Drugi domaci zadatak iz SSE

Radila: Dragana Ninkovic 2019/0052

Prvi zadatak

a) analiticki izrazi

$$E(w[n]) = \mu + \beta\gamma \qquad \mu = 0 \quad \beta = 1$$

$$Var(w[n]) = \beta^2 \frac{\pi^2}{6} \qquad E(w[n]) = \gamma = 0.577$$

$$F(w; \mu, \beta) = e^{-e^{-\frac{(x-\mu)}{\beta}}} \qquad var(w[n]) = \frac{\pi^2}{6}$$

$x[n] = \theta + w[n] \Rightarrow x[n]$ мора имати исту расподелу само са другачијим параметрима $\left[\begin{matrix} x = g(w) \\ w = x - \theta \end{matrix}\right]$

$$\left(g(w) = \theta + w \quad g'(w) = 1 \Rightarrow fgvx = fgvg(w) = g(\widetilde{x-\theta})\right)$$

$$Ex[n] = Ew[n] + \theta = \gamma + \theta = \mu_1 + \beta_1 \gamma \Rightarrow \beta_1 = 1, \ \mu_1 = \theta$$

$$x[n] \sim e^{-z - e^{-z}} \qquad z = \frac{x-\mu}{\beta} = x - \theta$$

$$x[n] \sim e^{-(x-\theta) - e^{-(x-\theta)}}$$

независне су опсервације

$$p(x;\theta) = \prod_{n=0}^{N-1} p(x[n];\theta) = e^{-\sum_{n=0}^{N-1} x[n] + N\theta - e^{\theta} \sum_{n=0}^{N-1} e^{-x[n]}}$$

$$\log p(x;\theta) = -\sum_{n=0}^{N-1} x[n] + N\theta - e^{\theta} \sum_{n=0}^{N-1} e^{-x[n]}$$

$$\frac{\partial \ln p(x;\theta)}{\partial \theta} = N - e^{\theta} \sum_{n=0}^{N-1} e^{-x[n]}$$

$$\frac{\partial^2 \ln p(x;\theta)}{\partial \theta^2} = -e^{\theta} \sum_{n=0}^{N-1} e^{-x[n]}$$

$$\underset{\theta}{\operatorname{argmax}}\, p(x;\theta) = \underset{\theta}{\operatorname{argmax}}\, \ln p(x;\theta)$$

$$\Rightarrow \frac{\partial \ln p(x;\theta)}{\partial \theta} = 0 \Rightarrow \hat{\theta} = \ln \frac{N}{\sum_{n=0}^{N-1} e^{-x[n]}}$$

```
clear; close all; clc;
```

```
global x;
global N;
```

```
N = 10;
Nr = 100;
syms theta
```

```
x = csvread('dom2_zad1.csv');
```

optimalni estimator metodom maksimalne verodostojnosti

```
theta_est_opt = log( N./sum(exp(-x),2) );
```

funkcija log verodostojnosti

Editor - C:\Users\test4\Desktop\SSE_DZ2\l.m

zad6_1.m | kons4_1.m | Zad1.m | NewtonRapson.m | l.m | grad_l.m | grad_grad_l.m

```
function reliability=l(theta)
global x;
global N;
reliability=-sum(x,2)+N*theta-sum(exp(-x),2)*exp(theta);
end
```

funkcija prvog izvoda log verodostojnosti

Editor - C:\Users\test4\Desktop\SSE_DZ2\grad_l.m

zad6_1.m | kons4_1.m | Zad1.m | NewtonRapson.m | l.m | grad_l.m | grad_grad_l.m

```
function grad_reliability=grad_l(theta)
global N
global x
grad_reliability=N-sum(exp(-x),2).*exp(theta);
end
```

funkcija drugog izvoda log verodostojnosti

Editor - C:\Users\test4\Desktop\SSE_DZ2\grad_grad_l.m

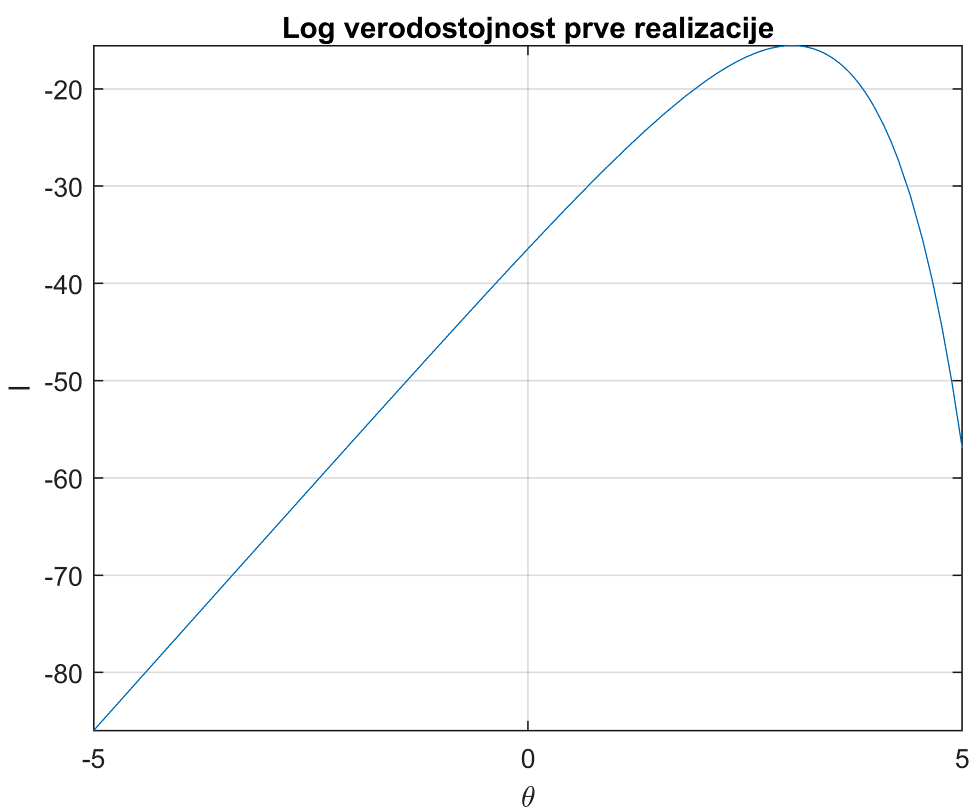
zad6_1.m | kons4_1.m | Zad1.m | NewtonRapson.m | l.m | grad_l.m | grad_grad_l.m

```
function grad_grad_r=grad_grad_l(theta)
global x
grad_grad_r=-sum(exp(-x),2).*exp(theta);
end
```

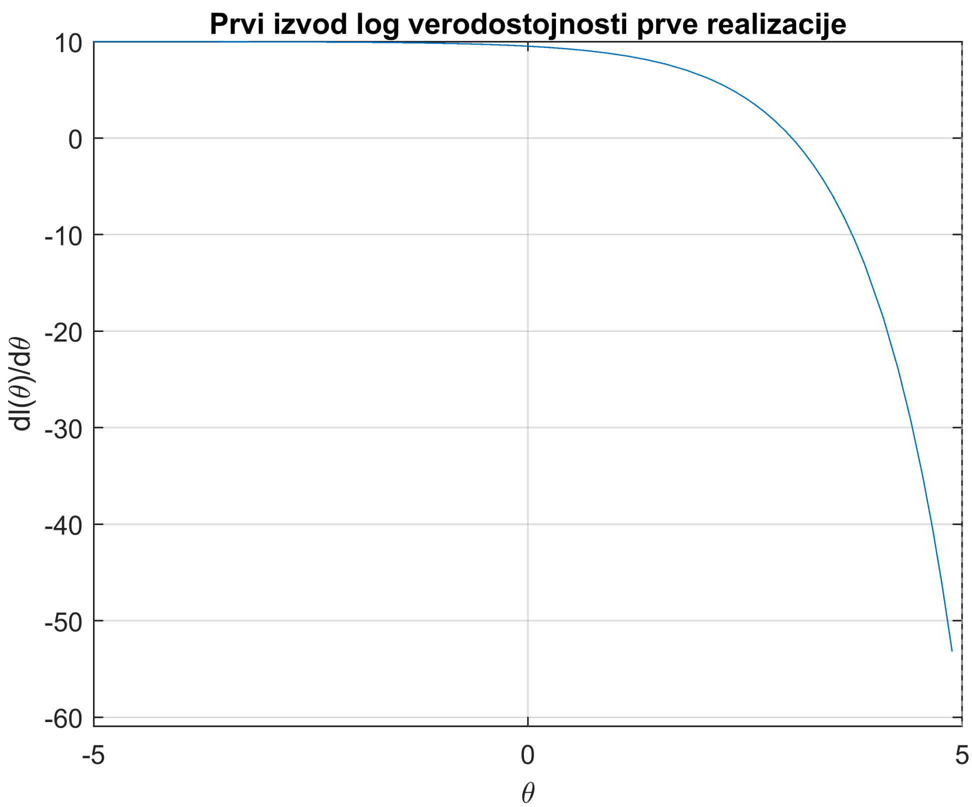
```
        l_ = l(theta);
    grad_l_ = grad_l(theta);
grad_grad_l_ = grad_grad_l(theta);
```

graficki prikaz ovih funkcija

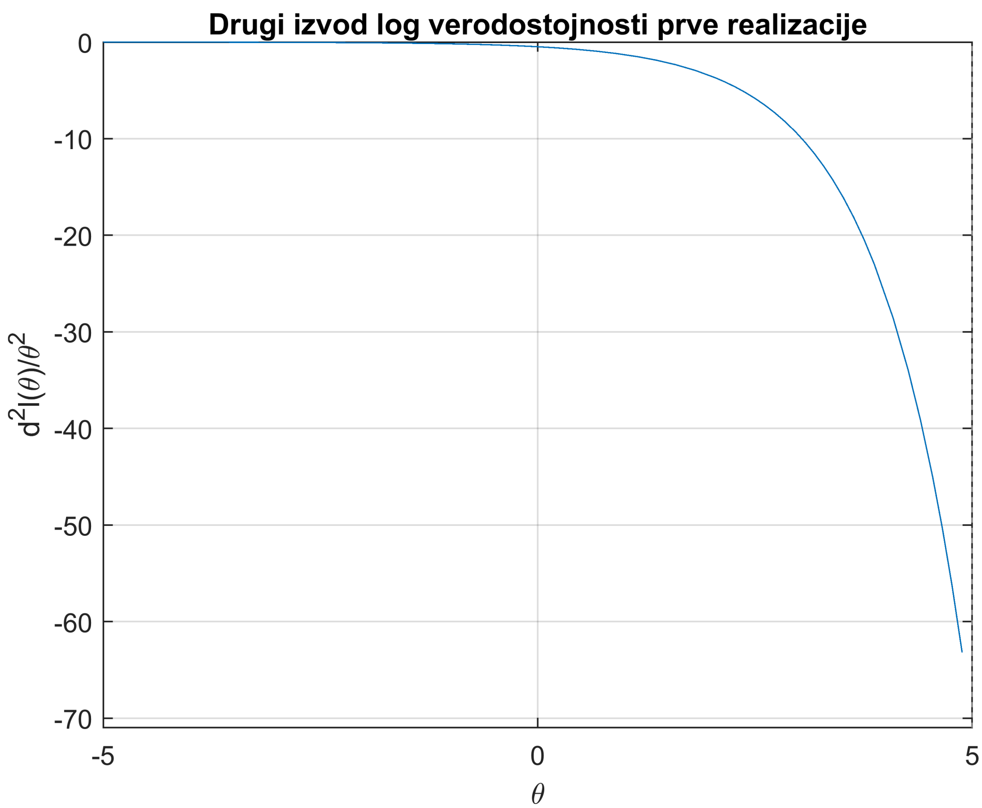
```
figure(1)
fplot( l_(1,:) )
title('Log verodostojnost prve realizacije'); grid on;
ylabel('l'); xlabel('\theta')
```



```
figure(2)
fplot( grad_l_(1,:) );
title("Prvi izvod log verodostojnosti prve realizacije"); grid on;
ylabel('dl(\theta)/d\theta'); xlabel('\theta');
```



```
figure(3)
fplot( grad_grad_l_(1,:) );
title('Drugi izvod log verodostojnosti prve realizacije'); grid on;
ylabel('d^2l(\theta)/\theta^2'); xlabel('\theta');
```



b) funkcija koja realizuje Newton Rapson-ovu metodu

Editor - C:\Users\test4\Desktop\SSE_DZ2\NewtonRapson.m

zad6_1.m | kons4_1.m | Zad1.m | NewtonRapson.m | l.m | grad_l.m | grad_grad_l.m

```
function theta = NewtonRapson(theta_0,grad_f,grad_grad_f,tol,max_iter)
theta_k=theta_0;
theta_k_prev=zeros(size(theta_0));
num_it=1;

while (any(abs(theta_k-theta_k_prev)>=tol) & (num_it<max_iter))
    theta_k_prev=theta_k;
    theta_k=theta_k-(grad_f(theta_k)./grad_grad_f(theta_k));
    num_it=num_it+1;
end
theta=theta_k;
end
```

```
theta = NewtonRapson( 2*ones(Nr,1), @grad_l, @grad_grad_l, 0.00001, 100);

   mu = mean(theta)
```

```
mu = 2.3514
```

```
sigma = std(theta)
```

```
sigma = 0.3154
```

provera Newton Rapson-ove metode

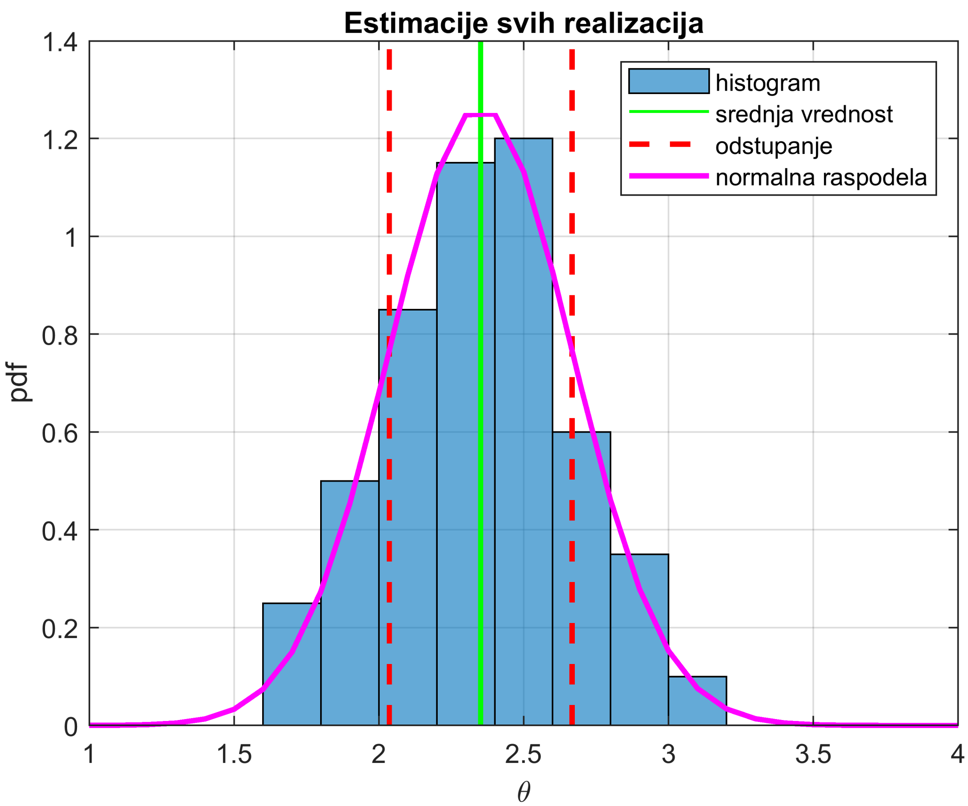
```
all(round(theta, 4) == round(theta_est_opt, 4))
```

```
ans = logical
   1
```

c) graficki prikaz dobijenih Nr estimacija

```
figure(4)
p1 = histogram(theta, 'Normalization', 'pdf');
hold on;
p2 = plot([mu, mu], [0, 1.4], 'Color', 'g', 'LineWidth', 2);
hold on;
```

```
p3 = plot([mu - sigma, mu - sigma],[0, 1.4],'Color', 'r', 'LineWidth', 2, 'LineStyle', '--');
hold on;
p4 = plot([mu + sigma, mu + sigma],[0, 1.4], 'Color', 'r', 'LineWidth', 2, 'LineStyle', '--');
hold on;
p5 = plot(1:0.1:4, normpdf(1:0.1:4, mu, sigma), 'Color', 'm', 'LineWidth', 2);
title('Estimacije svih realizacija'); grid on;
ylabel('pdf'); xlabel('\theta');
legend([p1,p2,p3,p5],...
    {'histogram', 'srednja vrednost',...
    'odstupanje', 'normalna raspodela'});
hold off;
```



Drugi zadatak

(1) $v[t+1] = v[t] + T_s\, a[t]$

(2) $p[t+1] = p[t] + T_s\, v[t] + \frac{T_s^2}{2} a[t]$

$$\left.\begin{aligned} a[t] &= \mu_a[t] + \tilde{a}[t] \\ \tilde{a}[t+1] &= \tilde{a}[t] + u[t] \end{aligned}\right\}$$

$\Rightarrow a[t+1] - \mu_a[t+1] = a[t] - \mu_a[t] + u[t]$

(3) $a[t+1] = a[t] + (\mu_a[t+1] - \mu_a[t]) + u[t]$

$$s[t] = \begin{bmatrix} p[t] \\ v[t] \\ a[t] \end{bmatrix}$$

$$s[t+1] = \overbrace{\begin{bmatrix} 1 & T_s & \frac{T_s^2}{2} \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix}}^{A} s[t] + \overbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}^{B} u[t]$$

$$+ \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} (\mu_a[t+1] - \mu_a[t])$$

$$x[t] = \underbrace{[1\ 0\ 0]}_{H} s[t]_{3\times 1} + w[t]$$

```
clear; close all; clc;
x = csvread('gnss_data.csv');
n = 0:(length(x) - 1);
Ts = 1;
```

sum merenja

```
% vreme krece od 0 ali indeksiranje ide od 1 zato je npr 44.sekund == 45. odbirak
sigma_w = 10 * ones(size(n));
% raste za 10 od 40. sekunde == 41.odbirak dostize 10+50=60 u 44.skekundi==45.odbirku
sigma_w(41:45) = sigma_w(41:45) + [10, 20, 30, 40, 50];
% od 45.sekunde tj 46.odbirka signal se gubi
sigma_w(46:55) = Inf;
% od 55. sekunde tj 56. odbirka opada od 60 do 10 i on da ostaje const
```

```
sigma_w(56:60) = sigma_w(56:60) + [50, 40, 30, 20, 10];
```

srednja vrednost a

```
mu_a = [5 * ones(1, 30), zeros(1, 40), -5 * ones(1, 30)];
```

devijacija stohastickog dela ubrzanja koja potice od u

```
sigma_a = [5 * ones(1, 30), ones(1, 40), 5 * ones(1, 30)];
var_a = (sigma_a./3).^2;% 3 sigma interval jer vrlo retko odstupa
```

matrice u modelu stanja

```
v = [0; 0; 1];% vektor kojim mnozim deterministicki ulaz
% koji utice samo na ubrsanje
A = [1, Ts, Ts*Ts/2; 0, 1, Ts; 0, 0, 1];
B = [0; 0; 1];
H = [1, 0, 0];
R = (sigma_w).^2;% varijansa suma merenja
```

pocetne vrednosti parametara KF

```
x_est = [0; 0; 5];% pocetne vrednosti vektora koji estimiramo
M_est = [0, 0, 0; 0, 0, 0; 0, 0, 25];% pocetne nesigurnosti
% sigurni smo da u pocetnom trenutku vozilo polazi iz
% pocetne stanice
% takodje smo sigurni da je onda pocetna brzina 0
% iz navedenih razloga pocetna nesigurnost je nula
% pocetno odstupanje ubrzanja je +/- 5

x_estimirano = zeros(3, length(x) + 1);
M_estimirano = zeros(3, length(x) + 1);
K_ = zeros(3, length(x));

x_estimirano(:,1) = x_est;
M_estimirano(:,1) = [M_est(1,1); M_est(2,2); M_est(3,3)];
```

iteracije KF

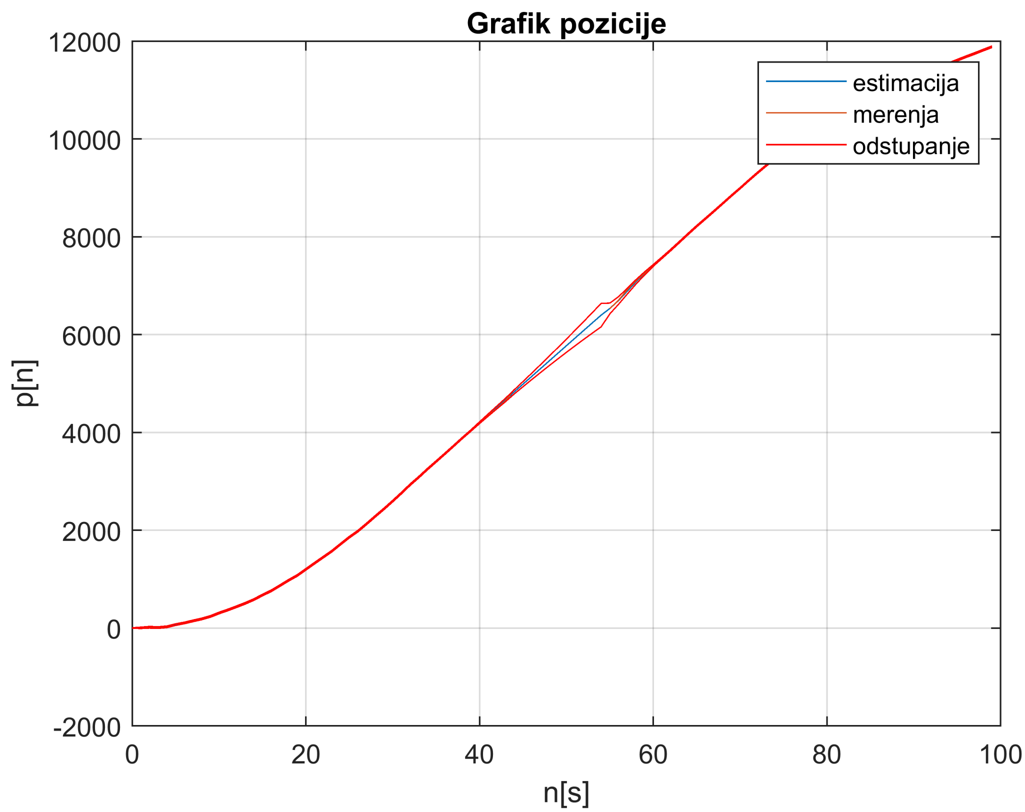
```
for i = 1:length(x)
    % Q je promenljiva u vremenu
    Q = var_a(i);

    % predikcija
    x_pred = A *  x_est;
     if(i < length(x))
```

```
        % dodavanje deterministickog ulaza za sve odbirke
        % poslednji nije bitan jer je svakako mu_a konstantno na kraju putanje
        x_pred = x_pred + v * (mu_a(i + 1) - mu_a(i));
    end

    M_pred = A*M_est*A' + B*Q*B';
    % estimacija
    if i<46 || i>55
       % kada nije u tunelu
       K = M_pred*H' * (R(i) + H*M_pred*H')^-1;
       x_est = x_pred + K*(x(i) - H*x_pred);
       M_est = (eye(3)- K*H) * M_pred;

    else% od 45.sekunde==46.odbirka do 54 sekunde tj 55.odbirka(ukljucujuci njega)
        % signal je izgubljen->stacionarno stanje
       x_est = x_pred;
       M_est = M_pred;
       K = 0;
    end

    x_estimirano(:,i+1) = x_est;
    M_estimirano(:,i+1) = [M_est(1,1); M_est(2,2); M_est(3,3)];
    K_(:,i) = K;

end
```

graficki prikaz estimacija dobijenih pomocu KF

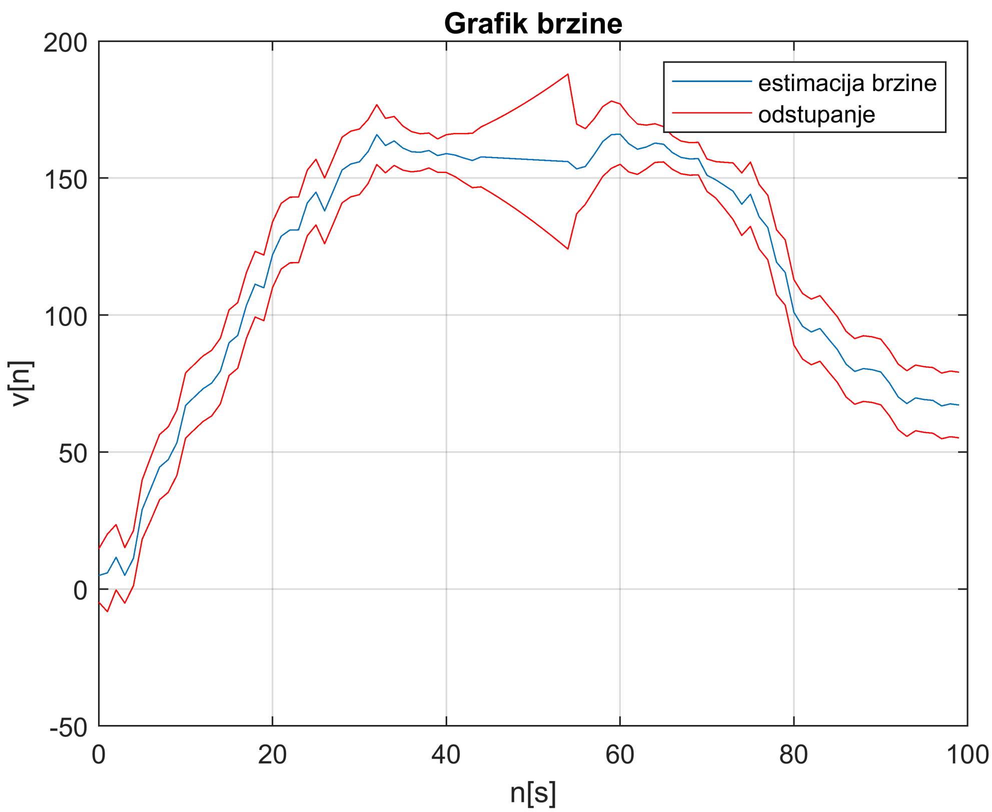
1) pozicija

```
M2sigma = 2 * sqrt(M_estimirano);
% sigma je koren iz varijanse

figure(4)
p1 = plot(n, x_estimirano(1,2:end));
hold on;
p2 = plot (n, x);
hold on
p3 = plot(n, x_estimirano(1,2:end) + M2sigma(1,2:end),'Color','r');
hold on
p4 = plot(n, x_estimirano(1,2:end) - M2sigma(1,2:end),'Color','r');
legend([p1, p2, p3], {'estimacija', 'merenja', 'odstupanje'});
title('Grafik pozicije');grid on;
xlabel('n[s]'); ylabel('p[n]');
xlim([0, 100]); hold off;
```



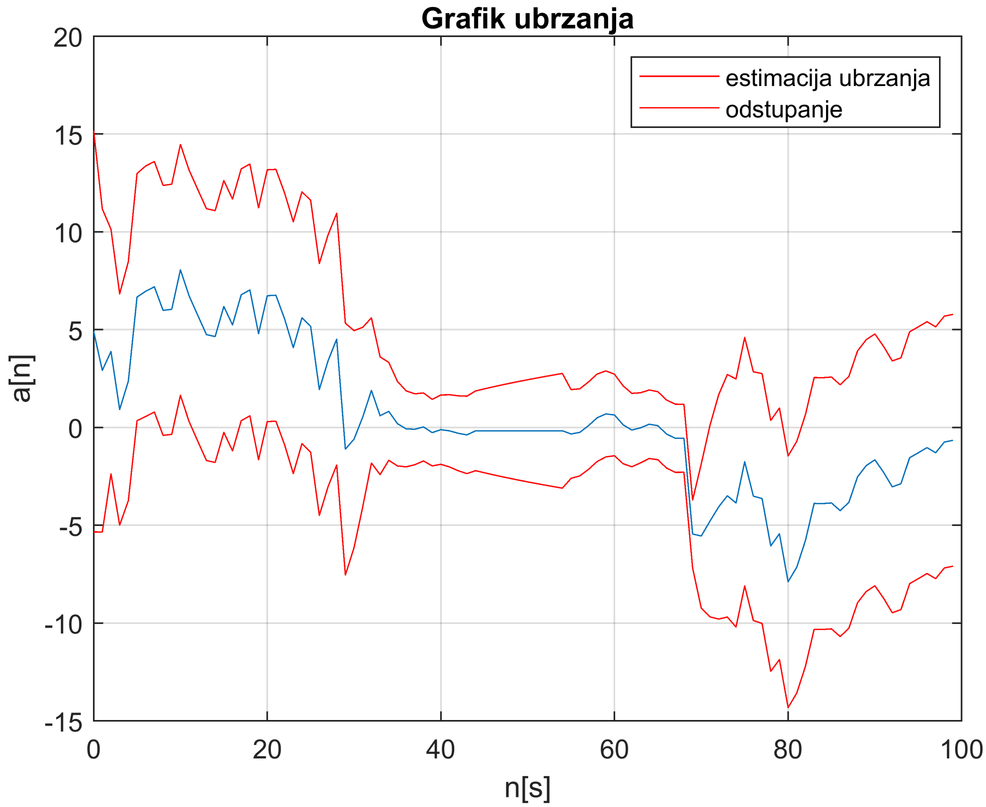
2) brzina

```
figure(5)
p5 = plot(n, x_estimirano(2, 2:end));
hold on;
p6 = plot(n, x_estimirano(2, 2:end) + M2sigma(2, 2:end),'Color','r');
hold on;
p7 = plot(n, x_estimirano(2, 2:end) - M2sigma(2, 2:end),'Color','r');
title('Grafik brzine'); grid on;
xlabel('n[s]'); ylabel('v[n]');
legend([p5, p6], {'estimacija brzine','odstupanje'});
xlim([0, 100]); hold off;
```



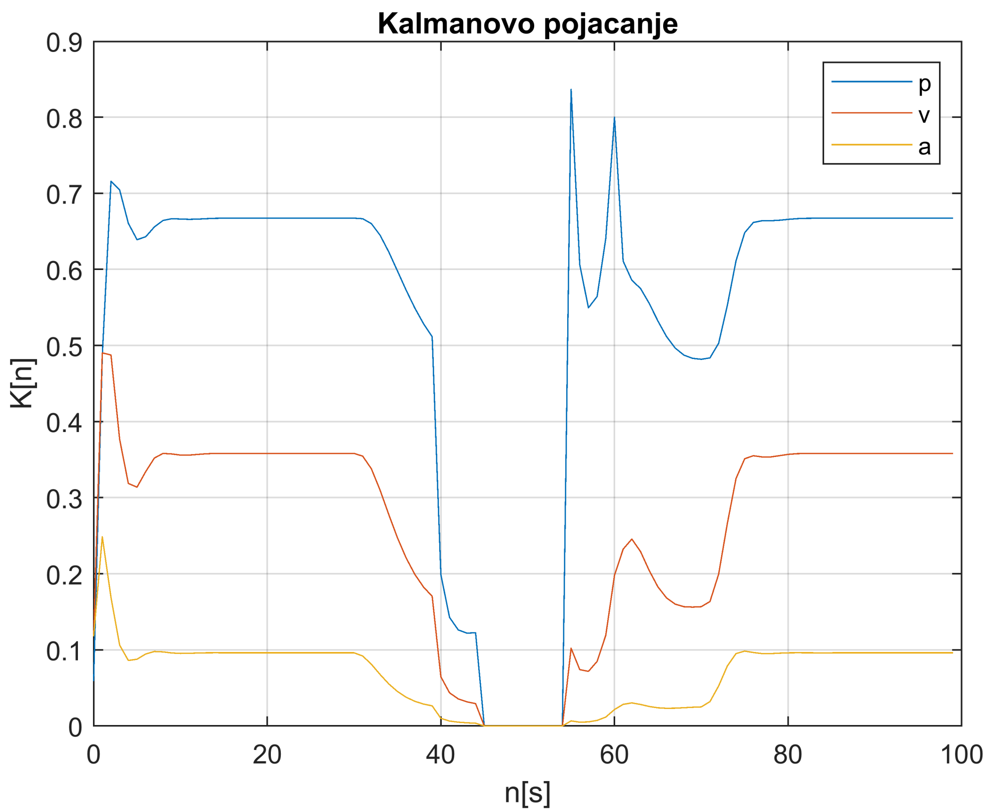
3) ubrzanje

```
figure(6)
plot(n, x_estimirano(3, 2:end));
hold on;
p8=plot(n, x_estimirano(3, 2:end) + M2sigma(3, 2:end),'Color','r');
hold on;
p9=plot(n, x_estimirano(3,2:end) - M2sigma(3,2:end),'Color','r');
title('Grafik ubrzanja'); grid on;
xlabel('n[s]'); ylabel('a[n]');
legend([p8, p9], {'estimacija ubrzanja', 'odstupanje'});
xlim([0, 100]); hold off;
```



4)Kalmanovo pojacanje

```
figure(7)
plot(n, K_(1,:))
hold on;
plot(n, K_(2,:))
hold on;
plot(n, K_(3,:));
title("Kalmanovo pojacanje"); grid on;
xlabel('n[s]'); ylabel('K[n]')
legend('p','v','a')
xlim([0, 100]); hold off;
```


Kalmanovo pojacanje
K[n]
n[s]
p
v
a
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0
20
40
60
80
100