

Osnove robotike

LV2 Inverzna kinematika

Denis Lazor

Zadatak: Napraviti funkciju u MATLAB-u koja će za zadanu matricu homogene transformacije između alata i baznog koordinatnog sustava T60 vratiti popis kuteva zglobova potrebnih da se robotska ruka postavi u pravilan položaj za hvatanje zadanog predmeta.

Manipulator: Motoman MH50-35

Model: Nacrt i bokocrt u AutoCad-u skalirani s 0.1 uz poštivanje vidljivih dimenzija i odokativnog određivanja ostalih. Na modelu se nalaze i označeni koordinatni sustavi zajedno sa svim njihovim osima.

Dijelovi MATLAB skripti:

➔ *my_robot_script_inv.m* skripta

Zadavanje položaja i dimenzija predmeta:

```
ta0 = [20, -73, -48.5];  
➔ [x, y, z] -- x, y koordinate skoro proizvoljne (u području maksimalnog dohvata i bez preklapanja  
s bazom robota), z koordinata fiksna (na istoj površini kao i robot)  
  
A=20; B=11; C=10;
```

Zadavanje kuta rotacije predmeta oko z osi:

```
obj_alpha = 0;  
  
ca = cos(obj_alpha);  
sa = sin(obj_alpha);
```

Matrica transformacije iz koordinatnog sustava 6 u koordinatni sustav predmeta A . Rotacija oko x6(hvatanje predmeta s gornje strane) i translacija po z(prislanjanje predmeta na dno alata):

```
T6A=[1 0 0 0;  
      0 -1 0 0;  
      0 0 -1 -5;  
      0 0 0 1];
```

Rotacijska matrica objekta s obzirom na bazni koordinatni sustav:

```
RA0 = [ca -sa 0  
       sa ca 0  
       0 0 1];
```

Matrica transformacije iz koordinatnog sustava objekta u bazni koord. sustav:

```
TA0_ = [RA0, (ta0)'];  
TA0 = [TA0_; [0 0 0 1]];
```

Matrica transformacije iz koordinatnog sustava 6(alat) u bazni koord. sustav:

```
T60 = TA0 * T6A;
```

Pozivanje funkcije za izračun theta parametara pomoću inverzne kinematike:

```
q=my_invkin(T60);
```

Dodavanje predmeta na scenu koristeći zelenu boju (dodan parameter u postojeću plot3dobj() funkciju):

```
obj=cuboid(A,B,C);  
obj.X=TA0*obj.X;  
hold on;  
plot3dobj(obj, 'g');
```

→ *my_invkin.m* skripta

- Rješavanjem problema inverzne kinematike koristeći Pieperovo rješenje došlo se do jednadžbi za izračun theta parametara svih zglobova, iste su implementirane u prikazanoj m-skripti. Uzeto je u obzir samo jedno od mogućih rješenja.

```
function q = my_invkin(T60)

% p(1,2,3) = x ,y ,z koordinata sjecišta zadnja 3 zglobova
p = T60 * [0 0 -27.25 1]';
x=p(1); y=p(2); z=p(3);

% Theta 3
syms u3 'real';
c3 = (1-u3^2)/(1+u3^2);
s3 = 2 * u3 / (1+u3^2);

% Substitutions (theta 3)
f1 = 84.75*s3 + 27.6*c3 + 84;
f2 = 27.6*s3 - 84.75*c3;
a1 = 8.2526;
L = x^2+y^2+z^2;
t = f1^2+f2^2;
left = L^2-2*L*a1^2+(4*a1^2*z^2)+a1^4;
right = t*(2*a1^2+2*L)-t^2;

u3 = double(solve(left == right));
theta3 = 2*atan(u3(1));

% Theta 2
syms u2 'real';

c2 = (1-u2^2)/(1+u2^2);
s2 = 2 * u2 / (1+u2^2);

s3 = sin(theta3);
c3 = cos(theta3);

f1 = 84.75*s3 + 27.6*c3 + 84;
f2 = 27.6*s3 - 84.75*c3;

u2=double(solve(z-f1*s2-f2*c2));
theta2 = 2*atan(u2(2));
```

```

% Theta 1
syms c1 s1 'real';

s2 = sin(theta2);
c2 = cos(theta2);

g1 = c2*f1 - s2*f2 + a1;
eqn = [x == c1*g1, y == s1*g1]; % g2 = 0

[c,s] = solve(eqn,[c1 s1]);
c = double(c);
s = double(s);
theta1 = atan2(s,c);

% Matrices
T10 = dhtransf([theta1 0 8.2526 pi/2]);
T21 = dhtransf([theta2 0 84 0]);
T32 = dhtransf([theta3 0 27.6 pi/2]);
T30 = T10*T21*T32;
R30 = T30(1:3,1:3);
R60 = T60(1:3,1:3);
R63 = R30'*R60;
pao=(T30 * [0 0 84.75 1]')';

if single(pao(1))==x | single(pao(1))==y;
    disp("Catch ya!");
else
    disp("Unable to catch with these set of parameters");
end

% Theta 5
theta5 = acos(R63(3,3));

% Theta 4,6 (depending on theta 5)
if sin(theta5) ~= 0
    theta4 = atan2(-R63(2,3)/sin(theta5),-R63(1,3)/sin(theta5));
    theta6 = atan2(-R63(3,2)/sin(theta5),R63(3,1)/sin(theta5));
elseif cos(theta5) == 1
    theta4 = 0;
    theta6 = atan2(R63(2,1),R63(1,1));
else
    theta4 = 0;
    theta6 = atan2(R63(2,2),R63(1,2));
end

% Final theta vector(one of solutions)
q = [theta1 theta2 theta3 theta4 theta5 theta6];

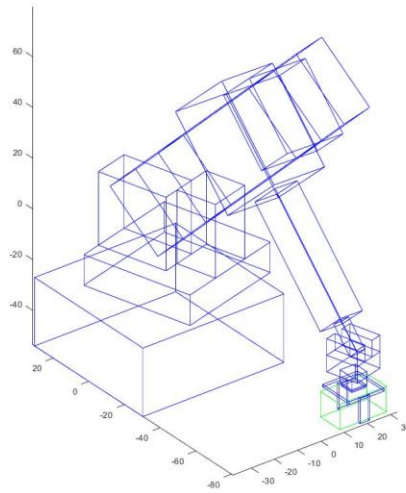
end

```

- Rezultati su provjereni računski (uspoređivanje određenih matrica) i grafički. Slijedi grafički prikaz rezultata za određene položaje predmeta izražene pomoću $ta0$ matrice i kuta rotacije oko z-osi:

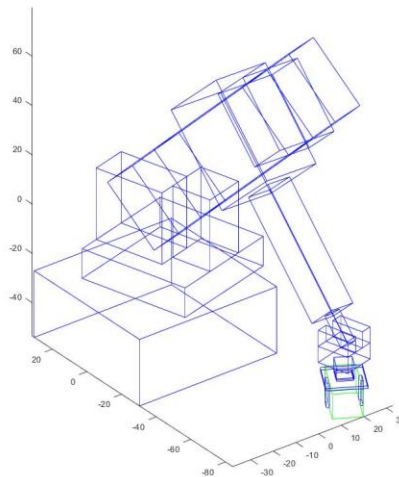
Položaj 1:

```
ta0 = [20,-73,-48.5];  
obj_alpha = 0;
```



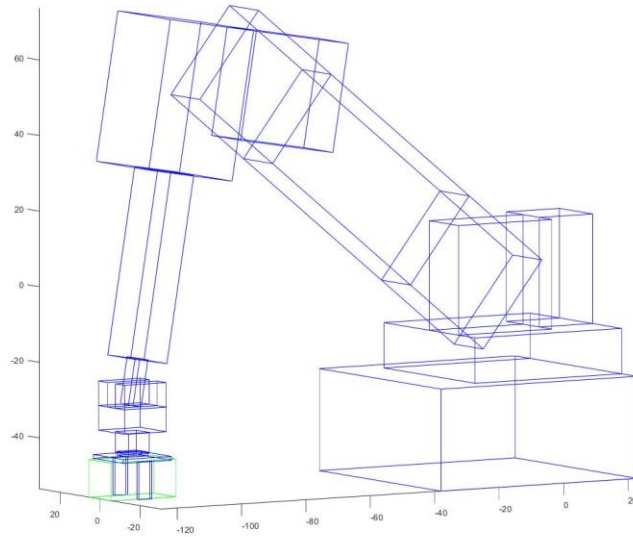
Položaj 2:

```
ta0 = [20,-73,-48.5];  
obj_alpha = pi/3;
```



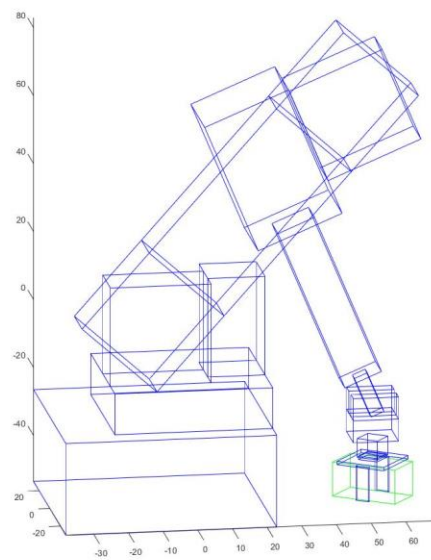
Položaj 3:

```
ta0 = [-115,0,-48.5];  
obj_alpha = pi;
```



Položaj 4:

```
ta0 = [55,0,-48.5];  
obj_alpha = -pi/5;
```



- **Pokus nije uspio za sve položaje predmeta za sve kombinacije parametara. Pokušavanjem omogućavanja hvatanja s donje strane i saznanjem o položaju alata kada nije mogao uhvatiti predmet ni s gornje ni donje strane (tik uz predmet) došlo se do zaključka da je posljedica netočnih rješenja vjerovatno ograničenje hvatanja predmeta samo s gornje strane.**

Rjesenje:

```
if single(pao(1))==x & single(pao(2))==y;  
disp("Catch ya!");  
  
else  
disp("Unable to catch with these set of parameters");  
end
```

- **Ostatak koda se nalazi u skripti:**
my_robot_script_inv.m

