

Drugi domaći zadatak iz predmeta Stohastički sistemi i estimacija

Danica Bandović 2018/0018

1.Zadatak

1.Određivanje izraza za $l(\phi)$, $l'(\phi)$, $l''(\phi)$:

$$x[n] = A \cos(2\pi f_0 n + \phi) + w[n]$$

Funkcija gustine verovatnoće opservacija:

$$p(x, \phi) = \frac{1}{(2\pi\sigma^2)^{N/2}} \cdot \exp \left(-\frac{1}{2\sigma^2} \sum_{i=0}^{N-1} [x[n] - A \cos(2\pi f_0 n + \phi)]^2 \right)$$

Log-verodostojnost:

$$\begin{aligned} l(\phi) &= \ln(p(x, \phi)) \\ &= \ln \left(\frac{1}{(2\pi\sigma^2)^{N/2}} \cdot \exp \left(-\frac{1}{2\sigma^2} \sum_{i=0}^{N-1} [x[n] - A \cos(2\pi f_0 n + \phi)]^2 \right) \right) \end{aligned}$$

$$l(\phi) = -\frac{N}{2} \ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=0}^{N-1} [x[n] - A \cos(2\pi f_0 n + \phi)]^2$$

Prvi izvod $l'(\phi)$:

$$\begin{aligned} l'(\phi) &= -\frac{A}{2\sigma^2} \sum_{i=0}^{N-1} 2[x[n] - A \cos(2\pi f_0 n + \phi)] \sin(2\pi f_0 n + \phi) \\ &= -\frac{A}{\sigma^2} \sum_{i=0}^{N-1} [x[n] \sin(2\pi f_0 n + \phi) - A \cos(2\pi f_0 n + \phi) \sin(2\pi f_0 n + \phi)] \end{aligned}$$

$$l'(\phi) = -\frac{A}{\sigma^2} \sum_{i=0}^{N-1} \left[x[n] \sin(2\pi f_0 n + \phi) - \frac{A}{2} \sin(4\pi f_0 n + 2\phi) \right]$$

Drugi izvod $l''(\phi)$:

$$l''(\phi) = -\frac{A}{\sigma^2} \sum_{i=0}^{N-1} [x[n] \cos(2\pi f_0 n + \phi) - A \cos(4\pi f_0 n + 2\phi)]$$

```
close all
clc
clear

N = 10;
A = 1;
f = 0.2;
sgm_kvadr = 1;
tacna_vrednost = 1;

%generisanje vrednosti opservacija
n=(0:N-1);
x = A.*cos(2*pi.*f.*n + tacna_vrednost) + sgm_kvadr.*randn(1,N);
```

tacka prva

```
%log-verodostojnost
phi_osa = -3:0.1:5;
suma = zeros(size(phi_osa));
for i=0:N - 1
    suma = suma + (x(1, i + 1) - A*cos(2*pi*f*i + phi_osa)).^2;
end
l = -N/2*log(2*pi*sgm_kvadr) - 1/(2*sgm_kvadr)*suma;
figure(1)
plot(phi_osa, l);
xlabel('fi');
ylabel('l(fi)');
title('Verodostojnost');
grid on

% prvi izvod log-verodostojnosti
suma = zeros(size(phi_osa));
for i=1:N
    suma = suma + x(1,i)*sin(2*pi*f*(i-1)+phi_osa)-A/2*sin(4*pi*f*(i-1)+2*phi_osa);
end
dl = -A/(2*sgm_kvadr)*2*suma;

figure(2)
plot(phi_osa,dl); grid on
title('Prvi izvod log-verodostojnosti l''(\phi), N = 10, \sigma = 1');
xlabel('\phi');

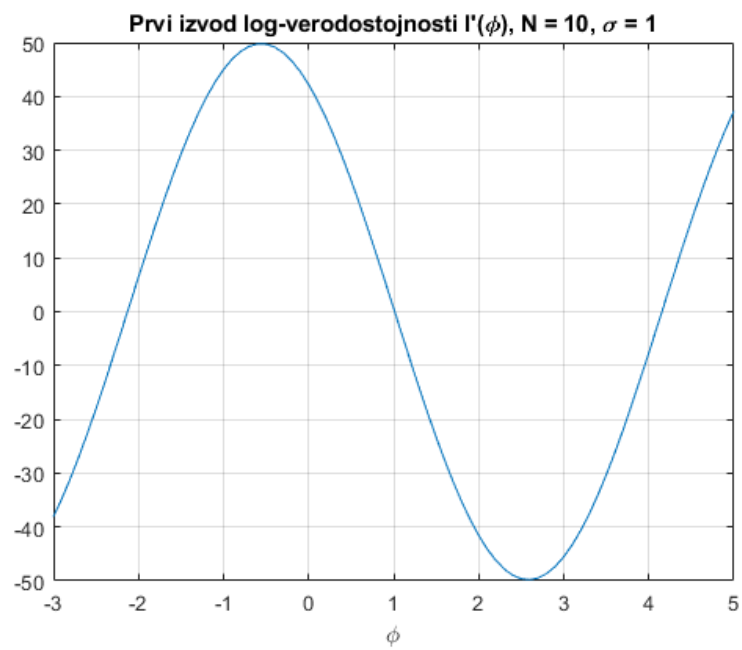
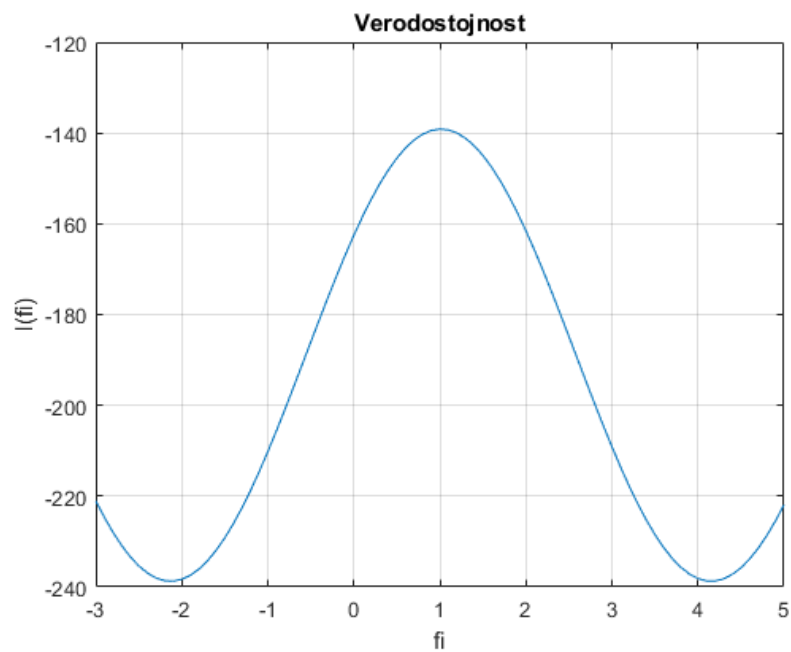
%drugi izvod log-verodostojnosti
suma = zeros(size(phi_osa));
for i=1:N
```

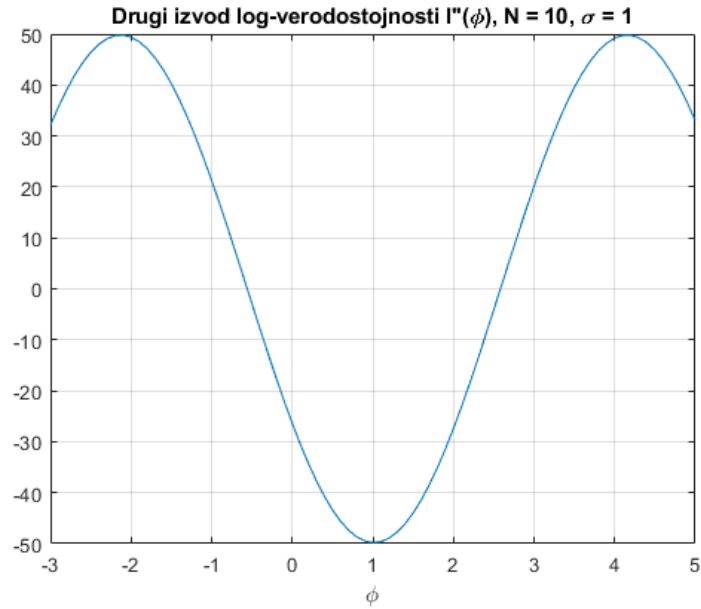
```

suma = suma + x(1,i)*cos(2*pi*f*(i-1)+phi_osa)-A*cos(4*pi*f*(i-1)+2*phi_osa);
end
ddl = -A/(2*sgm_kvadr)*2*suma;

figure(3)
plot(phi_osa,ddl); grid on
title('Drugi izvod log-verodostojnosti l''''(\phi), N = 10, \sigma = 1');
xlabel('\phi');

```





2. Grafici na kojima su vrednosti procene $\hat{\phi}_k$ za različite realizacije i različite početne uslove, u zavisnosti od iteracije, kao i kod u MATLAB-u koji generiše date grafike:

tacka dva

```
%pocetni uslovi
pu=[pi/2, 2*pi/3, 3*pi/2];
%pu=[0.5 1 1.5];
N1=100;
n1=0:N1-1;

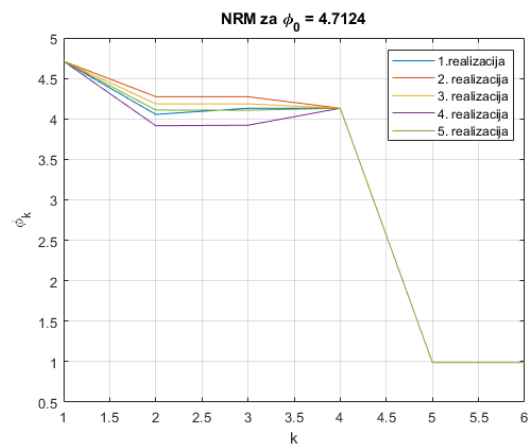
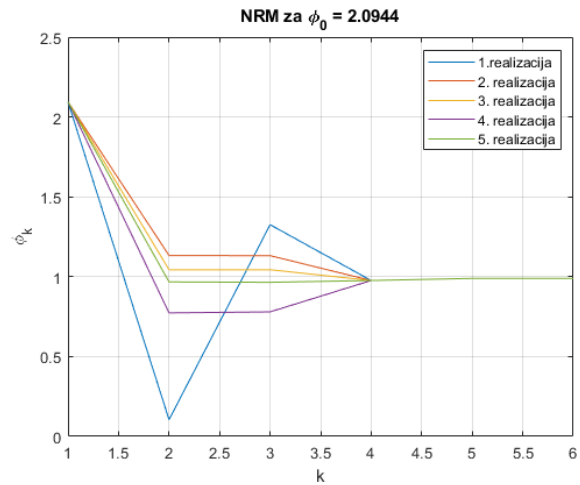
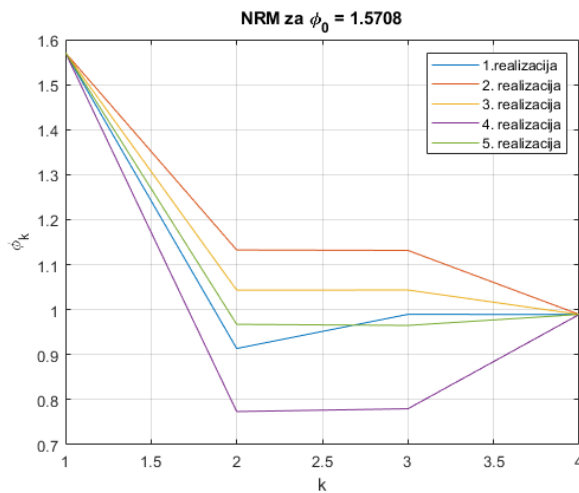
%generisanje x
x1=zeros(N1,5);
for i=1:5
    x1(:,i) = randn(N1,1)*sgm_kvadr + A*cos(2*pi*f*n1+tacna_vrednost)';
end

%Newton-Raphsonova metoda
for kor=1:3
    figure()
    phi_k=pu(kor);
    procena(1)=pu(kor);
    for j=1:5
        for i=2:20
            suma = 0;
            for k=1:N1
                suma = suma + x1(k,j)*sin(2*pi*f*(k-1) + phi_k) - A/2*sin(4*pi*f*(k-1) +
2*pi*phi_k);
            end
            d11 = -A/(sgm_kvadr)^2*suma;
            suma = 0;
            for k=1:N1
```

```

        suma = suma + x1(k, j)*cos(2*pi*f*(k-1) + phi_k) - A*cos(4*pi*f*(k-1) + 2*phi_k);
    end
    dd11 = -A/(sgm_kvadr)^2*suma;
    phi_k1 = phi_k - d11/dd11;
    phi_k1=mod(phi_k1,2*pi);
    procena(i)=phi_k1;
    if (abs(procena(i)-phi_k)<0.01)
        break;
    end
    phi_k=procena(i);
end
plot(procena);hold on;
end
xlabel('k');
ylabel('\phi_k');
title(['NRM za \phi_0 = ', num2str(pu(kor))] );
grid on
legend('1.realizacija','2. realizacija', '3. realizacija', '4. realizacija', '5.
realizacija');
hold off
end

```



Pomoću Newton-Raphsonove metode traži se maksimum funkcije verodostojnosti, odnosno log-verodostojnosti. Tačnije, ono što algoritam radi jeste traženje nule prvog izvoda log-verodostojnosti. Može se desiti da NRM konvergira ka vrednosti 1 ili 4.14 ($\pi + 1$). Ukoliko za početni uslov uzmemo neku od vrednosti $\pi/2$, $2\pi/3$, $3\pi/2$ NRM konvergira ka istoj vrednosti za sve realizacije, dok je primećeno da za neke druge početne uslove (1.5 na primer) NRM ne konvergira ka istim vrednostima za sve realizacije. NRM je pri simuliranju konvergirao nakon 4 iteracije, ali se ponekad desilo da konvergira nakon 5 iteracija, i broj iteracija posle koliko NRM konvergira nije zavisio od početne procene.

3. Izvođenje izraza za Fisherovu informaciju, koristeći aproksimaciju:

$$\begin{aligned}
 I(\phi) &= -E \frac{\partial^2 \ln(p(x, \phi))}{\partial \phi^2} = -E\{l''(\phi)\} \\
 I(\phi) &= -E \left\{ -\frac{A}{\sigma^2} \sum_{i=0}^{N-1} [x[n] \cos(2\pi f_0 n + \phi) - A \cos(4\pi f_0 n + 2\phi)] \right\} \\
 &= -E \left\{ -\frac{A}{\sigma^2} \sum_{i=0}^{N-1} [A \cos^2(2\pi f_0 n + \phi) - A \cos(4\pi f_0 n + 2\phi)] \right\} \\
 &= \frac{A^2}{\sigma^2} E \left\{ \sum_{i=0}^{N-1} \left[\frac{1}{2} + \frac{1}{2} A \cos(4\pi f_0 n + 2\phi) - A \cos(4\pi f_0 n + 2\phi) \right] \right\}
 \end{aligned}$$

$$I(\phi) = \frac{A^2}{\sigma^2} \sum_{i=0}^{N-1} \frac{1}{2} = \frac{A^2 N}{2\sigma^2}$$

tacka treca

```

phi0 = 0.5;
sigma = sqrt(0.1);
N = 10;
Nr = 1000;
n=0:N-1;
estimacija = zeros(1, Nr);
for j=1:Nr
    x = zeros(1, N);

```

```

for i = 1:N
    x(1, i) = A*cos(2*pi*f*(i-1) + tacna_vrednost) + randn*sigma;
end
phi_k1 = phi0;
maxiter = 5;
l1(j,:) = -(N/2)*log(2*pi*sigma^2)-(1/(2*sigma^2))*sum((x-A*cos(2*pi*f*n+phi_k1)).^2);
while(maxiter > 0)
    maxiter = maxiter - 1;
    phi_k = phi_k1;
    suma = 0;
    for i=1:N
        suma = suma + x(1,i)*sin(2*pi*f*(i-1) + phi_k) - A/2*sin(4*pi*f*(i-1) + 2*phi_k);
    end
    dl1 = -A/(sigma)^2*suma;
    suma = 0;
    for i=1:N
        suma = suma + x(1, i)*cos(2*pi*f*(i-1) + phi_k) - A*cos(4*pi*f*(i-1) + 2*phi_k);
    end
    ddl1 = -A/(sigma)^2*suma;
    phi_k1 = phi_k - dl1/ddl1;
    %phi_k1 = phi_k1 - 2*pi*floor(phierr/2/pi);
    phi_k1=mod(phi_k1,2*pi);
end
estimacija(1, j) = phi_k1;
end
figure();
histogram(estimacija, 25, 'Normalization', 'pdf');
hold all
I = A^2*N/(sigma^2*2);
varijansa = 1/I;
x = xlim;
x = x(1):0.01:x(2);
phi_a = 1/(sqrt(2*pi*varijansa))*exp(-(1/(2*varijansa)).*(x - tacna_vrednost).^2);
plot(x, phi_a);
legend('EMV pomocu NRM','asimptotska fgv EMV');
title('\sigma=0.01,N=10');
hold off

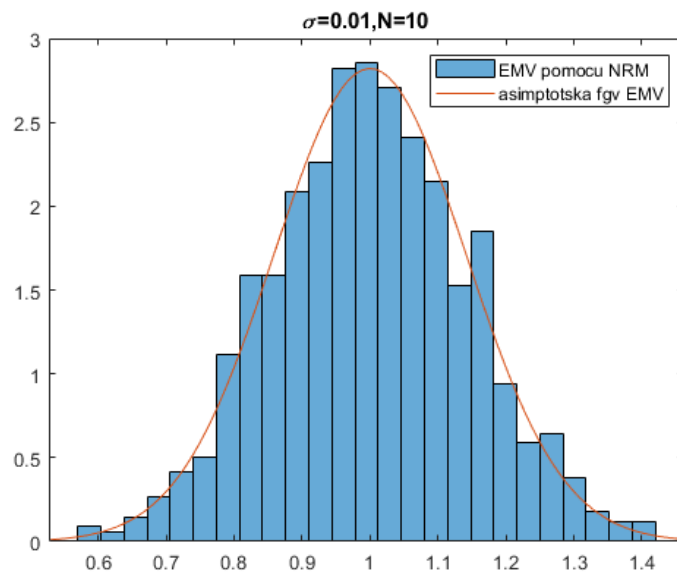
phi0 = 0.5;
sigma = sqrt(1);
N = 10;
Nr = 1000;
n=0:N-1;
estimacija = zeros(1, Nr);
for j=1:Nr
    x = zeros(1, N);
    for i = 1:N
        x(1, i) = A*cos(2*pi*f*(i-1) + tacna_vrednost) + randn*sigma;
    end
    phi_k1 = phi0;
    maxiter = 5;
    l1(j,:) = -(N/2)*log(2*pi*sigma^2)-(1/(2*sigma^2))*sum((x-A*cos(2*pi*f*n+phi_k1)).^2);
    while(maxiter > 0)

```

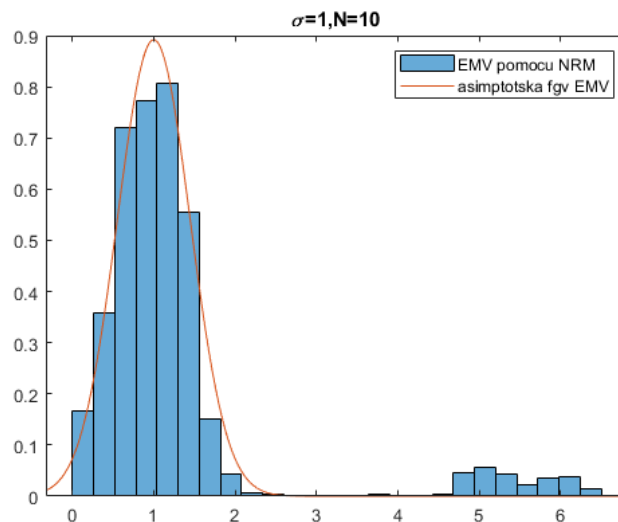
```

maxiter = maxiter - 1;
phi_k = phi_k1;
suma = 0;
for i=1:N
    suma = suma + x(1,i)*sin(2*pi*f*(i-1) + phi_k) - A/2*sin(4*pi*f*(i-1) + 2*phi_k);
end
dl1 = -A/(sigma)^2*suma;
suma = 0;
for i=1:N
    suma = suma + x(1, i)*cos(2*pi*f*(i-1) + phi_k) - A*cos(4*pi*f*(i-1) + 2*phi_k);
end
ddl1 = -A/(sigma)^2*suma;
phi_k1 = phi_k - dl1/ddl1;
%phi_k1 = phi_k1 - 2*pi*floor(phierr/2/pi);
phi_k1=mod(phi_k1,2*pi);
end
estimacija(1, j) = phi_k1;
end
figure();
histogram(estimacija, 25, 'Normalization', 'pdf');
hold all
I = A^2*N/(sigma^2*2);
varijansa = 1/I;
x = xlim;
x = x(1):0.01:x(2);
phi_a = 1/(sqrt(2*pi*varijansa))*exp(-(1/(2*varijansa)).*(x - tacna_vrednost).^2);
plot(x, phi_a);
legend('EMV pomocu NRM','asimptotska fgv EMV');
title('\sigma=1,N=10');
hold off

```



Znamo da je EMV asimptotski normalan, nepomeren i efikasan estimator. Za $N=10$ i varijansu šuma 0.1, histograma procena se prilično poklapa sa teorijskom asimptotskom procenom koja je dobijena preko Fisherove informacije. Očekivanje EMV-a koji je dobijen preko Newton-Rapshonove metode razlikuje se od teorijskog očekivanja za red veličine 10^{-3} . Zaključujemo da je 10 opservacija dovoljno da se dobije dovoljno precizna procena. Ako je varijansa šuma 1, histograma procena odstupa od asimptotske. Tek će se za 50 ili više opservacija histograma procena relativno dobro poklapati sa asimptotskom, a greška odstupanja očekivanja od teorijskog će biti reda veličine 10^{-3} . U zavisnosti od varijanse šuma vrednost koju šum može da uzme može biti iz određenog opsega, a to direktno utiče i na opseg vrednosti koje merenja tj. opservacije može imati. Pošto vrednosti šuma mogu biti za veliku varijansu i dosta udaljene od očekivane vrednosti, tako i merenja koja dobijamo mogu imati vrednosti koje odstupaju dosta od tačne vrednosti. Zbog toga nam je potreban veći broj opservacija, jer što više merenja imamo, imamo veću količinu informacija o vrednosti koju treba da procenimo, te je estimator tačniji. Dakle, ako želimo da povećamo tačnost estimatora koja nije velika jer je opseg razumnih vrednosti za parametar koji procenjujemo veliki, potrebno je da povećamo broj opservacija, i time bismo našli neki balans između uticaja varijanse i broja opservacija na procenu.



2.Zadatak

$$s[n] = \begin{bmatrix} p[n] \\ v[n] \\ a[n] \end{bmatrix} = \begin{bmatrix} 1 & T_s & \frac{T_s^2}{2} \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix} s[n-1] + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u[n]$$

$$x[n] = [1 \ 0 \ 0]s[n] + w[n]$$

Iteracija Kalmanovog filtra:

- Predikcija: $s[n|n-1] = As[n-1|n-1]$
- Predikcija MSKG: $M[n|n-1] = AM[n-1|n-1]A^T + BQB^T$
- Kalmanovo pojačanje: $K[n] = M[n|n-1]H^T(C + HM[n|n-1]H^T)^{-1}$
- Korekcija: $s[n|n] = s[n|n-1] + K[n](x[n] - s[n|n-1])$
- Korekcija MSKG $M[n|n] = (1 - K[n])M[n|n-1]$

Eksperimentišići sa različitim varijansama šuma procesa primećeno je da je “najbolja” vrednost za ovaj parametar neka vrednost koja je blizu nule, tačnije što manja vrednost. Ipak, ako uzmemo vrednost 0 doći će do overfitovanja. Takođe, što je šum procesa veći biće veće i Kalmanovo pojačanje. To znači da što je manja standardna devijacija šuma, to više verujemo usvojenom modelu. Uzeta vrednost za optimalnu standardnu devijaciju je 0.005. Za standardnu devijaciju $\sigma_u^*/5$ dolazi do overfitovanja, tj. procenjene vrednosti su previse idealizovane. Kalmanovo pojačanje je najmanje za ovu vrednost standardne devijacije. Najveće Kalmanovo pojačanje je za $5\sigma_u^*$.

```
clc
clear
close all

FID = fopen('gps_data.txt');
formatSpec = '%f';
Opservacije = fscanf(FID, formatSpec);
%s=[p;v;a];
pr_put(1) = Opservacije(1);
for i=1:(length(Opservacije)-1)
pr_put(i+1) = Opservacije(i+1)-Opservacije(i);
end
ubrzanje(1) = pr_put(1);
for i=1:(length(pr_put)-1)
ubrzanje(i+1) = pr_put(i+1)-pr_put(i);
end
Ts = 1;
sgmw = 1;
```

```

sgmu = 0.005;
sgmu1 = 5*0.005;
sgmu2 = 0.005/5;
varu= sgmw^2;
varu1=sgmu1^2;
varu2=sgmu2^2;
H = [1, 0, 0];
B = [0; 0; 1];
A =[1, Ts, Ts^2/2; 0, 1, Ts; 0, 0, 1];
%Q = zeros(3,3);
Q = varu;
Q1=varu1;
Q2=varu2;
C = sgmw^2;
% -1 je 1
s_kapa = zeros(3,length(Observacije));
s=[0;0;0];
%K_pojacanje(:,1)=[0;0;0];
%s_kapa(:,1) = [0;0;0];
s_kapa(:,1) = A*s;
M=eye(3);
M_pret(:,1) = diag(M);
M = A*M*transpose(A) + B*Q*transpose(B);
M_tren(:,1) = diag(M);
K_pojacanje(:,1)=M*H'.*inv(C + H*M*H');
for i=2:(length(Observacije))

    s_kapa(:,i) = A*s_kapa(:,i-1);
    M = A*M*transpose(A) + B*Q*transpose(B);
    M_pret(:,i) = diag(M);
    K_pojacanje(:,i) = M*H'.*inv(C + H*M*H');
    s_kapa(:,i) = s_kapa(:,i)+K_pojacanje(:,i)*(Observacije(i)-H*s_kapa(:,i));
    M = (eye(3)-K_pojacanje(:,i)*H)*M;
    M_tren(:,i) = diag(M);
end
figure(1)
plot(1:length(s_kapa),s_kapa(3,:));
title('Procenjeno ubrzanje');
figure(2)
plot(1:length(s_kapa),s_kapa(2,:));
title('Procenjena brzina');

figure(3)
plot(1:length(s_kapa),s_kapa(1,:));
hold all
plot(1:length(Observacije),Observacije);
title('Pozicije');
legend('estimirano','opservirano');
hold off

figure(4)
plot(1:length(K_pojacanje),K_pojacanje);
title('Pojacanje K');

```

```
legend('pozicija','brzina','ubrzanje');
```

```
figure(5)
```

```
plot(1:length(M_pret),M_pret);
```

```
title('M[n|n-1]');
```

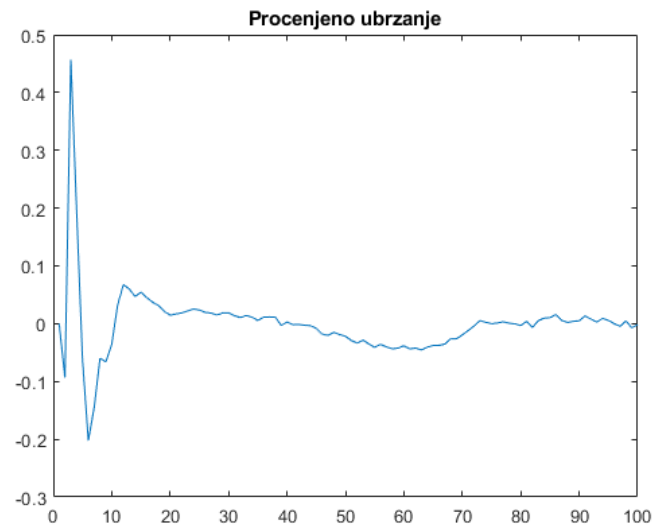
```
legend('pozicija','brzina','ubrzanje');
```

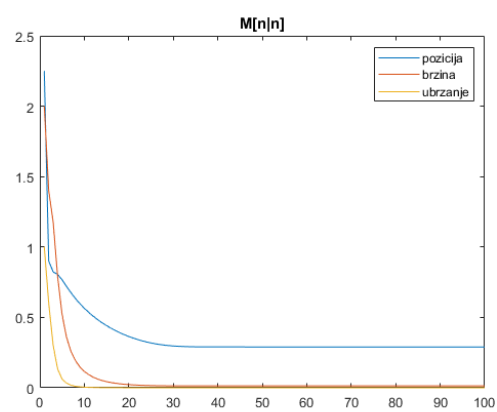
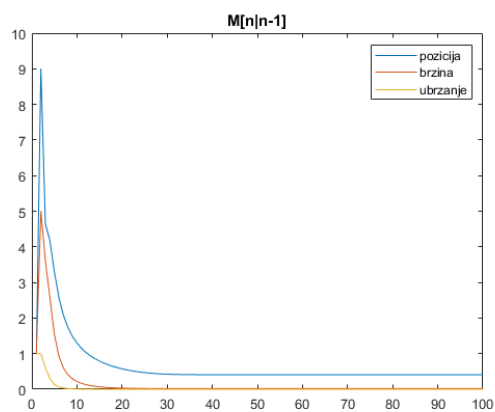
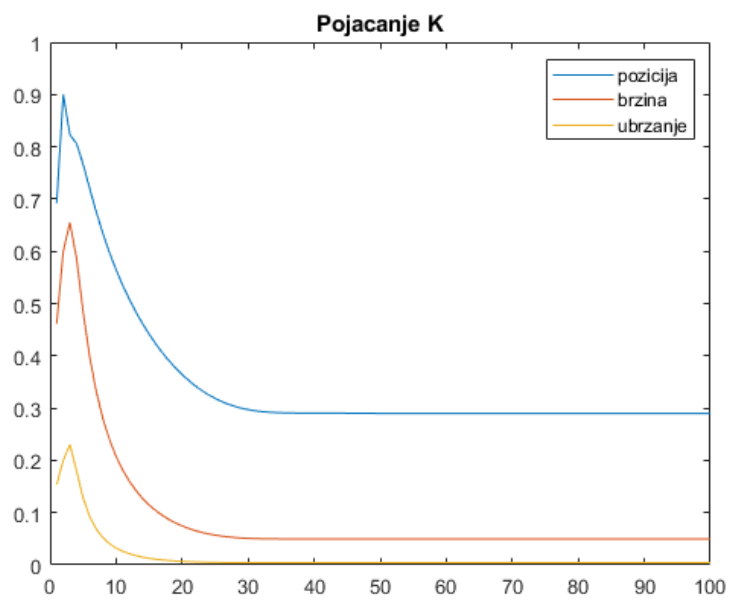
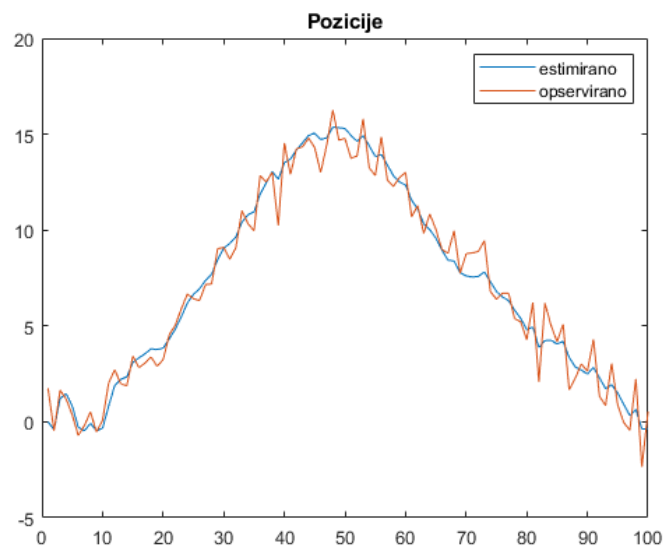
```
figure(6)
```

```
plot(1:length(M_tren),M_tren);
```

```
title('M[n|n]');
```

```
legend('pozicija','brzina','ubrzanje');
```





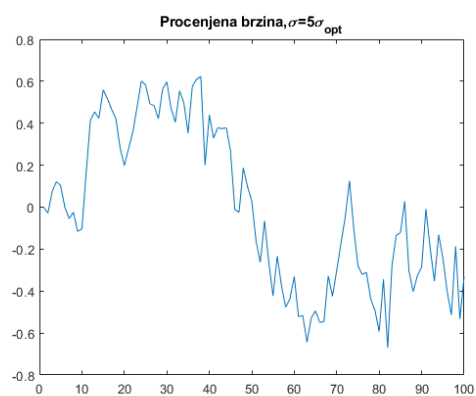
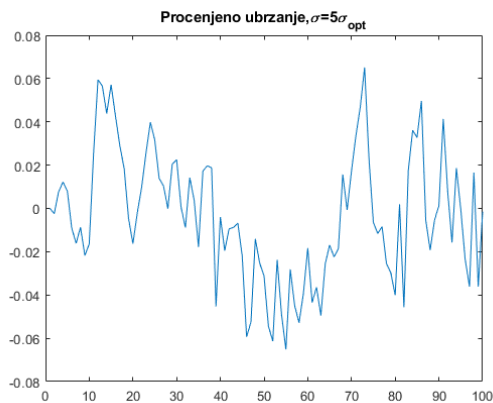
za 5sigma

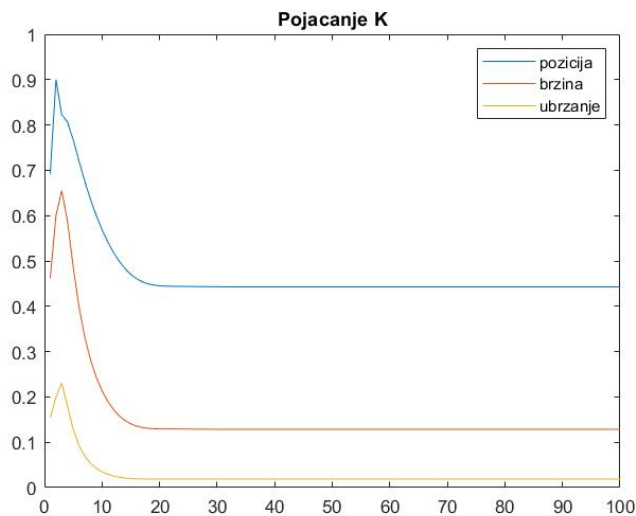
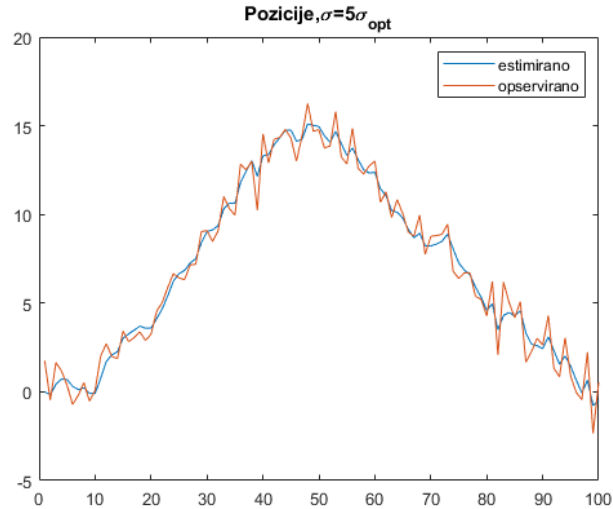
```
M = A*M*transpose(A) + B*Q1*transpose(B);
%M_tren(:,1)=[1;1;1];
M_tren(:,1) = diag(M);
K_pojacanje(:,1)=M*H'.*inv(C + H*M*H');
for i=2:(length(opservacije))

    s_kapa(:,i) = A*s_kapa(:,i-1);
    M = A*M*transpose(A) + B*Q1*transpose(B);
    M_pret(:,i) = diag(M);
    K_pojacanje(:,i) = M*H'.*inv(C + H*M*H');
    s_kapa(:,i) = s_kapa(:,i)+K_pojacanje(:,i)*(opservacije(i)-H*s_kapa(:,i));
    M = (eye(3)-K_pojacanje(:,i)*H)*M;
    M_tren(:,i) = diag(M);
end
figure()
plot(1:length(s_kapa),s_kapa(3,:));
title('Procenjeno ubrzanje,\sigma=5\sigma_{opt}');
figure()
plot(1:length(s_kapa),s_kapa(2,:));
title('Procenjena brzina,\sigma=5\sigma_{opt}');

figure()
plot(1:length(s_kapa),s_kapa(1,:));
hold all
plot(1:length(opservacije),opservacije);
title('Pozicije,\sigma=5\sigma_{opt}');
legend('estimirano','opservirano');
hold off

figure(4)
plot(1:length(K_pojacanje),K_pojacanje);
title('Pojacanje K');
legend('pozicija','brzina','ubrzanje');
```





$\sigma = \sigma / 5$

```
M = A*M*transpose(A) + B*Q2*transpose(B);
%M_tren(:,1)=[1;1;1];
M_tren(:,1) = diag(M);
K_pojacanje(:,1)=M*H'.*inv(C + H*M*H');
for i=2:(length(Observacije))

    s_kapa(:,i) = A*s_kapa(:,i-1);
    M = A*M*transpose(A) + B*Q2*transpose(B);
    M_pret(:,i) = diag(M);
    K_pojacanje(:,i) = M*H'.*inv(C + H*M*H');
    s_kapa(:,i) = s_kapa(:,i)+K_pojacanje(:,i)*(Observacije(i)-H*s_kapa(:,i));
    M = (eye(3)-K_pojacanje(:,i)*H)*M;
    M_tren(:,i) = diag(M);
end
figure()
```

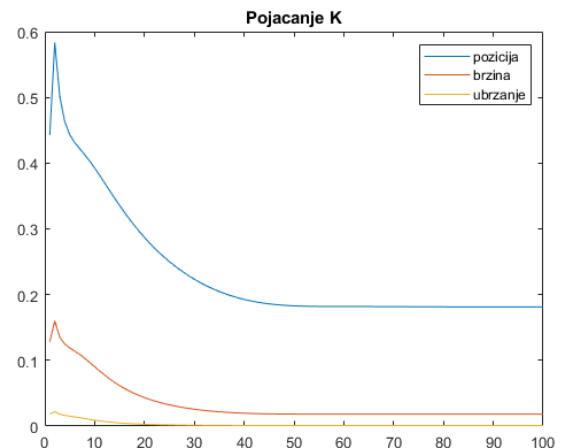
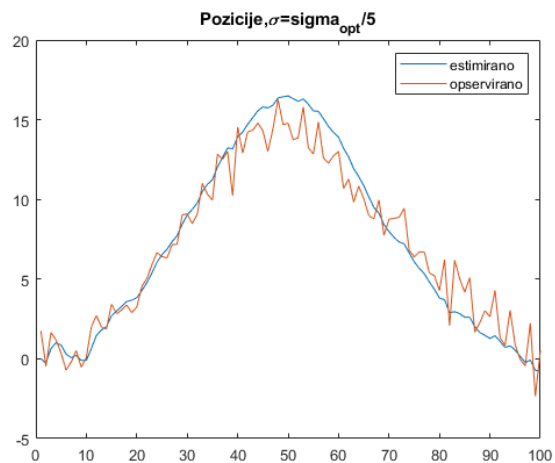
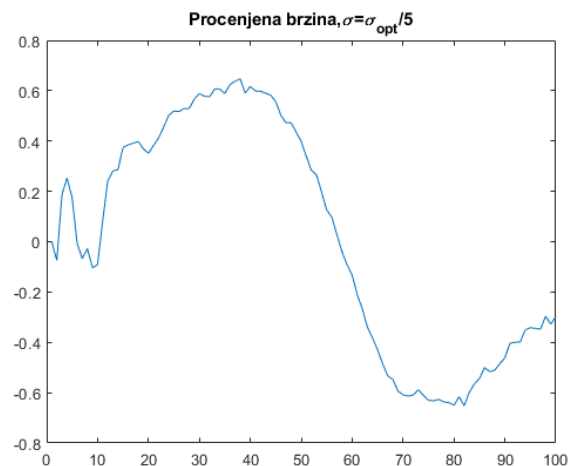
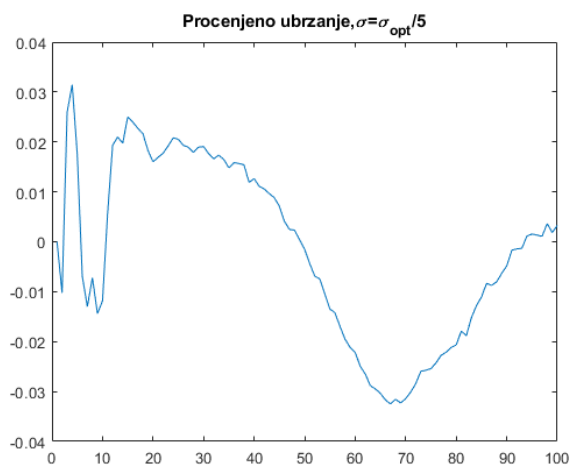
```

plot(1:length(s_kapa),s_kapa(3,:));
title('Procenjeno ubrzanje,\sigma=\sigma_{opt}/5');
figure()
plot(1:length(s_kapa),s_kapa(2,:));
title('Procenjena brzina,\sigma=\sigma_{opt}/5');

figure()
plot(1:length(s_kapa),s_kapa(1,:));
hold all
plot(1:length(Opservevacije),Opservevacije);
title('Pozicije,\sigma=\sigma_{opt}/5');
legend('estimirano','opservevirano');
hold off

figure(4)
plot(1:length(K_pojacanje),K_pojacanje);
title('Pojacanje K');
legend('pozicija','brzina','ubrzanje');

```



Promena početnih uslova ne utiče na Kalmanovo pojačanje, kao ni na srednju kvadratnu grešku. Promena početnog uslova $M[-1]$ ne menjaju se stacionarne vrednosti Kalmanovog pojačanja i srednje kvadratne greške.

Tacka druga

```
s_pocetno = [5;5;5];
M_pocetno = 10*eye(3);
s = [5;5;5];
M=10*eye(3);

M_pret(:,1) = diag(M);
M = A*M*transpose(A) + B*Q*transpose(B);
M_tren(:,1) = diag(M);
K_pojacanje(:,1)=M*H'.*inv(C + H*M*H');
for i=2:(length(opservacije))

    s_kapa(:,i) = A*s_kapa(:,i-1);
    M = A*M*transpose(A) + B*Q*transpose(B);
    M_pret(:,i) = diag(M);
    K_pojacanje(:,i) = M*H'.*inv(C + H*M*H');
    s_kapa(:,i) = s_kapa(:,i)+K_pojacanje(:,i)*(opservacije(i)-H*s_kapa(:,i));
    M = (eye(3)-K_pojacanje(:,i)*H)*M;
    M_tren(:,i) = diag(M);
end
figure()
plot(1:length(s_kapa),s_kapa(3,:));
title('Procenjeno ubrzanje');
figure()
plot(1:length(s_kapa),s_kapa(2,:));
title('Procenjena brzina');

figure()
plot(1:length(s_kapa),s_kapa(1,:));
hold all
plot(1:length(opservacije),opservacije);
title('Pozicije');
legend('estimirano', 'opservirano');
hold off

figure()
plot(1:length(K_pojacanje),K_pojacanje);
title('Pojacanje K');
legend('pozicija', 'brzina', 'ubrzanje');

figure()
plot(1:length(M_pret),M_pret);
title('M[n|n-1]');
legend('pozicija', 'brzina', 'ubrzanje');

figure()
plot(1:length(M_tren),M_tren);
```

```
title('M[n|n]');
legend('pozicija','brzina','ubrzanje');
```

