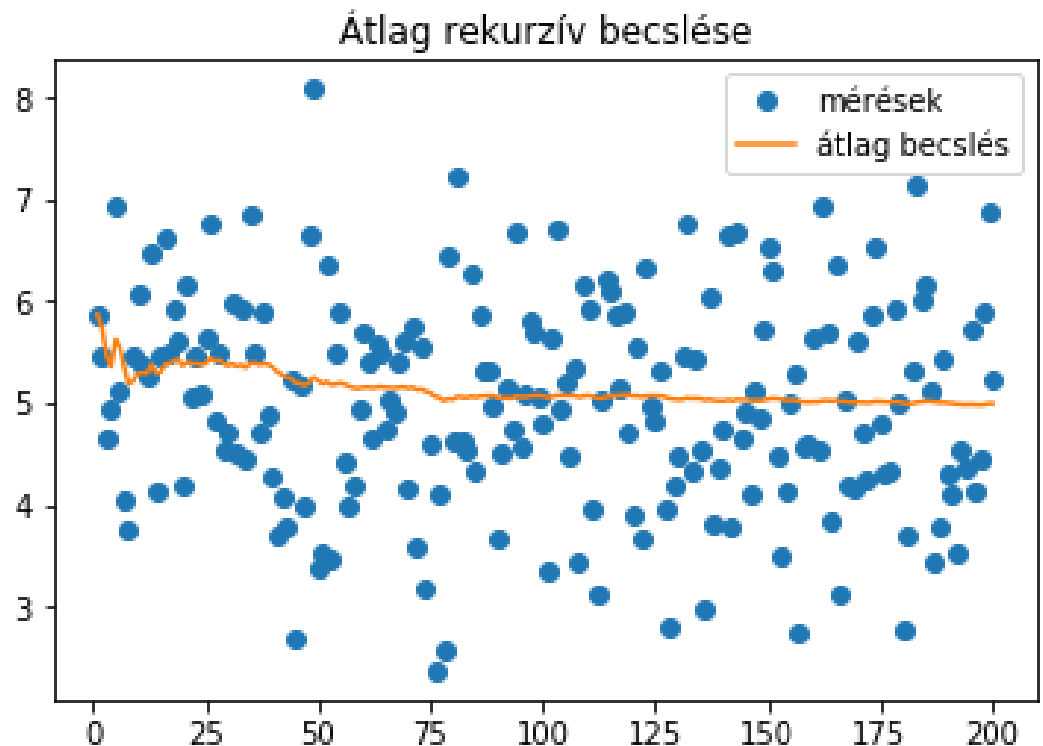
```
import numpy as np
import matplotlib.pyplot as plt

""" 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

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

- 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:

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

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

Rekurzív optimális becslés

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

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

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

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

Közvetett állapot becslés

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

$$\mathbf{z} = \mathbf{H} \mathbf{x} + \mathbf{r}$$

Közvetett állapot becslés

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

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

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

Első időpont

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

- javítási egyenletek:
$$\mathbf{r}_1 = \mathbf{z}_1 - \mathbf{H}_1 \mathbf{x}_1$$

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

- becsült paraméterek:
$$\mathbf{x}_1 = \mathbf{N}_1^{-1} \mathbf{H}_1^T \mathbf{R}_{11}^{-1} \mathbf{z}_1$$

- kiegyenlített paraméterek kovariancia mátrixa

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

Második időpont

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

- együttes normálegyenlet:

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

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

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

- átalakított normálegyenlet:

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

Becslés \mathbf{x}_2 -re

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

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

- Megoldás Gauss-Jordan kiküszöböléssel:

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

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

Becslés \mathbf{x}_2 -re

- Az \mathbf{x}_2 -re vonatkozó megoldás

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

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

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

a megoldás:

$$\mathbf{x}_2 = \mathbf{x}_1 - \mathbf{K} \mathbf{H}_2 \mathbf{x}_1 + \mathbf{K} \mathbf{z}_2 = \mathbf{x}_1 + \mathbf{K} (\mathbf{z}_2 - \mathbf{H}_2 \mathbf{x}_1)$$

Becslés \mathbf{x}_2 -re

- Az \mathbf{x}_2 -re vonatkozó megoldás

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

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

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

a megoldás:

$$\mathbf{x}_2 = \mathbf{x}_1 - \mathbf{K} \mathbf{H}_2 \mathbf{x}_1 + \mathbf{K} \mathbf{z}_2 = \mathbf{x}_1 + \mathbf{K} (\mathbf{z}_2 - \mathbf{H}_2 \mathbf{x}_1)$$

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

Erősítési és kovariancia mátrix

- Az \mathcal{S} mátrixot visszaírva a \mathbf{K} erősítési mátrix:

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

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

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

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

Erősítési és kovariancia mátrix

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

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

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

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

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

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

Dinamikus Kálmán szűrő

- A rendszer \mathbf{x} állapota is változik

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

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

- az állapot átmenet utáni \mathbf{P}^- kovariancia mátrixot a kovariancia terjedés adja:

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

A dinamikus Kálmán szűrő egyenletei

- A rendszer állapot és kovariancia előrejelzése

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

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

- A mérés utáni állapot és kovariancia frissítés

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

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

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

A Kálmán szűrő működése

előrejelzés

1. állapot előrejelzése

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

2. hiba kovariancia előrejelzése

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

kiinduló becslések:

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

frissítés

1. erősítés számítása

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

2. becslés frissítése a \mathbf{z}_k méréssel

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

3. hiba kovariancia frissítése

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

K erősítési mátrix szerepe

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

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

- Ha a mérési zaj, \mathbf{R} nagy lesz, akkor az inverze kicsivé válik és így \mathbf{K}_k zérushoz közelít
a mérések kis szerepet fognak játszani a frissítésnél
- Ha a mérési zaj, \mathbf{R} kicsi lesz, akkor az inverze naggyá válik
a mérések nagy szerepet fognak játszani a frissítésnél

A szűrő paraméterezése

- a mérési zaj R kovariancia mátrixa általában előzetesen ismert
- a folyamat zaj Q kovariancia mátrixának felvétele nehezebb: általában nem tudjuk közvetlenül megfigyelni a becslendő állapotot

rendszer azonosítás: egy másik Kálmán szűrővel kapjuk meg a Q , R szűrő paramétereket

ha Q , R állandó, akkor P_k és 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
- x állapot vektor: p helyzet és v sebesség
- input vektor 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:

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

$$y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} x_k + z_k$$

szimuláció paraméterek

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

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

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 program

```
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
```

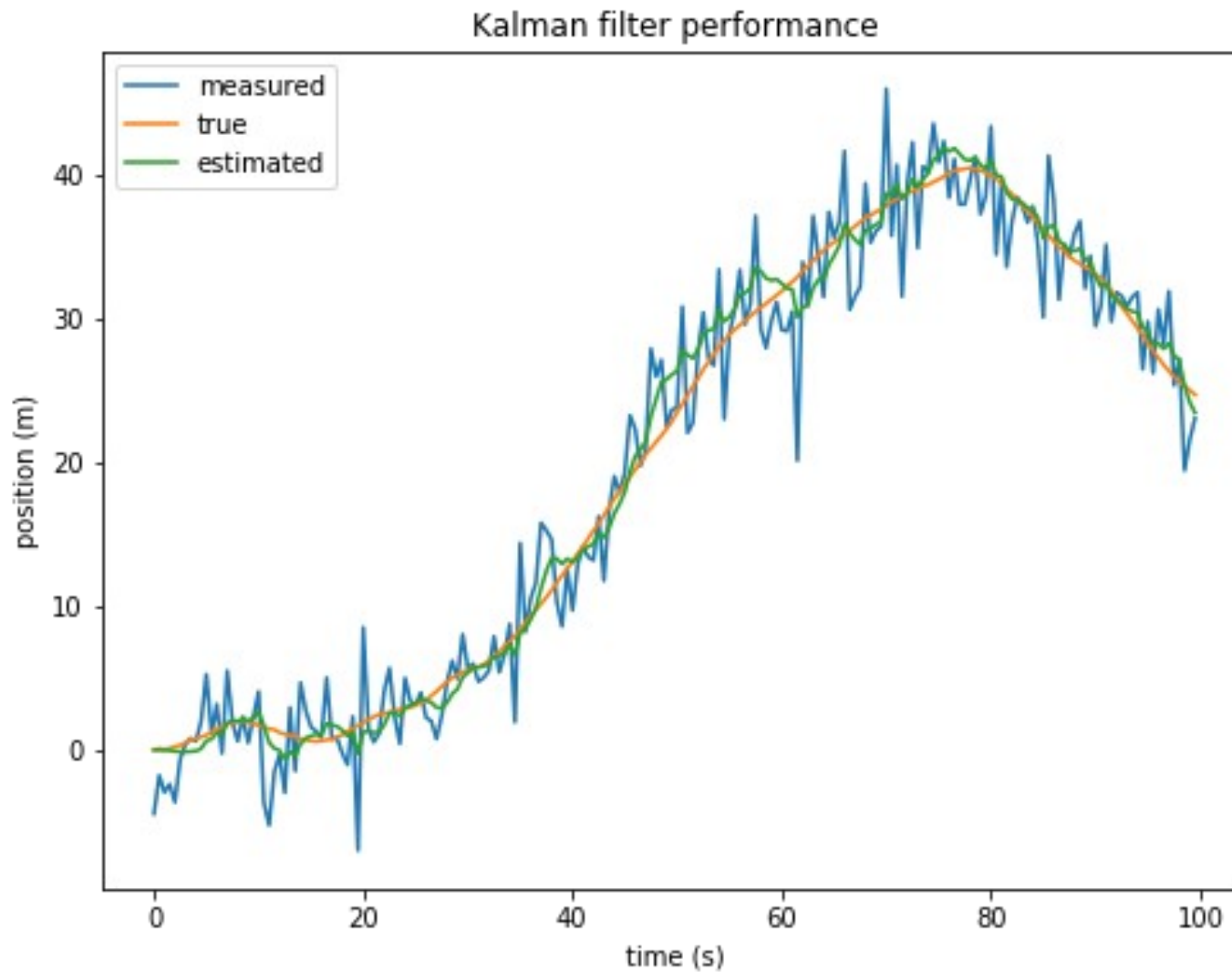
Python program

```
## 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



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 Filter, TR 95-041, UNC, 2006