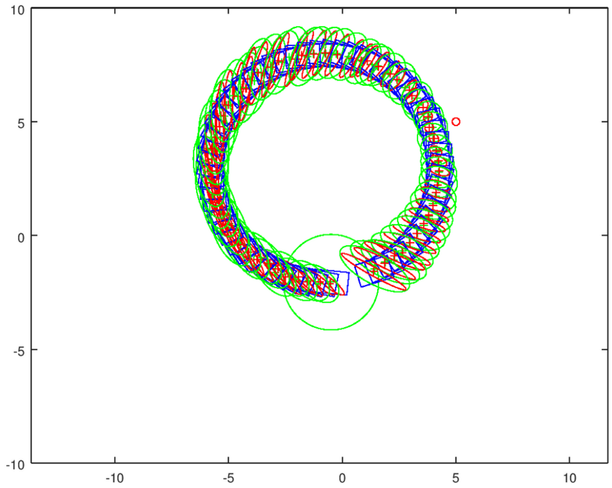
1. Házi feladat: Kálmán szűrés

A feladat egy mozgó mérőkocsiról egy rögzített prizmára vonatkozó távolság és iránymérések feldolgozása kibővített Kálmán szűréssel (EKF).

A `NEPTUNKÓD.dat` állományban megtalálhatóak egy rögzített, $P = (X, Y) = (5, 5)$ koordinátájú helyen található prizmára vonatkozóan egy mozgó mérőkocsiról $\Delta t = 1$ másodpercenként rögzített távolság és szögmérések az alábbi formában:




```
8.8575  -0.8125361
9.2263  -0.6643254
9.6070  -0.5230728
…
```

Az első oszlopban a távolság található (m), a második oszlopban pedig a prizmára mutató iránynak a mérőkocsi tengelyével bezárt szöge van (rad).

Feladat: A mérőkocsi (x, y) pozíciójának és a koordináta-rendszer x tengelyével bezárt ϕ szögének a kiszámítása a 60 mérési epochában kibővített Kálmán szűréssel. Ábrázolja a kocsi becsült helyzeteit az egyes mérési időpontokban és a helyzet becsléshez tartozó kovariancia mátrixból kiszámított hibaellipsziseket. A hibaellipszisek számításhoz lásd pl. Detrekői: Kiegyenlítő számítások tankönyv 4.4.2 pontját.

A kocsi vezérlését a konstans $u = [v_k; \phi_k] = [-0.5; 0.1]$ oszlopvektor írja le, melynek első eleme a kocsi sebessége, a második pedig a jármű kerekének a mérőkocsi tengelyével bezárt szöge (radiánban). A konstans vezérlés miatt a jármű körpályán halad. A kocsi B tengelytávolsága 0.5 m.

A számításhoz ismert a kocsi kiinduló állapota: $x = [0; -2; 0]$. Az állapot vektor első eleme az x, a második az y koordináta (m), a harmadik eleme pedig a kocsi tengelyének az x tengellyel bezárt szöge (mely az y felé mérve pozitív).
A rendszer állapot zaj kovariancia mátrixa: $Q = [0.1, 0, 0; 0, 0.1, 0; 0, 0, 0.0002]$
A mérési vektor (z) elemei a mért d távolság és a mért θ szög: $z = [d; \theta]$
A mérési zaj kovariancia mátrixa: $R = [0.01, 0; 0, 0.00001]$
A kezdeti rendszer állapot kovariancia mátrixa: $P = 10*Q$

A nem lineáris állapot átmeneti egyenlet ($\mathbf{q}_k$ a rendszer állapot zaj):

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \phi_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + \Delta t \cdot v_k \cos(\phi_k + \varphi_k) \\ y_k + \Delta t \cdot v_k \sin(\phi_k + \varphi_k) \\ \phi_k + \Delta t \dfrac{v_k}{B} \sin(\varphi_k) \end{bmatrix} + \boldsymbol{q}_k$$

A nem lineáris mérési egyenlet ($\mathbf{r}_k$ a mérési zaj):

$$\boldsymbol{z}_k = \begin{bmatrix} d_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} \sqrt{(X-x_k)^2 + (Y-y_k)^2} \\ \arctan\left(\dfrac{Y-y_k}{X-x_k}\right) - \phi_k \end{bmatrix} + \boldsymbol{r}_k$$

A szükséges Jacobi mátrixok:

$$\left[\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right] = \begin{bmatrix} 1 & 0 & -\Delta t \cdot v_k \sin(\phi_k + \varphi_k) \\ 0 & 1 & \Delta t \cdot v_k \cos(\phi_k + \varphi_k) \\ 0 & 0 & 1 \end{bmatrix} \quad \text{és} \quad \left[\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right] = \begin{bmatrix} \dfrac{x_k - X}{d_k} & \dfrac{y_k - Y}{d_k} & 0 \\ -\dfrac{y_k - Y}{d_k^2} & \dfrac{x_k - X}{d_k^2} & -1 \end{bmatrix}$$

A Kálmán szűrő lépései minden epochában a következők:

Predikció

$$\hat{\boldsymbol{x}}_k^- = \boldsymbol{f}(\hat{\boldsymbol{x}}_k, u_k) \quad , \quad \boldsymbol{P}_{k+1}^- = \left[\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right] \boldsymbol{P}_k \left[\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right]^T + \boldsymbol{Q}_k$$

Frissítés

$$\hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_{k+1}[\boldsymbol{z}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_k^-)] \quad , \quad \boldsymbol{P}_{k+1} = \boldsymbol{P}_{k+1}^- - \boldsymbol{K}_{k+1} \boldsymbol{S}_{k+1} \boldsymbol{K}_{k+1}^{\boldsymbol{T}} \quad ,$$

ahol

$$\boldsymbol{K}_{k+1} = \boldsymbol{P}_{k+1}^- \left[\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right]^T \boldsymbol{S}_{k+1}^{-1} \quad \text{és} \quad \boldsymbol{S}_{k+1} = \left[\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right] \boldsymbol{P}_{k+1}^- \left[\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right]^T + \boldsymbol{R}_{k+1}$$

A számításhoz segítségképpen megadunk néhány Matlab (Octave) eljárást

```
function [zhat, dh] = meres(x,X)
% x: kocsi helyzete, X: prizma helyzete
d = sqrt((x(1)-X(1))^2 + (x(2)-X(2))^2)
zhat = [ d ; -atan2(X(2)-x(2),X(1)-x(1))-x(3) ]
dh = [-(x(1)-X(1))/d     -(x(2)-X(2))/d     0
      -(x(2)-X(2))/(d^2) (x(1)-X(1))/(d^2) -1 ]
```

```
function [xnew, xnewdot] = atmenet(x, u)

B = 0.5; % tengelytávolság
xnew = x;
xnew(1) = x(1) + u(1)*cos(x(3)+u(2));
xnew(2) = x(2) + u(1)*sin(x(3)+u(2));
xnew(3) = x(3) + u(1)/B*sin(u(2));
xnewdot = [ 1 0 -u(1)*sin(x(3)+u(2))
            0 1  u(1)*cos(x(3)+u(2))
            0 0  1                   ];


function [xpred, Ppred] = predikcio(fuggveny_neve, x, u, P, Q)

[xpred, df] = feval(fuggveny_neve, x, u);
Ppred = df*P*df' + Q;


function [xnew, Pnew] = frissit(xpred, Ppred, z, meresi_fv, X, R)

[zhat, dh] = feval(meresi_fv, xpred, X);
nu = zhat - z;          %% innováció
S = R + dh*Ppred*dh';   %% innováció kovarianciája
K = Ppred*dh'*inv(S);   %% Kálmán mátrix
xnew = xpred + K*nu;    %% új állapot
K*S*K';
Pnew = Ppred - K*S*K';  %% új kovariancia

function x = normalgen(mu, P)
%% normál eloszlású x vektor előállítása
%% mu átlaggal és P kovarianciával

    x = sqrtm(P)*randn(size(mu)) + mu;
```

A kezdeti értékek:

```
u = [ -0.5; 0.1 ];
XP = [5, 5];
Q = [ 0.1 0    0
      0    0.1 0
      0    0     0.0002];
R = [ 0.01 0
      0  0.00001 ];
xhat = [0; -2; 0 ];
P = 10*Q;
```

Epochánkénti számítás:

```
w = normalgen([0;0;0], Q);          % folyamat zaj
x = atmenet(x, u); + w;             % állapot számítása

% z-ben az epochához tartozó mérés van!

% predikció
[xpred, Ppred] = predikcio('atmenet', xhat, u, P, Q);

% frissítés
[xhat, P] = frissit(xpred, Ppred, z, 'meres', XP, R);
```