

DRUGI DOMAĆI ZADATAK IZ PREDMETA ROBOTIKA I AUTOMATIZACIJA

Dušan Šalović 2018/0067, Danica Bandović 2018/0018

Jun 2021.

1 Parametri simulacije

Prema tekstu zadatka, u sledećem kodu podešeni su parametri simulacije:

```
1 global m I l l1 l2 g F_ext dt mt
2
3 t = 0;          %inicijalizacija vremena [s]
4 dt = 0.001;    %korak simulacije [s]
5 i = 1;         %brojac
6 g = 9.81;      %gravitaciono ubrzanje [m/s^2]
7 mt = 0.25;     %masa tereta na završnom uredjaju
8
9 F_ext = [0; -mt*g]; %spoljasnja sila [N]
10
11 %% Parametri segmenata
12 l1 = 0.5;      %duzina prvog segmenta [m]
13 l2 = 0.5;      %duzina drugog segmenta [m]
14
15 m1 = 2;        %masa prvog segmenta [kg]
16 m2 = 2;        %masa prvog segmenta [kg]
17
18 I1 = 1/3*m1*l1^2; %moment prvog segmenta [kgm^2]
19 I2 = 1/3*m2*l2^2; %moment drugog segmenta [kgm^2]
20
21 l=l1; m=m1; I=I1;
22
23 %% Inicijalizacija pocetnog stanja
24 q=[0;0];       %pozicija
25 dq =[0;0];     %brzina
26 ddq = [0;0];  %ubrzanje
```

2 Kinematika robota

Ako su unutrašnje koordinate robota $q = [q_1 \ q_2]^T$, spoljašnje koordinate $s = [x \ z]^T$, odnosno koordinate završnog uređaja su:

$$x = l_1 \cdot \cos(q_1) + l_2 \cdot \cos(q_1 + q_2)$$

$$z = l_1 \cdot \sin(q_1) + l_2 \cdot \sin(q_1 + q_2)$$

Na osnovu ovih jednačina, matrica Jakobijana je:

$$J = \begin{bmatrix} \frac{dx}{dq_1} & \frac{dx}{dq_2} \\ \frac{dz}{dq_1} & \frac{dz}{dq_2} \end{bmatrix} = \begin{bmatrix} -l_1 \cdot \sin(q_1) - l_2 \cdot \sin(q_1 + q_2) & -l_2 \cdot \sin(q_1 + q_2) \\ l_1 \cdot \cos(q_1) + l_2 \cdot \cos(q_1 + q_2) & l_2 \cdot \cos(q_1 + q_2) \end{bmatrix}$$

Između unutrašnjih i spoljašnjih koordinata važi relacija $\dot{s} = J \cdot \dot{q}$.

Pošto je pozicija tačke C, (0.25,0.933)m data u spoljašnjim koordinatama, izračunaćemo kolike su vrednosti unutrašnjih koordinata u toj tački:

$$\cos(q_{C2}) = \frac{x^2 + z^2 - l_1^2 - l_2^2}{2l_1l_2} = 0.866$$

$$\sin(q_{C2}) = \pm\sqrt{1 - \cos^2(q_{C2})} = \pm 0.5001$$

$$q_{C2} = \arctg\left(\frac{\sin(q_{C2})}{\cos(q_{C2})}\right) = \pm 0.523698...(\pm 30.01^\circ)$$

Usvojimo $q_{C2} = 30.01^\circ$, odakle sledi da je koordinata q_{C1} :

$$q_{C1} = \arctg\left(\frac{z}{x}\right) - \arctg\left(\frac{l_2 \sin(q_{C2})}{l_1 + l_2 \cos(q_{C2})}\right) = 1.04714074426$$

Dakle, pozicije tačaka su A: $(q_{A1}, q_{A2}) = (0, 0)$ radijana, B: $(q_{B1}, q_{B2}) = (\frac{\pi}{4}, \frac{7\pi}{18})$ radijana, i C: $(q_{C1}, q_{C2}) = (0.523698, 1.0471407)$

3 Dinamika robota

Osnovna jednačina dinamike je:

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau$$

odnosno, u matričnom obliku, s obzirom da manipulator ima dva zglobova, jednačina glasi

$$\begin{bmatrix} H_{11}(q) & H_{12}(q) \\ H_{21}(q) & H_{22}(q) \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} C_{11}(q, \dot{q}) & C_{12}(q, \dot{q}) \\ C_{21}(q, \dot{q}) & C_{22}(q, \dot{q}) \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} G_1(q) \\ G_2(q) \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

Potrebno je naći matrice H, C i G. Jednačine kretanja mogu se naći Njutn-Ojlerovim pristupom, i nakon što se prikažu u formi pogodnoj za dinamičku analizu, iz njih nalazimo elemente traženih matrica. Elemente inercijalne matrice nalazimo tako što znamo da se oni u jednačinama uvek javljaju uz ubrzanja zglobova robota. Prema tome, elementi inercijalne matrice su:

$$m_1 = m_2 = m$$

$$l_1 = l_2 = l$$

$$I_1 = I_2 = I$$

$$H_{11} = I + l^2 m + \left(\frac{l}{2}\right)^2 m + \left(\frac{l}{2}\right)^2 m + l \frac{l}{2} m \cos(q_2)$$

$$H_{12} = I + \left(\frac{l}{2}\right)^2 m + l \frac{l}{2} m \cos(q_2)$$

$$H_{21} = \left(\frac{l}{2}\right)^2 m + l \frac{l}{2} m \cos(q_2)$$

$$H_{22} = I + \left(\frac{l}{2}\right)^2 m$$

Članovi matrice brzinskih efekata, C, prepoznaju se u jednačinama po tome što množe brzine zglobova.

$$C_{11} = -2l \frac{l}{2} m \sin(q_2) \dot{q}_2$$

$$C_{12} = -l \frac{l}{2} m \sin(q_2) \dot{q}_2$$

$$C_{21} = l \frac{l}{2} m \sin(q_2) \dot{q}_1$$

$$C_{22} = 0$$

Članovi gravitacione matrice prepoznaju se u jednačinama po tome što sadrže gravitaciono ubrzanje g . U našem slučaju, jednaki su:

$$G_1 = -(l_2 m_2 \cos(q_1 + q_2) + l_1 m_1 \cos(q_1) + l_1 m_2 \cos(q_1))g$$

$$G_2 = -l_2 m_2 \cos(q_1 + q_2)g$$

Računanje matrica dinamike nalazi se u skripti *matrice_dinamike.m*

4 Planiranje trajektorija

Pri planiranju trajektorije, znajući da je vreme kretanja oba zgloba isto, prvo je potrebno naći koliko je vreme kretanja.

$$T_{UK} = T_{AB} + T_{BC}$$

Od tačke A do tačke B kretanje se realizuje trapeznim profilom brzine. Za oba zgloba smo izračunali minimalno vreme kretanja. Nakon što smo ustanovili da je vreme ubrzavanja prvog zgloba T_{a1} veće od polovine ukupnog vremena kretanja tog zgloba T_{f1} , trapezni profil za prvi zglob smo sveli na trougaoni i izračunali tražena vremena. Nakon što smo dobili minimalna vremena kretanja za oba zgloba, vreme kretanja je

$$T_{AB} = \max(T_{f1}, T_{f2})$$

Ubrzanja i brzine smo izračunali ponovo na osnovu vremena kretanja, i nakon toga odredili funkciju pozicije, brzine i ubrzanja u unutrašnjim koordinatama. Slično se određuje vreme kretanja realizovanog interpolacijom petog stepena. Skripta planiranje_trajektorija.m:

```

1  global dt
2
3  % Parametri kretanja - u radianima!
4  q_A = [0; 0];
5  q_B = [pi/4; 70*pi/180];
6  q_C = inverzni_problem_kinematike(0.25, 0.933);
7
8  % Ogranicenja brzine i ubrzanja
9  dq1_max = 1.5;
10 ddq1_max = 2;
11
12 dq2_max = 2;
13 ddq2_max = 4;
14
15 %% od A do B
16 % Trapezni profil brzine - prvi zglob
17 Ta1 = dq1_max/ddq1_max; % 0.75s
18 Tf1 = (q_B(1)-q_A(1))/dq1_max + Ta1; % 1.2736s
19 % Ta > Tf/2 za trapezni profil, pa je Tf = 2*Ta
20
21 Ta1 = sqrt((q_B(1)-q_A(1))/ddq1_max);
22 Tf1 = 2*Ta1;
23
24 % Trapezni profil brzine - drugi zglob
25 Ta2 = dq2_max/ddq2_max; % 0.5s
26 Tf2 = (q_B(2)-q_A(2))/dq2_max + Ta1; % 1.3609s
27 % Ta < Tf/2, koristimo trapez
28
29 % ukupno vreme kretanja
30 Tf = max(Tf1, Tf2);
31 T = [Tf; Tf];
32

```

```

33 % Ostvarene maksimalne brzine i ubrzanja
34 ddq1_max = ((q_B(1)-q_A(1))/(Ta1*(T(1)-Ta1)));
35 dq1_max = ddq1_max*Ta1;
36
37 ddq2_max = ((q_B(2)-q_A(2))/(Ta2*(T(2)-Ta2)));
38 dq2_max = ddq2_max*Ta2;
39
40 % Izracunavanje za prvi zglob
41 [q1_ref, dq1_ref, ddq1_ref] = trapezni_profil_brzine(Ta1, Tf, ddq1_max, q_A(1), q_B(1));
42
43 time = linspace(0, Tf, Tf/dt + 1); %vremenska osa
44 % iscrtavanje referenci za prvi zglob
45 figure();
46 subplot(3,1,1);
47 plot(time, q1_ref);
48 subplot(3,1,2);
49 plot(time, dq1_ref);
50 subplot(3,1,3);
51 plot(time, ddq1_ref);
52
53 % izracunavanje za drugi zglob
54 [q2_ref, dq2_ref, ddq2_ref] = trapezni_profil_brzine(Ta2, Tf, ddq2_max, q_A(2), q_B(2));
55
56 time = linspace(0, Tf, Tf/dt + 1); %vremenska osa
57 % iscrtavanje referenci za drugi zglob
58 figure();
59 subplot(3,1,1);
60 plot(time, q2_ref);
61 subplot(3,1,2);
62 plot(time, dq2_ref);
63 subplot(3,1,3);
64 plot(time, ddq2_ref);
65
66 % skladistenje vrednosti
67 q1_ref_A_DO_B = q1_ref;
68 q2_ref_A_DO_B = q2_ref;
69 dq1_ref_A_DO_B = dq1_ref;
70 dq2_ref_A_DO_B = dq2_ref;
71 ddq1_ref_A_DO_B = ddq1_ref;
72 ddq2_ref_A_DO_B = ddq2_ref;
73
74 Tf_A_DO_B = Tf;
75
76 %% Od B do C
77
78 % Ogranicenja brzine i ubrzanja
79 dq1_max = 1.5;
80 ddq1_max = 2;
81
82 dq2_max = 2;
83 ddq2_max = 4;
84
85 dp_max = 1.875;
86 ddp_max = 5.773;
87
88 % racunanje za prvi zglob - polinom petog stepena
89
90 Tv1 = (q_C(1)-q_B(1))/dq1_max*dp_max; % 0.3914
91 Ta1 = sqrt((q_C(1)-q_B(1))/ddq1_max*ddp_max); % 0.8690
92 Tf1 = max(Tv1, Ta1); % 0.8690
93
94 % racunanje za prvi zglob - polinom petog stepena
95
96 Tv2 = (abs(q_C(2)-q_B(2)))/dq2_max*dp_max; % 0.8067
97 Ta2 = sqrt(abs((q_C(2)-q_B(2))/ddq2_max*ddp_max)); % 1.1144
98 Tf2 = max(Tv2, Ta2); % 1.1144
99
100 Tf = max(Tf1, Tf2);

```

```

101
102 dq1_max = abs(q_C(1)-q_B(1))/Tf*dp_max;
103 ddq1_max = abs((q_C(1)-q_B(1)))/(Tf^2)*ddp_max;
104
105 dq2_max = abs(q_C(2)-q_B(2))/Tf*dp_max;
106 ddq2_max = abs((q_C(2)-q_B(2)))/(Tf^2)*ddp_max;
107
108 % izracunavanje za prvi zglob
109 [q1_ref, dq1_ref, ddq1_ref] = interpolacija_peti_stepen(Tf,q_B(1),q_C(1));
110
111 time = linspace(0,Tf,Tf/dt + 1); %vremenska osa
112 % iscrtavanje referenci za prvi zglob
113 figure();
114 subplot(3,1,1);
115 plot(time,q1_ref);
116 subplot(3,1,2);
117 plot(time,dq1_ref);
118 subplot(3,1,3);
119 plot(time,ddq1_ref);
120
121 % izracunavanje za drugi zglob
122 [q2_ref, dq2_ref, ddq2_ref] = interpolacija_peti_stepen(Tf,q_B(2),q_C(2));
123
124 time = linspace(0,Tf,Tf/dt + 1); %vremenska osa
125 % iscrtavanje referenci za drugi zglob
126 figure();
127 subplot(3,1,1);
128 plot(time,q2_ref);
129 subplot(3,1,2);
130 plot(time,dq2_ref);
131 subplot(3,1,3);
132 plot(time,ddq2_ref);
133
134 Tf_B_DO_C = Tf;
135
136 T = [Tf_A_DO_B + Tf_B_DO_C; Tf_A_DO_B + Tf_B_DO_C];
137
138 % skladistenje vrednosti
139 q1_ref_B_DO_C = q1_ref;
140 q2_ref_B_DO_C = q2_ref;
141 dq1_ref_B_DO_C = dq1_ref;
142 dq2_ref_B_DO_C = dq2_ref;
143 ddq1_ref_B_DO_C = ddq1_ref;
144 ddq2_ref_B_DO_C = ddq2_ref;
145
146
147 %% Spajanje cele trajektorije
148 q_ref = [q1_ref_A_DO_B, q1_ref_B_DO_C ; q2_ref_A_DO_B, q2_ref_B_DO_C];
149 dq_ref = [dq1_ref_A_DO_B, dq1_ref_B_DO_C ; dq2_ref_A_DO_B, dq2_ref_B_DO_C];
150 ddq_ref = [ddq1_ref_A_DO_B, ddq1_ref_B_DO_C ; ddq2_ref_A_DO_B, ddq2_ref_B_DO_C];

```

Pomoćna funkcija za trapezni profil brzine:

```

1 function [q_ref, dq_ref, ddq_ref] = trapezni_profil_brzine(Ta,Tf,ddq_max,q_i,q_f)
2 global m l l1 l2 g Fint dt
3 lengthT = Tf/dt + 1;
4
5 % racunanje q, dq i ddq
6 time = linspace(0,Tf,lengthT);
7
8 %putanje
9 q_ref = 0.*heaviside(time);
10 dq_ref = 0.*heaviside(time);
11 ddq_ref = 0.*heaviside(time);
12
13
14 % % ovo je za trougaoni

```

```

15 % q_ref(time≤Ta) = q_i+0.5*ddq_max.*time(time≤Ta).^2;
16 % q_ref(time>Ta) = q_f-0.5*ddq_max.*(Tf-time(time>Ta)).^2;
17
18 % pozicija zgloba
19 q_ref(time≤Ta) = q_i+0.5.*ddq_max.*time(time≤Ta).^2;
20 q_ref(Ta<time & time≤Tf-Ta) = q_i+ddq_max*Ta.*(time(Ta<time & time≤Tf-Ta)-Ta/2);
21 q_ref(Tf-Ta<time) = q_f-0.5*ddq_max.*(Tf-time(Tf-Ta<time)).^2;
22
23 % brzina zgloba
24 dq_ref(time≤Ta) = ddq_max.*time(time≤Ta);
25 dq_ref(time>Ta & time≤Tf-Ta) = ddq_max*Ta;
26 dq_ref(time>(Tf-Ta)) = ddq_max.*(Tf-time(time>(Tf-Ta)));
27
28 %ubrzanje zgloba
29 ddq_ref(time≤Ta) = ddq_max;
30 ddq_ref(time>Ta & time≤Tf-Ta) = 0;
31 ddq_ref(time>(Tf-Ta)) = -ddq_max;
32
33 end

```

Pomoćna funkcija za interpolaciju polinomom petog stepena:

```

1 function [q_ref, dq_ref, ddq_ref] = interpolacija_peti_stepen(Tf,q_i,q_f)
2 global m l l1 l2 g Fint dt
3 lengthT = Tf/dt + 1;
4
5 % racunanje q, dq i ddq
6 time = linspace(0,Tf,lengthT);
7
8 q_ref = q_i+(q_f-q_i)*(6*(time/Tf).^5-15*(time/Tf).^4+10*(time/Tf).^3);
9 dq_ref = (q_f-q_i)/Tf*(30*(time/Tf).^4-60*(time/Tf).^3+30*(time/Tf).^2);
10 ddq_ref = (q_f-q_i)/Tf.^2*(120*(time/Tf).^3-180*(time/Tf).^2+60*(time/Tf));
11
12
13 end

```

5 Upravljanje

Celokupno upravljanje se svodi na PID regulatore u svakom zglobu. Dakle, koristimo dva PID regulatora sa parametrima $K_p = [1500 \ 1500]^T$, $K_i = [50 \ 50]^T$, $K_d = [200 \ 200]^T$. Pošto nam je poznat vremenski interval u kom eksterna sila deluje i njen intenzitet, upravljanju uz pomoć PID-a ćemo dodati vrednost upravljanja koje kompenzuje tu silu.

```

1 function [Tau,Tau_FF,Tau_FB] = kontroler(q, q_ref, dq, dq_ref, ddq, ddq_ref, Kp, ...
    Kd,Ki, H, C, G, J, F_ext)
2 global m l l g sum_e e_p mt
3
4 % H11n = 2 * (0.5*m*l^2 + 0.5*m*(l/2)^2 + 0.5*m*(l/2)^2 + 0.5*I);
5 % H22n = (l/2)^2*m + I;
6 % H_nadvuceno = [H11n,0; 0,H22n];
7 % Fv = [0,0; 0,0];
8 % Tau_FF = H_nadvuceno*ddq;
9
10 F_ext1 = F_ext - [0;-mt*g]; %eksterna sila, ne racunajuci dodatni teret
11 Tau_FF = -J'*F_ext1; %upravljanje kojim kompenzujemo eksternu silu
12
13 Tau_FB = [0; 0]; %PID upravljanje kojim ostvarujemo pracenje reference
14 e = q_ref-q;
15 sum_e = sum_e+e;
16 for j=1:2
17 % feedback PID control
18 Tau_FB(j) = Kp(j)*(q_ref(j)-q(j)) + Kd(j)*(dq_ref(j)-dq(j)) + ...
    Ki(j)*sum_e(j)*0.001;

```

```

19     end
20     e_p = e;
21
22     Tau = Tau_FF + Tau_FB; %ukupno upravljanje
23 end

```

6 Main, integracija i iscrtavanje

```

1  function dydt_Q_4 = int_2DoF(t, Q_4)
2  % Q_4(1) = q(1)
3  % Q_4(2) = q(2)
4  % Q_4(3) = dq(1)
5  % Q_4(4) = dq(2)
6
7  global Tau F_ext
8
9  %modeliranje dinamike robota
10
11 % izracunavamo Jakobijan matricu
12 J = jakobijan_matrica([Q_4(1); Q_4(2)]);
13
14 % izracunavamo matrice dinamike H, C, G
15 [H,C,G] = matrice_dinamike([Q_4(1); Q_4(2)], [Q_4(3); Q_4(4)]);
16
17 % integracija
18 dydtQ_2 = H \ (Tau + J'*F_ext - C*[Q_4(3);Q_4(4)] - G);
19
20 %izlaz
21 dydt_Q_4(1:2) = Q_4(3:4);
22 dydt_Q_4(3:4) = dydtQ_2;
23 dydt_Q_4 = dydt_Q_4';
24 end

```

```

1  clear
2  close all
3  clc
4
5  global l l1 l2 g F_ext Tau sum_e e_p mt
6
7  %%
8  parametri_simulacije;
9
10 %%
11 planiranje_trajektorija;
12
13 %% PID regulator
14 PID_parametri;
15
16 %% glavna petlja
17 i = 1; %inicijalizacija brojaca
18 sum_e = 0; %inicijalizacija integrala greske
19 e_p = 0; %inicijalizacija greske
20
21 figure();
22 while (t < T(1))
23     t = i*dt; %trenutno vreme
24
25     %odredjivanje sile F_ext
26     if (t ≥ 0.9*T(1))
27         F_ext = [0; -mt*g] + [-2; -4];
28     end
29

```

```

30 %izracunavanje matrice J
31 J = jakobijan_matrica(q); %na osnovu ostvarenih pozicija
32 J_ref = jakobijan_matrica(q_ref(:,i)); %na osnovu referentnih pozicija
33
34 %izracunavanje referentnih i trenutnih vrednosti u spoljasnjim koordinatama
35 % na osnovu ostvarenih pozicija
36 [X, dX] = direktna_kinematika(q,dq,J);
37 % na osnovu referentnih pozicija
38 [X_ref, dX_ref] = direktna_kinematika(q_ref(:,i),dq_ref(:,i),J_ref);
39
40 %izracunavanje matrica dinamike
41 % H, C, G matrice estimacije
42 [H,C,G] = matrice_dinamike(q,dq);
43
44 %projektovanje kontrolera
45 [Tau,Tau_FF,Tau_FB] = ...
    kontroler(q,q_ref(:,i),dq,dq_ref(:,i),ddq,ddq_ref,Kp,Kd,Ki,H,C,G,J,F_ext);
46
47 %numericka integracija
48 Q_4 = [q; dq];
49 options = odeset('RelTol',1e-2,'AbsTol',1e-3,'MaxOrder',3);
50 [tout,Q_4_out] = ode45(@int_2DoF,[t t+dt], Q_4, options);
51 Q_4 = Q_4_out(end,:);
52 q = Q_4(1:2);
53 dq = Q_4(3:4);
54
55 %ubrzanje za sledecu iteraciju na osnovu jednacine dinamike
56 ddq = H \ (Tau(:,end) + J'*F_ext - C*[dq(1); dq(2)] - G);
57
58 % simulacija
59 if mod(i,30)==0
60     plot([0 11*cos(q(1)) 11*cos(q(1))+12*cos(q(1)+q(2))], [0 11*sin(q(1)) ...
        11*sin(q(1))+12*sin(q(1)+q(2))], ...
        '--rs','LineWidth',2,'MarkerEdgeColor','k','MarkerFaceColor','g','MarkerSize',10)
61
62     axis equal;
63     axis([-2*1 2*1 -2*1 2*1]);%definisanje osa
64     title('2DoF robot')
65     ylabel('z position [m]')
66     xlabel('x position [m]')
67     grid
68     pause(0.001)
69 end
70
71 write_in_memory; %upis u memoriju vrednosti od znacaja iz ove iteracije
72 i = i+1; %uvecavamo brojac
73 end
74
75 iscrtavanje;

```

```

1 close all;
2
3 % q1_ref, q1
4 figure()
5 plot(Ps.t,Ps.q_ref(1,:),Ps.t,Ps.q(1,:), 'r','LineWidth',2)
6 grid;
7 title('Unutrasnja koordinata q1-pozicija prvog zgloba');
8 xlabel('vreme [s]');
9 ylabel('stepeni [dg]');
10 legend('referentna vrednost','ostvarena vrednost');
11
12 % q2_ref, q2
13 figure()
14 plot(Ps.t,Ps.q_ref(2,:),Ps.t,Ps.q(2,:), 'r','LineWidth',2)
15 grid;
16 title('Unutrasnja koordinata q2-pozicija drugog zgloba');
17 xlabel('vreme [s]');

```



```

18 ylabel('stepeni [dg]');
19 legend('referentna vrednost','ostvarena vrednost');
20
21 % dq1_ref, dq1
22 figure()
23 plot(Ps.t,Ps.dq_ref(1,:),Ps.t,Ps.dq(1,:), 'r','LineWidth',2)
24 grid;
25 title('Unutrasnja koordinata dq1-brzina prvog zgloba');
26 xlabel('vreme [s]');
27 ylabel('stepeni/sekund [dg/s]');
28 legend('referentna vrednost','ostvarena vrednost');
29
30 % dq2_ref, dq2
31 figure()
32 plot(Ps.t,Ps.dq_ref(2,:),Ps.t,Ps.dq(2,:), 'r','LineWidth',2)
33 grid;
34 title('Unutrasnja koordinata dq2-brzina drugog zgloba');
35 xlabel('vreme [s]');
36 ylabel('stepeni/sekund [dg/s]');
37 legend('referentna vrednost','ostvarena vrednost');
38
39 % X_ref(1), X(1)
40 figure()
41 plot(Ps.t,Ps.X_ref(1,:),Ps.t,Ps.X(1,:), 'r','LineWidth',2)
42 grid;
43 title('Spoljasnja koordinata x-pozicija na x osi');
44 xlabel('vreme [s]');
45 ylabel('pozicija [m]');
46 legend('referentna vrednost','ostvarena vrednost');
47
48 % X_ref(2), X(2)
49 figure()
50 plot(Ps.t,Ps.X_ref(2,:),Ps.t,Ps.X(2,:), 'r','LineWidth',2)
51 grid;
52 title('Spoljasnja koordinata z-pozicija na z osi');
53 xlabel('vreme [s]');
54 ylabel('pozicija [m]');
55 legend('referentna vrednost','ostvarena vrednost');
56
57 % pogonski moment prvog zgloba Tau(1)
58 figure()
59 plot(Ps.t,Ps.Tau(1,:), 'LineWidth',2)
60 grid;
61 title('Ostvareni pogonski moment prvog zgloba');
62 xlabel('vreme [s]');
63 ylabel('moment [Nm]');
64
65 % pogonski moment drugog zgloba Tau(2)
66 figure()
67 plot(Ps.t,Ps.Tau(2,:), 'LineWidth',2)
68 grid;
69 title('Ostvareni pogonski moment drugog zgloba');
70 xlabel('vreme [s]');
71 ylabel('moment [Nm]');
72
73 % kretanje u X-Z ravni
74 figure()
75 plot(Ps.X_ref(1,:),Ps.X_ref(2,:),Ps.X(1,:),Ps.X(2,:), 'r','LineWidth',2)
76 grid;
77 title('Kretanje robota u X-Z ravni');
78 xlabel('X [m]');
79 ylabel('Z [m]');
80 legend('referentna vrednost','ostvarena vrednost');
81
82 % dX_ref(1), dX(1)
83 figure()
84 plot(Ps.t,Ps.dX_ref(1,:),Ps.t,Ps.dX(1,:), 'r','LineWidth',2)
85 grid;

```

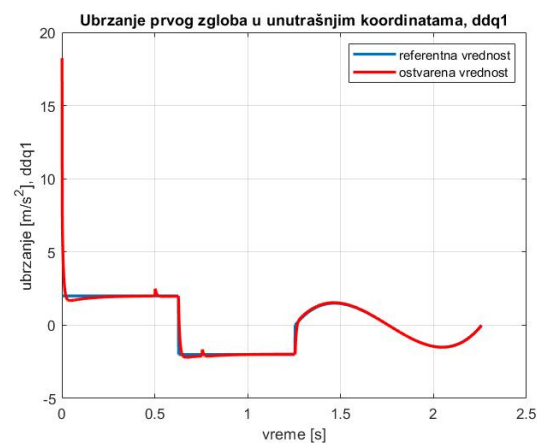
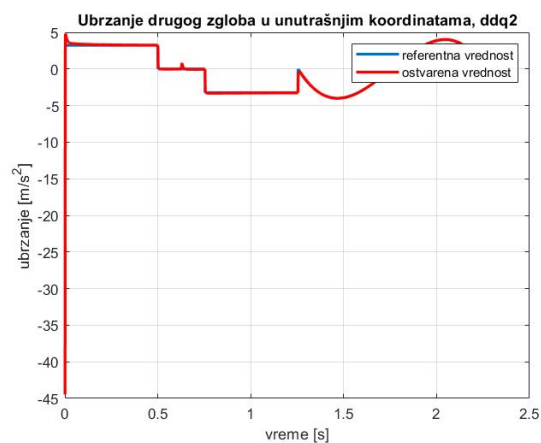
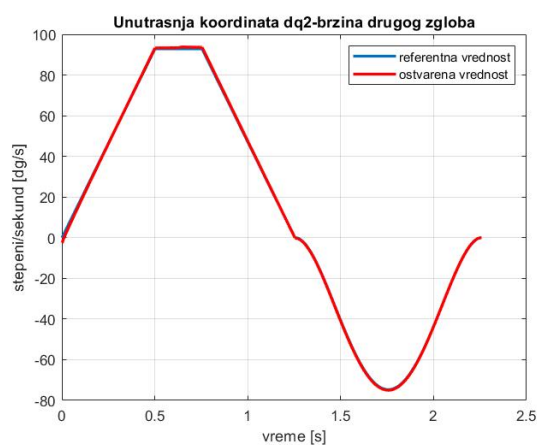
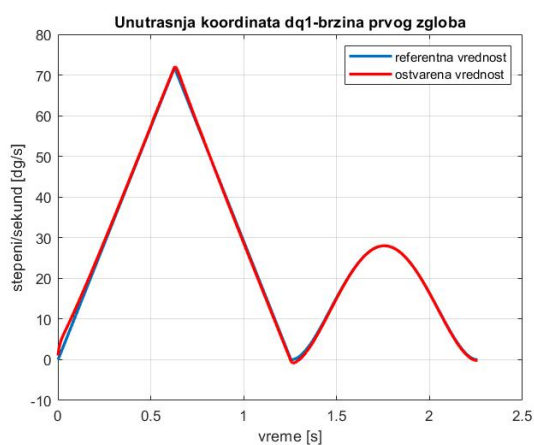
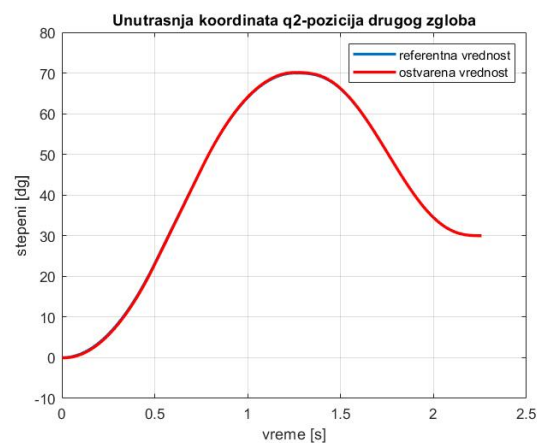
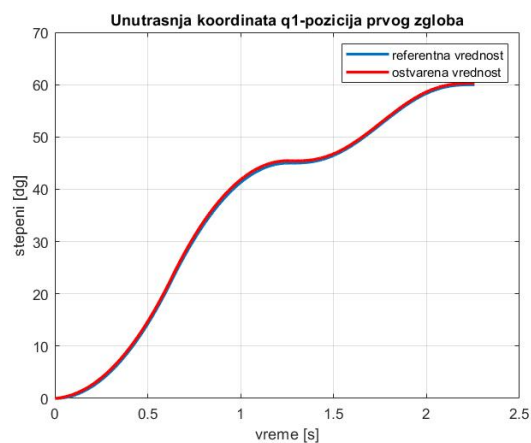
```

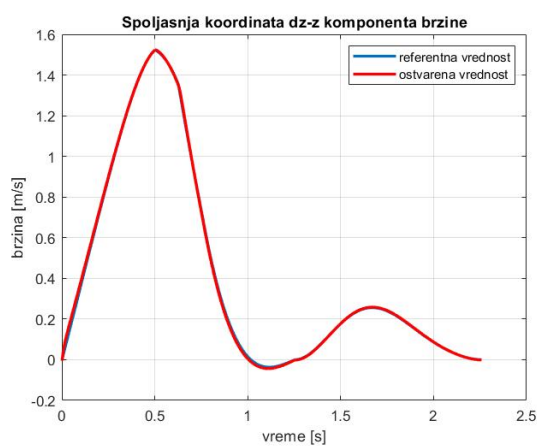
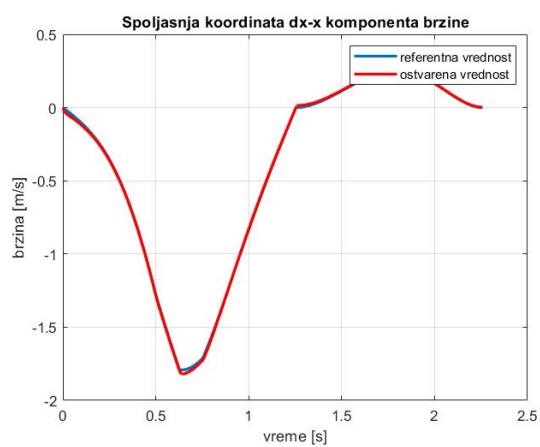
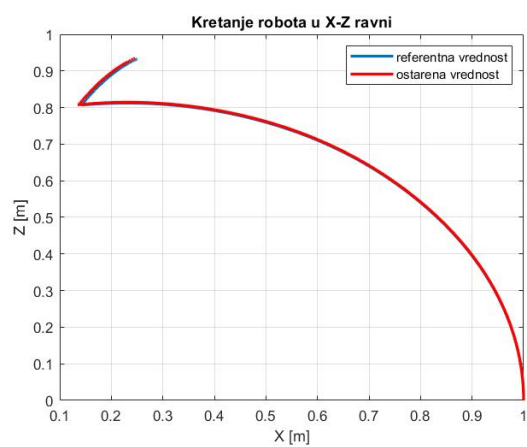
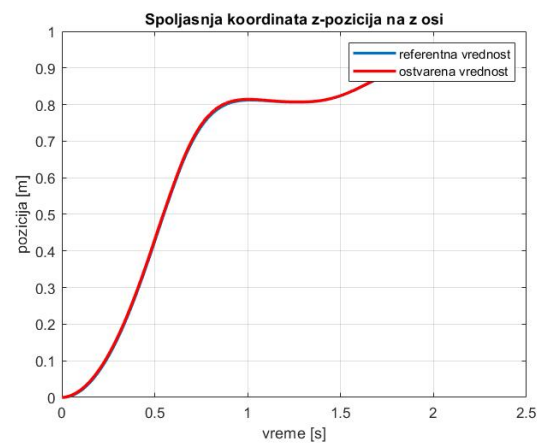
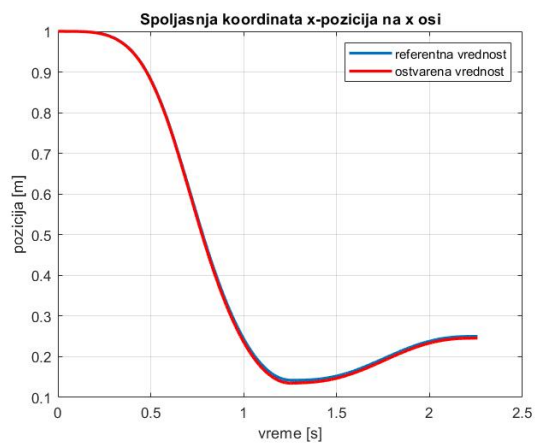
86 title('Spoljasnja koordinata dx-x komponenta brzine');
87 xlabel('vreme [s]');
88 ylabel('brzina [m/s]');
89 legend('referentna vrednost','ostvarena vrednost');
90
91 % dX_ref(2), dX(2)
92 figure()
93 plot(Ps.t,Ps.dX_ref(2,:),Ps.t,Ps.dX(2,:), 'r','LineWidth',2)
94 grid;
95 title('Spoljasnja koordinata dz-z komponenta brzine');
96 xlabel('vreme [s]');
97 ylabel('brzina [m/s]');
98 legend('referentna vrednost','ostvarena vrednost');
99
100 % ddq_ref(1), ddq(1)
101 figure()
102 plot(Ps.t,Ps.ddq_ref(1,:),Ps.t,Ps.ddq(1,:), 'r','LineWidth',2)
103 grid;
104 title('ubrzanje');
105 xlabel('vreme [s]');
106 ylabel('ubrzanje [m/s^2], ddq1');
107 legend('referentna vrednost','ostvarena vrednost');
108
109 % ddq_ref(2), ddq(2)
110 figure()
111 plot(Ps.t,Ps.ddq_ref(2,:),Ps.t,Ps.ddq(2,:), 'r','LineWidth',2)
112 grid;
113 title('ubrzanje');
114 xlabel('vreme [s]');
115 ylabel('ubrzanje [m/s^2], ddq2');
116 legend('referentna vrednost','ostvarena vrednost');
117
118 % kompenzacija eksterne sile F_ext
119 figure()
120 subplot(2,1,1)
121 plot(Ps.t,Ps.Tau_FF(1,:), 'LineWidth',2)
122 grid;
123 title('Kompenzacija F_ext za prvi zglob');
124 xlabel('vreme [s]');
125 ylabel('Momenat');
126 subplot(2,1,2)
127 plot(Ps.t,Ps.Tau_FF(2,:), 'LineWidth',2)
128 grid;
129 title('Kompenzacija F_ext za drugi zglob');
130 xlabel('vreme [s]');
131 ylabel('Momenat');
132
133 % PID upravljanje
134 figure()
135 subplot(2,1,1)
136 plot(Ps.t,Ps.Tau_FB(1,:), 'LineWidth',2)
137 grid;
138 title('PID vrednost upravljanja za prvi zglob');
139 xlabel('vreme [s]');
140 ylabel('Momenat');
141 subplot(2,1,2)
142 plot(Ps.t,Ps.Tau_FB(2,:), 'LineWidth',2)
143 grid;
144 title('PID vrednost upravljanja za drugi zglob');
145 xlabel('vreme [s]');
146 ylabel('Momenat');

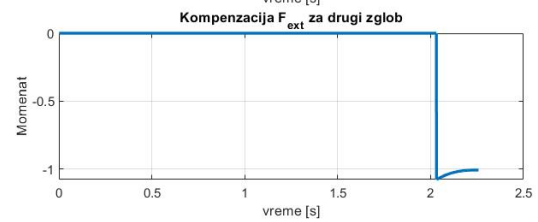
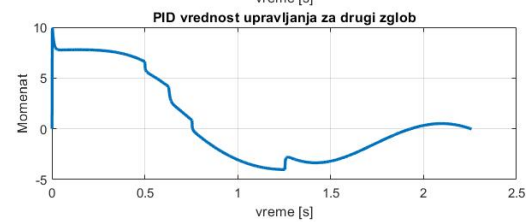
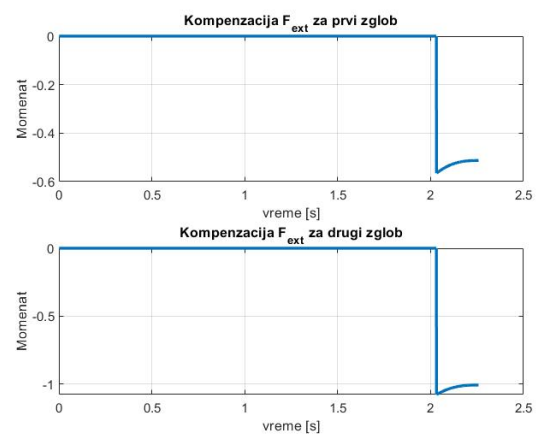
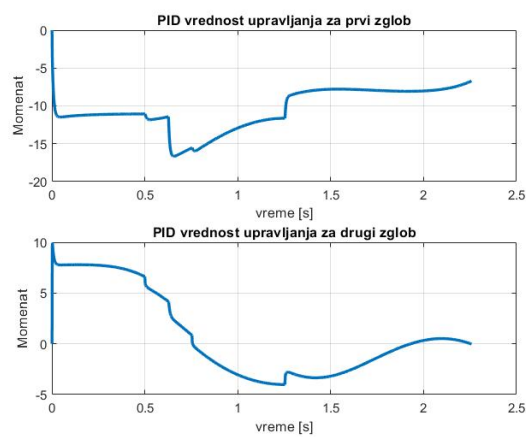
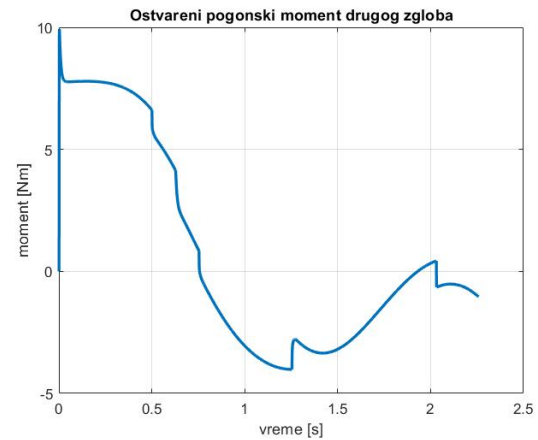
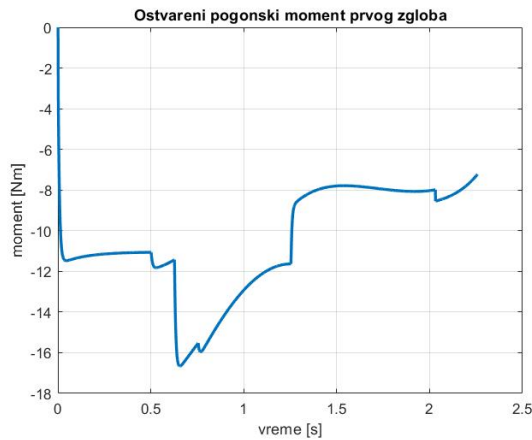
```

7 Grafici

Na sledećim graphicima prikazani su rezultati simulacije:







Pošto se referenca ubrzanja skokovito menja na samom početku, komponenta upravljanja koja potiče od diferencijalnog dejstva regulatora će biti velikog intenziteta. To se može uočiti na graficima ubrzanja u unutrašnjim koordinatama, kao i na graficima upravljanja. Komponenta koja kompenzuje eksternu silu se javlja samo kada eksterna sila deluje. Vidimo da sistem dobro prati reference pozicije i brzine (u unutrašnjim i spoljašnjim koordinatama).