به نام نور

تمرین شماره 4 رباتیک

دانشجو:ریحانه نیکوبیان

شماره دانشجویی:99106747

سال تحصیلی:1402

۱. در نرم افزار متلب تابعی بسازید که حل سینماتیک مستقیم (H) یک ربات فضایی را در قالب جدول دناویت-هارتنبرگ و بردار متغیرهای مفصل (Q) به صورت سمبولیک دریافت کرده و ماتریس‌های ژاکوبی انتقالی (Jv) و دورانی (Jw) را به صورت سمبولیک بازگرداند، بکمک این تابع، ژاکوبی **ربات‌های تمرین های ۲ و ۳** را تولید و به همراه گزارشی از فرآیند حل و کد متلب ارائه نمایید.

در متلب کدی می نویسیم که متغیرهای مفصلی و جدول دنویت هارتنبرگ را ورودی بگیرد و ماتریس همگن، jv و jw را خروجی دهد.

کد به این شکل کار می کند که ابتدا به روش دنویت هارتنبرگ Hn_0 را تولید می کند.(ماتریس همگن همانند تمرین های قبل تولید می شود.)

در مرحله بعد می دانیم برای محاسبه jv باید بردارO0On را تولید کنیم، و ان را از Hn_0 استخراج می کنیم.

$$\vec{v}_e = \underbrace{\begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} & \cdots & \frac{\partial x}{\partial \theta_n} \\ \vdots & & & \\ \frac{\partial z}{\partial \theta_1} & \cdots & \cdots & \frac{\partial z}{\partial \theta_n} \end{bmatrix}}_{J_v}$$

حال که x,y,z را داریم می دانیم ژاکوبین سرعت انتقالی به روش بالا تولید می شود. کد مربوط را می نویسیم. باید از هر کدام مولفه های O0On نسبت به متغیر های مفصلی مشتق گرفت و ماتریس را پر کرد.

در مرحله بعد برای محاسبه ژاکوبین سرعت دورانی می دانیم باید R0_n را از Hn_0 استخراج کنیم.و بعد ژاکوبین هست:

$$\vec{\omega}_n = \underbrace{\begin{bmatrix} \sum_{j=1}^{3} \frac{\partial r_{3j}}{\partial \theta_1} r_{2j} & \cdots & \sum_{j=1}^{3} \frac{\partial r_{3j}}{\partial \theta_n} r_{2j} \\ \sum_{j=1}^{3} \frac{\partial r_{1j}}{\partial \theta_1} r_{3j} & \cdots & \sum_{j=1}^{3} \frac{\partial r_{1j}}{\partial \theta_n} r_{3j} \\ \sum_{j=1}^{3} \frac{\partial r_{2j}}{\partial \theta_1} r_{1j} & \cdots & \sum_{j=1}^{3} \frac{\partial r_{2j}}{\partial \theta_n} r_{1j} \end{bmatrix}}_{J_\omega} \dot{\Theta}$$

برای هر سطر یک forمی نویسیم که و به ازای هر متغیر مفصلی، باید از هر سطر نسبت به یک متغیر مشتق گرفت و در یک سطر دیگر ضرب و مجموعش می شود یکی از درایه ها.

نهایت کد به شکل زیر است:

ماتریس ژاکوبین

```
function[H0_n,jv,jw]=jacob(denevit,mafsal)
 syms H0_n
 for i=1:size(denevit,1)
    if i==1

H0_n=Rot('z',denevit(i,1))*Trans('Z',denevit(i,2))*Rot('x',denevit(i,3))*Trans('x',denevit(i,4));
    else

H0_n=H0_n*Rot('z',denevit(i,1))*Trans('Z',denevit(i,2))*Rot('x',denevit(i,3))*Trans('x',denevit(i
,4));
    end
 end

 syms O0Oe jv;
 O0Oe=[H0_n(1,4);H0_n(2,4);H0_n(3,4)];

 for j=1:3
    for k=1:size(mafsal,1)
```

```
      jv(j,k)=diff(OOOe(j),mafsal(k));
   end
end

syms RO_n jw;
RO_n=HO_n(1:3,1:3);

for k=1:size(mafsal,1)
    jw(1,k)=0;
    for j=1:3
          jw(1,k)=jw(1,k)+diff(RO_n(3,j),mafsal(k))*RO_n(2,j);
    end
end

for k=1:size(mafsal,1)
    jw(2,k)=0;
    for j=1:3
          jw(2,k)=jw(2,k)+diff(RO_n(1,j),mafsal(k))*RO_n(3,j);
    end
end

for k=1:size(mafsal,1)
    jw(3,k)=0;
    for j=1:3
          jw(3,k)=jw(3,k)+diff(RO_n(2,j),mafsal(k))*RO_n(1,j);
    end
end
end
```

ماتریس Rot

```
function T=Rot(axis,angle)
  axis=upper(axis);
  if (axis=='X')
    T=[1,0,0,0;
        0,cos(angle),-sin(angle),0;
        0,sin(angle),cos(angle),0;
      0,0,0,1];
  end
  if (axis=='Y')
    T=[cos(angle),0,sin(angle),0;
        0,1,0,0;
        -sin(angle),0,cos(angle),0;
        0,0,0,1];
  end
  if (axis=='Z')
    T=[cos(angle),-sin(angle),0,0;
        sin(angle),cos(angle),0,0;
        0,0,1,0;
        0,0,0,1];
```

```
  end
end
```

ماتریس Trans

```
function T=Trans(axis, distance)

  axis=upper(axis);
  if (axis == 'X')
    T=[1 0 0 distance; 0 1 0 0; 0 0 1 0; 0 0 0 1];
  end
  if (axis == 'Y')
    T=[1 0 0 0; 0 1 0 distance; 0 0 1 0; 0 0 0 1];
  end
  if (axis =='Z')
    T=[1 0 0 0; 0 1 0 0; 0 0 1 distance; 0 0 0 1];
  end
end
```

حال به سراغ بخش بعدی خواسته می رویم ، تولید ژاکوبین مربوط به تمرین های 2 ، 3

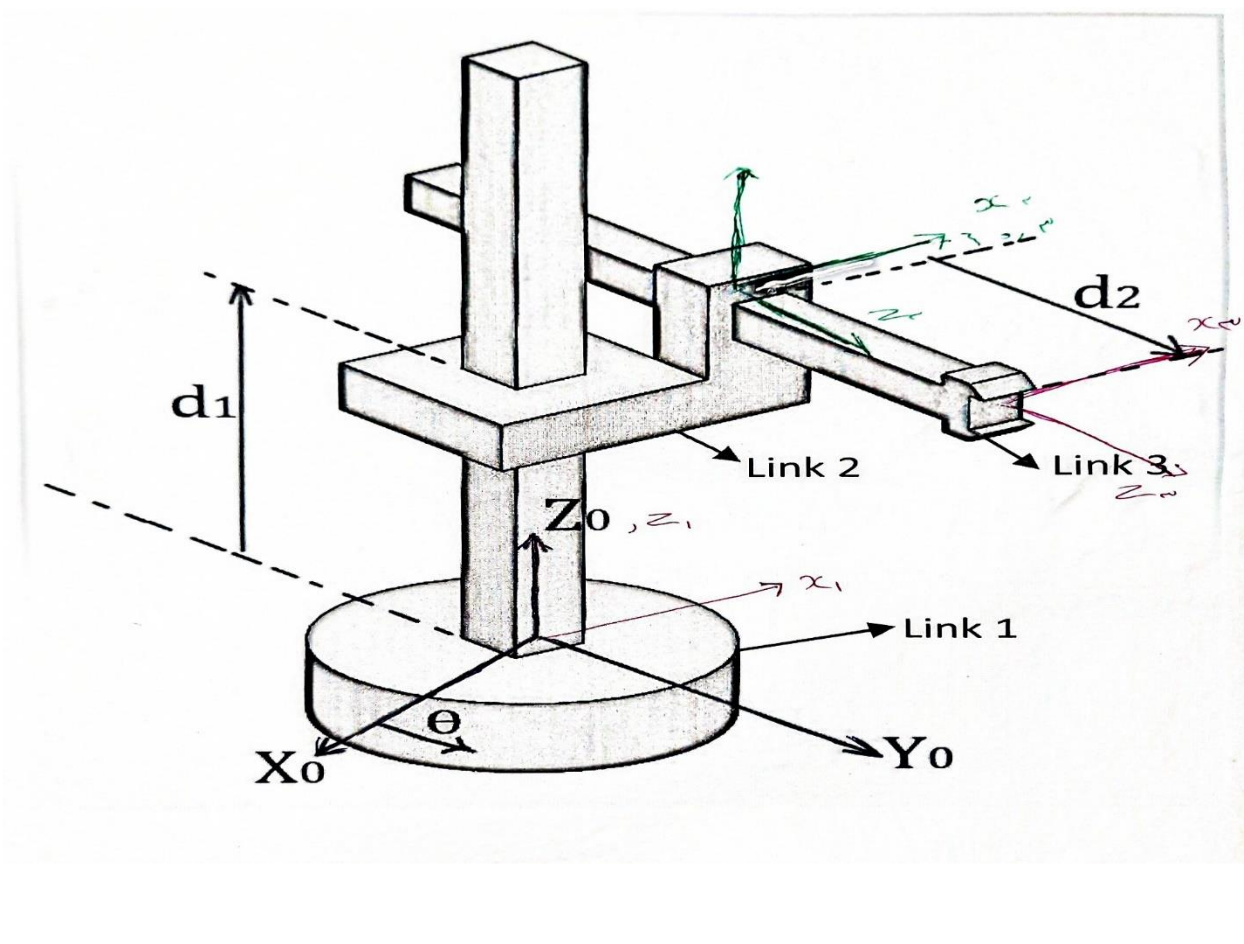
در ورودی جدول دنویت هارتنبرگ مربوط به کد و متغیرهای مفصلی را می دهیم.و تابع Jacob فرا خوانده می شود و ژاکوبین ها را برای ما محاسبه می کند:

کد مربوط به ربات تمرین 2:

```
syms denevit jv jw H0_n;
syms theta1 d1 d2;


denevit=[theta1+pi/2,0,0,0;
  0,d1+22.5,pi/2,45;
   0,d2+30,0,0];

mafsal=[theta1;d1;d2];
[H0_n,jv,jw]=jacob(denevit,mafsal);

jv=simplify(jv);
jw=simplify(jw);

disp('jw');
disp(jw);

disp('jv');
disp(jv);
```

```
jw
[0, 0, 0]
[0, 0, 0]
[1, 0, 0]

jv
[- 45*cos(theta1) - 30*sin(theta1) - d2*sin(theta1), 0, cos(theta1)]
[  30*cos(theta1) - 45*sin(theta1) + d2*cos(theta1), 0, sin(theta1)]
[                                                 0, 1,           0]
```

تنها تفاوت دستگاه های مختصات مربوط به کد نسبت به شکل پایین این است که در موقعیت صفر ایکس دستگاه مختصات صفر موازی لینک سه قرار می گیرد



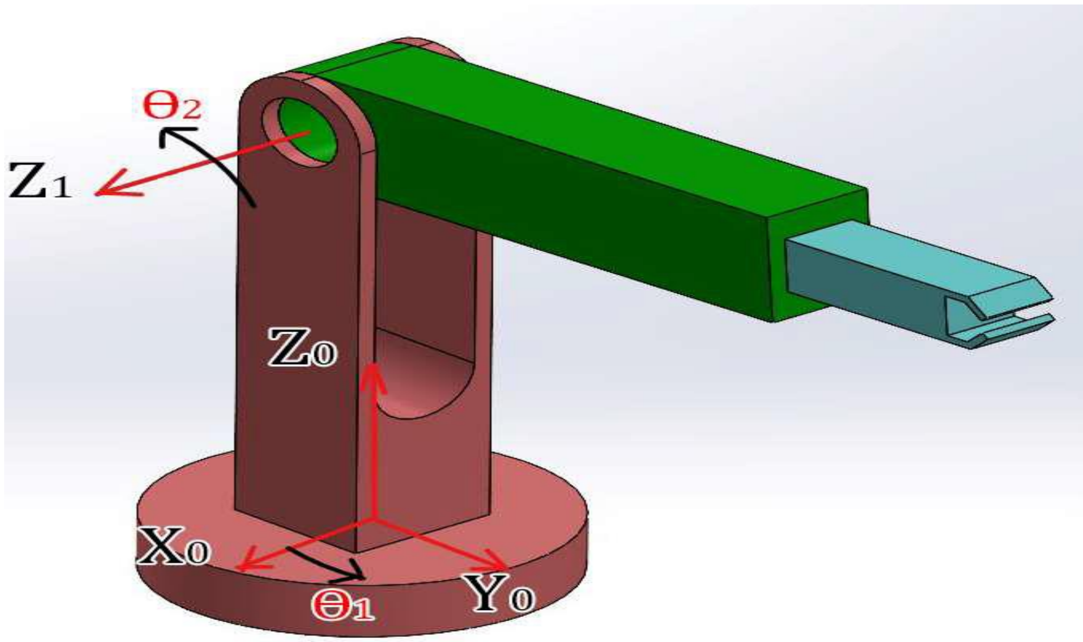
کد مربوط به ربات تمرین 3:

```
syms denevit jv jw H0_n speed;
syms theta1 theta2 d3;

denevit=[theta1,31,pi/2,0;
   pi/2+theta2,0,pi/2,0;
   0,d3+30,0,0];

mafsal=[theta1;theta2;d3];
[H0_n,jv,jw]=jacob(denevit,mafsal);

jv=simplify(jv);
jw=simplify(jw);

disp('jw');
disp(jw);

disp('jv');
disp(jv);
```

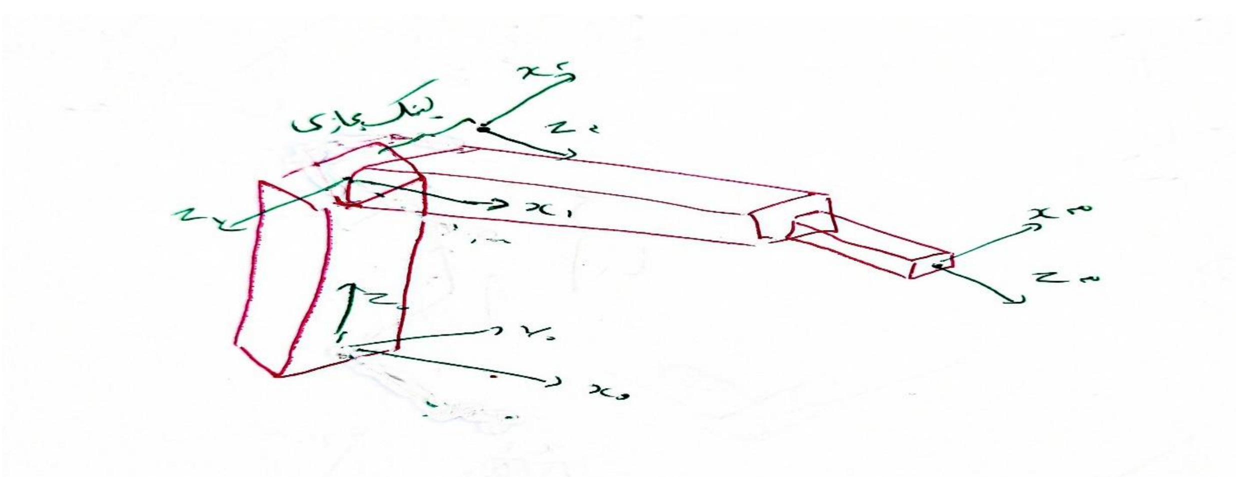
jw
[0, sin(theta1), 0]
[0, -cos(theta1), 0]
[1,   0, 0]

jv
[-cos(theta2)*sin(theta1)*(d3 + 30), -cos(theta1)*sin(theta2)*(d3 + 30), cos(theta1)*cos(theta2)]
[cos(theta1)*cos(theta2)*(d3 + 30), -sin(theta1)*sin(theta2)*(d3 + 30), cos(theta2)*sin(theta1)]
[0,   cos(theta2)*(d3 + 30),   sin(theta2)]

تنها تفاوت دستگاه های مختصات مربوط به این کد نسبت به شکل پایین این است که در موقعیت صفر ایکس دستگاه مختصات صفر موازی لینک سبز قرار می گیرد



شکل دستگاه مختصات ربات 3:



۲. نقاط تکین را در صورت وجود بصورت پارامتریک در **رباتهای تمرین ۲ و ۳** بدست آورید و نتایج را به همراه گزارش ارائه نمایید. (**امتیازی ۱۰٪**)

از انجا که عملگر نهایی سرعت خطی تولید می کند بنابراین نقاط تکین با توجه به ژاکوبین سرعت خطی به دست می ایند. دترمینان ژاکوبین های سرعت خطی ربات تمرین 2 و 3 به شرح زیر به دست می اید:

ربات تمرین 2:

```
a=det(jv);
equ=a==0;
s=simplify(a);

disp('detjv');
disp(s);
```

```
detjv
d2 + 30
```

می دانیم نقاط تکین نقاطی هستند که دترمینان ماتریس ژاکوبین مربوطط صفر شود. بنابراین نقطه تکین در ربات 2 می شود: d2=-30mm

ربات تمرین 3:

```
a=det(jv);
equ=a==0;
s=simplify(a);

disp('detjv');
disp(s);
```

```
detjv
cos(theta2)*(d3 + 30)^2
```

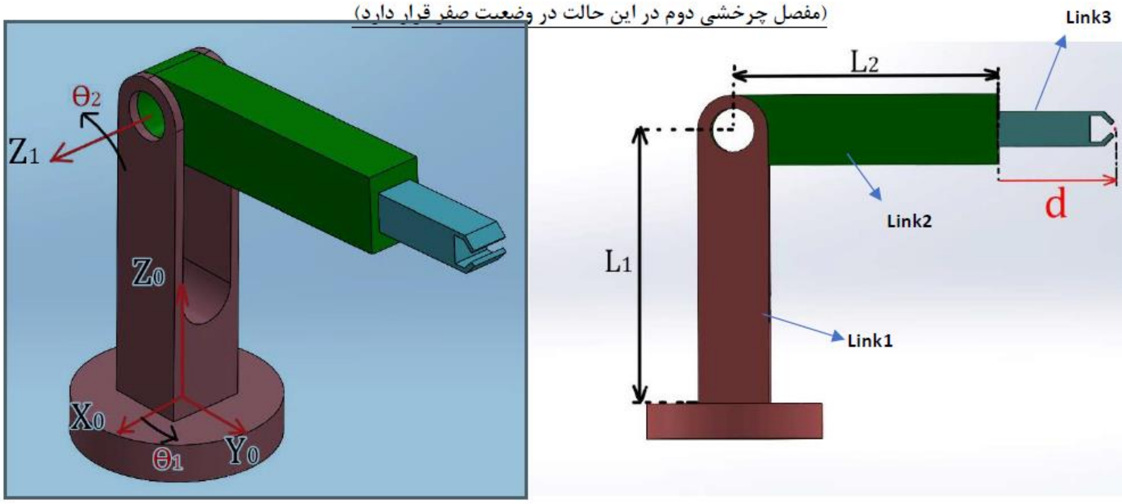
می دانیم برای محاسبه نقاط تکین داریم jv=0 ، *بنابراین* نقاط تکین ربات می شود:

theta2=kpi+pi/2 یا D3=-30cm

۳. فرض کنید عملگر نهایی **ربات کروی (تمرین ۳)** باید قادر باشد در فضای کاری خود، بردار سرعت $\vec{v} = (150,100,50)mm/s$ را در تمام نقاط ایجاد نماید. فضای کاری ربات را اینگونه در نظر بگیرید که مفصل اول (θ_1) از ۰ تا ۸۰ **درجه**، مفصل دوم از -۴۵ تا ۴۵ **درجه** و مفصل سوم از d=100mm تا d=200mm حرکت می‌کند. حداکثر سرعت مورد نیاز برای هریک از موتورهای سه گانه ربات را بدست آورید. گزارشی خلاصه از روند حل به همراه برنامه متلب و نتایج نهایی را ارائه نمایید.

(L_1 = 31 cm , L_2 = 30 cm)

(مفصل چرخشی دوم در این حالت در وضعیت صفر قرار دارد)



از رابطه روبرو می دانیم:

$$\vec{v}_e = J_v \dot{\Theta}$$

بردار ژاکوبین سرعت ضربدر ماتریس سرعت مفصل ها می شود سرعت عملگر نهایی.

بنابراین اگراینورس ژاکوبین سرعت انتقالی را درسرعت عملگر نهایی ضرب کنیم بردار سرعت مفاصل حساب می شود.

ژاکوبین را که در تمرین 1 محاسبه کرده بودیم به راحتی وارون کرده و در بردار سرعت داده شده ضرب می کنیم. سرعت مفاصل حساب می شود.

این معادله به صورت سمبولیک است. برای اینکه ببینیم بیشترین سرعت مفاصل چه قدر می شود باید تقریبا در همه فضای کاری داده شده متغیرهای مفاصل را جای گذاری کنیم و سرعت ها را در یک ماتریس ذخیره کنیم و برای هر مفصل، سرعت ماکسیمم را حساب کینم.

کد مربوط به سوال:

```
ve=[15;10;5];
speed=[];
speed_=inv(jv)*ve;
speed_=simplify(speed_);


for i=(0:4:80)*pi/180
    for j=(-45:4:45)*pi/180
        for k=10:0.5:20
            R=subs(speed_,[theta1,theta2,d3],[i,j,k]);
            speed=[speed,R];

        end
    end
end

v1=vpa(speed(1,:),3);
v2=vpa(speed(2,:),3);
v3=vpa(speed(3,:),3);

max_v1=max(abs(v1));
max_v2=max(abs(v2));
max_v3=max(abs(v3));

disp('maxv1 maxv2 maxv3')
disp([max_v1 max_v2 max_v3])
```

maxv1 maxv2 maxv3
[0.46087927826738450676202774047852, 0.40693843379267491400241851806641, 18.699995627626776695251464843 75]

max theta1 dot=0.460 rad/s

max theta2 dot=0.407 rad/s

max d3 dot=18.69 cm/s