Cvičení 10

Kalmanův filtr

Jan Přikryl

ČVUT FD

20. dubna 2020

Příklad 1

Navigace

V této úloze vyzkoušíme KF na syntetických datech polohy vozítka, pohybujícího se po modifikované kruhové dráze.

Prvním krokem je vytvoření generátoru měřených pozic dráhy. Budeme postupovat po etapách.

Etapa 1: Čistě kruhová dráha

Začneme čistou kruhovou dráhou o zadaném poloměru r reprezentovanou n měřeními, vozítko opíše přesně jeden kruh. Výstupem generátoru bude matice $(n \times 2)$, obsahující v každém řádku souřadnice $[x, y]$ vozítka.

Příklad 1

Navigace

Pro generování použijeme polární souřadnice a hodnoty φ rozdělíme rovnoměrně na intervalu $\langle 0; 2\pi \rangle$ pomocí funkce `linspace()`:

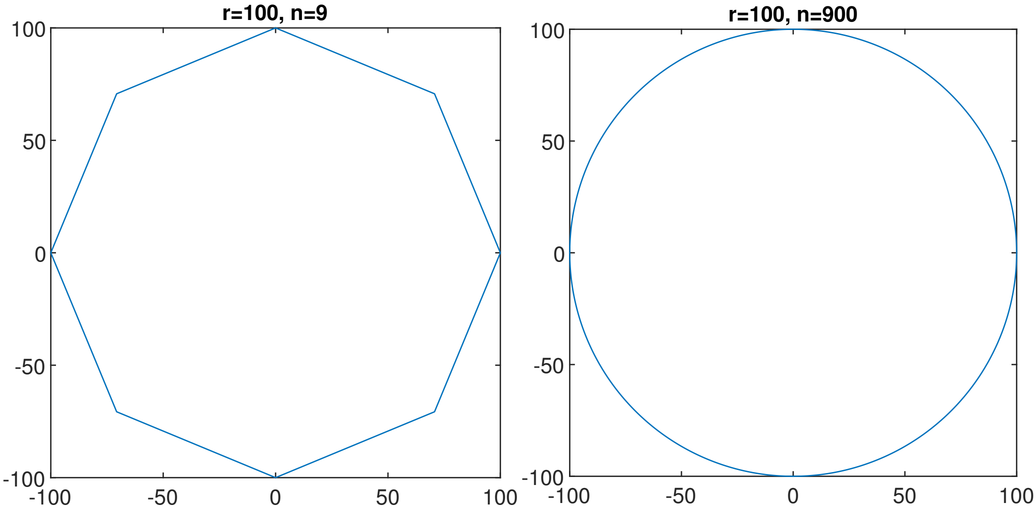
```
function gps=gps_generator0(r,n)
  phi = linspace(....);
  x = ....
  y = ....
  gps = [ .... ];
end
```

Dráhu pak pro zvolené r a n vykreslíme:

```
g=gps_generator0(100,100);
figure(1);
plot(g(:,1), g(:,2));
```

Příklad 1

Kruhová dráha pro $r = 100$ a 9 resp. 900 měřených bodů



Etapa 2: Modulovaná kruhová dráha

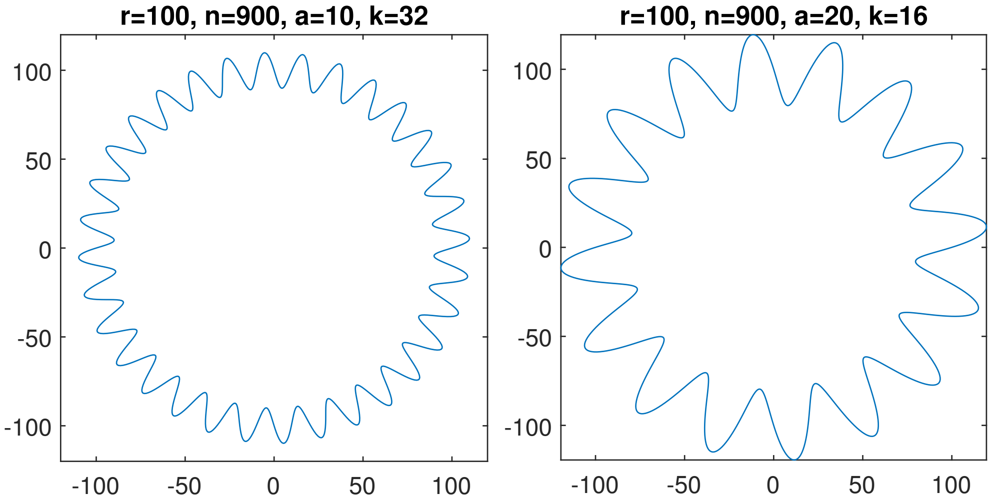
Upravíme gps_generator0() o harmonickou modulaci poloměru dráhy. Místo původního poloměru r budeme nyní používat $r + a \sin k\varphi$. Zbytek zůstává.

```
function gps=gps_generator1(r,n,a,k)
  phi = linspace(....);
  ro = r + .....
  x = ....
  y = ....
  gps = [ .... ];
end
```

Dráhu opět pro zvolené r, n, a a k vykreslíme.

Příklad 1

Modulovaná kruhová dráha pro $r = 100$, 900 měřených bodů a dvě různé modulace



Příklad 1

Simulace měření pozice vozidla na dráze

Etapa 3: Modulovaná kruhová dráha se šumem měření

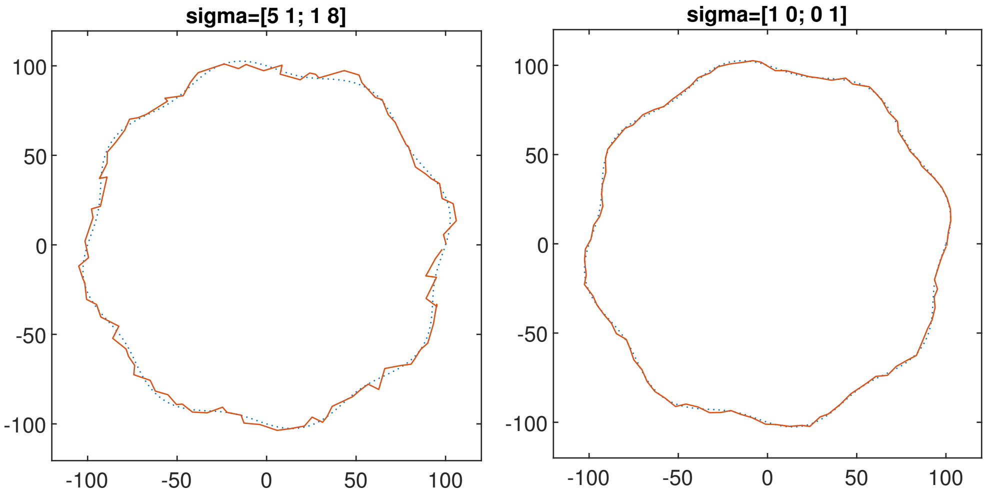
Upravíme gps_generator1() o normálně rozdělený šum pozice se střední hodnotou $\boldsymbol{\mu} = (0, 0)$ a nenulovou kovarianční matici $\boldsymbol{\Sigma}$. Vzorky z multinomiálního rozdělení generujeme v Matlabu funkcí mvnrnd().

```
function gps=gps_generator2(r,n,a,k,sigma)
  phi = linspace(....);
  ro = r + .....
  err = mvnrnd(....);
  x = ....
  y = ....
  gps = [ .... ] + err;
end
```

Dráhu opět pro zvolené r, n, a a k vykreslíme.

Příklad 1

Simulace nepřesných měření pozice pro $r = 100$, $a = 4$, $k = 8$ a 100 měřených bodů



Příklad 1

Kalmanův filtr – Model systému

Pro modelování pohybu vozidla použijeme napřed zcela primitivní a nepřesný model, který předpokládá, že vozítko se pohybuje rovnoměrným přímočarým pohybem, a jeho pozice v $k+1$ kroku je dána vztahem

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \Delta\mathbf{p}.$$

Hodnota $\Delta\mathbf{p}$ je pro rovnoměrný přímočarý pohyb konstantní.

Pohybujeme se ve 2D, uvažujeme tedy

- poloha je $\mathbf{p}_k = (x_k, y_k)^\mathsf{T}$
- změna polohy je $\Delta\mathbf{p}_k = (a_k, b_k)^\mathsf{T}$ a předpokládáme, že se v čase nemění

Příklad 1

Kalmanův filtr – Model systému

Sestavíme rovnice vývoje stavu

$$
\begin{aligned}
x_{k+1} &= x_k + a_k \\
y_{k+1} &= y_k + b_k \\
a_{k+1} &= a_k \\
b_{k+1} &= b_k
\end{aligned}
$$

Stav je tedy $\mathbf{x}_k = (x_k, y_k, a_k, b_k)^\mathsf{T}$.

Měříme pouze pozici, tj. souřadnice x_k a y_k.

Příklad 1

Kalmanův filtr – Model systému

Přepíšeme do maticového zápisu

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k \\ \mathbf{z}_k &= \mathbf{C}\mathbf{x}_k\end{aligned}$$

s nulovými pomocnými maticemi $\mathbf{B}$ a $\mathbf{D}$

$$\mathbf{A} = \begin{bmatrix}1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1\end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix}0 \\ 0 \\ 0 \\ 0\end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix}1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0\end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix}0 \\ 0\end{bmatrix}.$$

Příklad 1

Kalmanův filtr – Skript v Matlabu

Nastavíme základní parametry:

```
radius = 1000; % r = 1000 m
num_coords = 200; % stačí 200 hodnot
a = 4;
k = 8;
R_true = diag([10,10]); % simulované R
```

Vygenerujeme referenční souřadnice a souřadnice měřených pozic zatížené chybou s kovarianci $\mathbf{R}_{\mathrm{true}}$:

```
coord0 = gps_generator1(radius, num_coords, a, k);
coords = gps_generator2(radius, num_coords, a, k, R_true);
```

Příklad 1

Kalmanův filtr – Skript v Matlabu

Zadáme matice soustavy, budeme potřebovat i **B** a **D**:

```
A = [ 1 0 1 0; 0 1 0 1; 0 0 1 0; 0 0 0 1 ];
B = [ 0; 0; 0; 0];
C = [ 1 0 0 0; 0 1 0 0];
D = [ 0; 0 ];
```

Zadáme počáteční podmínky:

```
% Více důvěřujeme měřeným hodnotám
Q = diag([100,100,100,100]); % šum (chyba) modelu
R = diag([10,10]); % šum (chyba) měření
% Počáteční šum procesu a stav
P0 = diag([10, 10, 10, 10]); % bez měření roste o Q, jinak klesá
x0 = [ coords(1,:)'; 0; 0]; % stav je sloupcový vektor!
```

Příklad 1

Kalmanův filtr – Skript v Matlabu

Celkem provedeme `num_coords` iterací. Alokujeme si prostor, kam uložíme predikované souřadnice $\hat{\mathbf{p}}$ a filtrované (měřením korigované) souřadnice $\hat{\mathbf{p}}^*$

```
pred_history = zeros(4, num_coords);
filt_history = zeros(4, num_coords);
```

Nyní nás čeká vytvoření klíčové části skriptu, smyčky, která prochází „změřené“ hodnoty, uložené v proměnné `coords`, pro každou z nich zavolá externí funkci `kalman()` a aktualizuje stav systému a kovarianci šumu procesu. Ve smyčce si zároveň budeme pro pozdější vykreslení ukládat hodnoty predikovaných a filtrovaných souřadnic do matic `pred_history` a `filt_history`.

Podívejme se ale nejprve na rozhraní funkce `kalman()`,

Příklad 1

Kalmanův filtr – Volání `kalman()`

Funkci, implementující Kalmanův filtr, voláme jako

```
[x0f,x1p,Pf,P1] = kalman(x0,u0,zm,A,B,C,D,P,Q,R)
```

kde vstupní parametry jsou

- `x0` . . . stav $\mathbf{x}_k$
- `u0` . . . vstup $\mathbf{u}_k$, v našem případě bude $\mathbf{u}_k = u_k = 0$
- `zm` . . . měřené $\mathbf{z}_k$
- `A`, `B`, `C`, `D` . . . matice stavového modelu
- `P` . . . apriorní odhad kovariance stavu $\hat{\mathbf{x}}_k$
- `Q`, `R` . . . kovariance šumu model a měření

Příklad 1

Kalmanův filtr – Volání `kalman()`

Funkci, implementující Kalmanův filtr, voláme jako

```
[x0f,x1p,Pf,P1] = kalman(x0,u0,zm,A,B,C,D,P,Q,R)
```

kde výstupní parametry jsou

- `x0f` . . . filtrovaný stav $\mathbf{x}_k^*$
- `x1p` . . . predikovaný stav $\hat{\mathbf{x}}_{k+1}$, založený na $\mathbf{x}_k^*$
- `Pf` . . . aposteriorní odhad kovariance stavu $\hat{\mathbf{x}}_k$
- `P1` . . . apriorní odhad kovariance stavu $\hat{\mathbf{x}}_{k+1}$

Příklad 1

Kalmanův filtr – Skript v Matlabu

Doplňte kostru základní smyčky:

```
for k=1:num_coords
  ym = coords(k,:)'; % simulované měření, sloupcov vektor
  % aktualizace stavu modelu na základě starého stavu a měření
  [x0f,x1p,P0f,P1p]=kalman(x0,0,ym,A,B,C,D,P0,Q,R);
  % uložíme pro vykreslení
  pred_history(:,k) = ....; % predikovaný příští stav
  filt_history(:,k) = ....; % filtrovaný současný stav
  % příprava pro další krok
  x0 = ....; % odhad budoucího stavu
  P0 = ....; % odhad nepřesnosti budoucího stavu
end
```

Vykreslíme

```
figure(1);
plot(coord0(:,1),coord0(:,2),'-m', ...
     coords(:,1),coords(:,2),'.b', ...
     filt_history(1,:),filt_history(2,:),'-r', ...
     pred_history(1,:),pred_history(2,:),'-g');
legend('True␣position', 'Measured␣position', 'Filtered',
'Predicted');
xlabel('x_m');
ylabel('y_m');
```

Příklad 1

Obrázek části filtrované trajektorie