

SORU :

1. Bölüm

Hafta 13 den IMU ve VICON veri setini indiriniz.

http://yunus.hacettepe.edu.tr/~saritan/bca607/hafta_13.zip

- Zip dosya içerisinde bulunan Etiketlenmiş Vicon (*vicon_all_labelled.txt*) veri setini kullanarak IMU (Inertial **M**esurament **U**nit) sensörünün bağlandığı kalibrasyon cubuğunun yörüngesini çizin.
- M1, M3 ve M5 işaretlerini (marker) kullanarak bir katı cisim oluşturunuz ve hareketin animasyonunu yapınız. (*viconAnim.avi*)
- Bu katı cismin 3 boyutlu yönelimi tanımlamak için Euler Açılarının özelleşmiş bir versiyonu olan “Roll, Pitch, Yaw” açılarını hesaplayınız.

CEVAP :

Text dosyasından Marker değerleri ve EulerXYZ değerlerinin okunduğu kod kısmı.

EulerXYZ değerleri dosyadan eul_f 708x3 matris değişkenine yazılıyor.

Marker değerleri dosyadan data_f 5x3x708 matris değişkenine yazılıyor.

```
clc,clear
```

```
değişkenler tanımlamaları yapılmıştır.
```

```
counter = 1;
```

```
eul_c = 1;
```

```
fid = fopen('C:\vicon_all_labelled.txt','r');
```

```
Tüm dosya satır satır okunmuştur.
```

```
Her satır line değişkeninde tutulmuştur.
```

```
for c=1:41771
```

```
line=fgets(fid) ;
```

```
Satırdaki veriler textscan ile okunmuştur. Eğer o satırda EulerXYZ değerleri varsa E değişkenine x,y,z şeklinde yazılmıştır.
```

```
E = textscan(line, 'Local Rotation EulerXYZ: %f%f%f 0' ...  
, 'Delimiter', {' ',''}, 'Whitespace', '() ');
```

```
Textscan komutunun döndürdüğü değer boş olup olmadığı kontrol edilmiştir.
```

```
Eğer boş değilse x,y,z sırasında eul_f değişkenine yazılmıştır.
```

eul_c değeri sayarak satır numarasını belirtilmiştir.

```
if(E{1,1} ~= 0)
eul_x = E{1,1};
eul_y = E{1,2};
eul_z = E{1,3};
eul_f(eul_c,1) = eul_x;
eul_f(eul_c,2) = eul_y;
eul_f(eul_c,3) = eul_z;
eul_c = eul_c + 1;
end
```

Aynı işlem Markerları okumak için yapılıyor ve Marker değerleri data_f 5x3x708 matris değişkenine yazılmıştır.

```
M = textscan(line, 'Marker #f: M_f %f%f%f 0' ...
, 'Delimiter', {' ',''}, 'Whitespace', '() ');
if M{1,1} ~= 0
data(M{1,1},1) = M{1,3};
data(M{1,1},2) = M{1,4};
data(M{1,1},3) = M{1,5};
if M{1,1} == 5
data_f(:, :, counter) = data(:, :);
counter = counter + 1;
end
end
end c sayacı olan for döngüsü sonu
fclose(fid); dosya kapatılıyor.
```

eul_f				
708x3 double				
	1	2	3	
1	1.5759	-0.0071	-1.2342	
2	1.5759	-0.0069	-1.2343	
3	1.5760	-0.0069	-1.2342	
4	1.5760	-0.0072	-1.2343	
5	1.5759	-0.0069	-1.2341	
6	1.5760	-0.0069	-1.2342	
7	1.5760	-0.0071	-1.2341	
8	1.5760	-0.0073	-1.2339	
9	1.5761	-0.0071	-1.2342	
10	1.5760	-0.0071	-1.2341	
11	1.5759	-0.0071	-1.2342	
12	1.5761	-0.0068	-1.2341	
13	1.5767	-0.0058	-1.2346	
14	1.5759	-0.0071	-1.2341	
15	1.5764	-0.0069	-1.2345	
16	1.5763	-0.0069	-1.2344	

data_f			
5x3x708 double			
val(:, :, 1) =			
-75.287	49.015	-925.73	
7.7601	49.542	-1163.1	
6.4489	49.189	-1044.2	
4.6257	48.31	-925.11	
164.48	47.549	-923.47	
val(:, :, 2) =			
-75.291	48.989	-925.71	
7.7542	49.547	-1163.1	
6.4429	49.197	-1044.2	
4.5988	48.316	-925.08	
164.46	47.57	-923.54	

a.Kalibrasyon çubuğu ve yörüngesinin çizilmesi

data_f değişkeninden ilk marker yer işaretleri çubuğun modellenmesi için gerekli sırayla değişkenlere yazılmıştır.

```
x = [data_f(1,1,1) data_f(4,1,1) data_f(3,1,1) data_f(2,1,1)
data_f(4,1,1) data_f(5,1,1)];
```

```
y = [data_f(1,3,1) data_f(4,3,1) data_f(3,3,1) data_f(2,3,1)
data_f(4,3,1) data_f(5,3,1)];
```

```
z = [data_f(1,2,1) data_f(4,2,1) data_f(3,2,1) data_f(2,2,1)
data_f(4,2,1) data_f(5,2,1)];
```

x y z değişkenlerindeki sıraya göre plot edilmiş ve aralarına çizgi çizilmiştir.

```
figure(1)
plot3(x,y,z,'-
o','Color','b','MarkerSize',10,'MarkerFaceColor','red');
axis equal
hold on
```

plot edilen modelin üzerine x y z eksenleri çizilmiştir.

pt değişkeninde başlangıç noktası tutulmaktadır.

```
pt = [data_f(4,1,1) data_f(4,3,1) data_f(4,2,1)];
```

dir değişkeninde sırasıyla x y z de 50 birim olacağı belirtilmiştir. quiver komutu ile çizilmiştir.

%----- x eksenini çiziliyor.

```
dir = [50 0 0 1];
q = quiver3(pt(1),pt(2),pt(3), dir(1),dir(2),dir(3),1.5);
q.LineWidth = 3;
q.MaxHeadSize = 3;
q.ShowArrowHead = 'on';
```

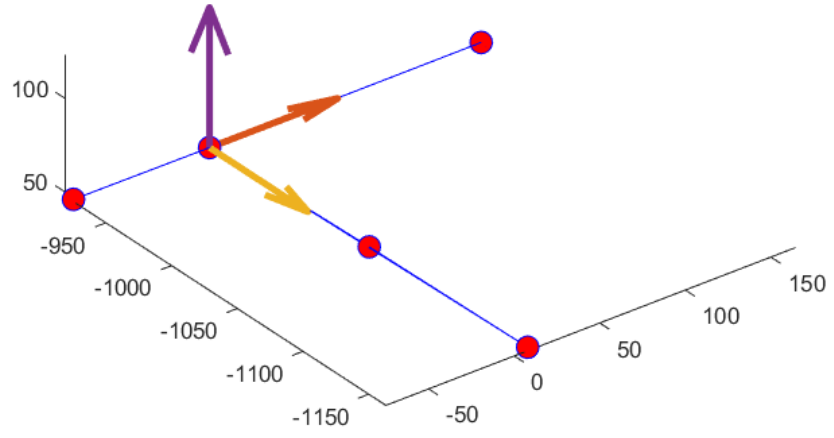
%----- y eksenini çiziliyor.

```
dir = [0 -50 0 1];
q = quiver3(pt(1),pt(2),pt(3), dir(1),dir(2),dir(3),1.5);
q.LineWidth = 3;
q.MaxHeadSize = 3;
q.ShowArrowHead = 'on';
```

%----- z eksenini çizilmiştir.

```
dir = [0 0 50 1];
q = quiver3(pt(1),pt(2),pt(3), dir(1),dir(2),dir(3),1.5);
q.LineWidth = 3;
q.MaxHeadSize = 3;
q.ShowArrowHead = 'on';
```

Ekran ıktısı



b. Katı cisim oluşturma ve animasyonu

Video nesnesi oluşturuldu.

```
aviobj = VideoWriter('viconAnim.avi', 'Uncompressed AVI');  
aviobj.FrameRate = 25;  
open(aviobj);
```

Eksenler ayarlandı.

```
xlim([-100 200])  
ylim([-2500 1000])  
zlim([0 1400])  
axis([-100 200 -2500 1000 0 1400])  
[mode,visibility,direction] = axis('state')  
grid on  
pbaspect ([1 7 5])  
hold on
```

fill3 komutu ile data_f değişkeninden M1,M3,M5 yer işaretleriyle ekrana yazılıyor.

```
for i=1:708  
X = [data_f(1,1,i) data_f(2,1,i) data_f(5,1,i)]  
Y = [data_f(1,3,i) data_f(2,3,i) data_f(5,3,i)];  
Z = [data_f(1,2,i) data_f(2,2,i) data_f(5,2,i)];  
fill3(X,Y,Z,1);
```

Ekrandaki çizim video nesnesine frame olarak ekleniyor.

```
frame = getframe(gcf);  
writeVideo(aviobj, frame);
```

cla Ekrandaki çizimin içi siliniyor.

```
end  
close(aviobj);
```

c. Roll, Pitch, Yaw açılarının hesaplanması

Text dosyasından okunan açıların `eul2rotm` komutu kullanılarak rotasyon matrisine dönüştürülmüştür.

<http://planning.cs.uiuc.edu/node103.html> internet sayfasında yardım alınarak roll, yaw, pitch değerleri hesaplanmıştır.

Determining yaw, pitch, and roll from a rotation matrix

It is often convenient to determine the α , β , and γ parameters directly from a given rotation matrix. Suppose an arbitrary rotation matrix

$$\begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$$

is given. By setting each entry equal to its corresponding entry in (3.42), equations are obtained that must be solved for α , β , and γ . Note that $r_{21}/r_{11} = \tan \alpha$ and $r_{32}/r_{33} = \tan \gamma$. Also, $r_{31} = -\sin \beta$ and $\sqrt{r_{32}^2 + r_{33}^2} = \cos \beta$. Solving for each angle yields

$$\alpha = \tan^{-1}(r_{21}/r_{11}),$$

$$\beta = \tan^{-1}\left(-r_{31}/\sqrt{r_{32}^2 + r_{33}^2}\right),$$

$$\gamma = \tan^{-1}(r_{32}/r_{33}).$$

Text dosyasından okunmuş ve `eul_f` değişkeninde tutulan açı değerleri rotasyon matrisinde dönüştürülmüştür. Rotasyon matrisinden ters tan uygulanarak yukarıda belirtilen formüller yardımıyla değerler hesaplanmış ve filtre işlemine tabii tutulmuştur.

```
for i=1:708
eul = [eul_f(i,1),eul_f(i,2),eul_f(i,3)];
rotm = eul2rotm(eul,'XYZ');
rotmZYX = eul2rotm(eul,'ZYX');
rotmYZZ = eul2rotm(eul,'YZZ');
roll(i,1) = atan2d(rotmZYX(2,1),rotmZYX(1,1));
pitch(i,1) = 1.5*atan((-
1*rotmZYX(3,1))/sqrt((rotmZYX(3,2)^2 + (rotmZYX(3,3)^2)));
yaw(i,1) = -atand(rotmYZZ(3,2)/rotmYZZ(3,3));
end
figure(2)
subplot(3,1,1);
[b2 a2] = butter(1,25/60,'low');
```

```
roll = filtfilt(b2,a2,roll);  
plot(roll)  
ylabel('roll')  
  
subplot(3,1,2);  
plot(pitch)  
ylabel('pitch')  
  
subplot(3,1,3);  
[b2 a2] = butter(1,5/30,'high');  
yaw = filtfilt(b2,a2,yaw);  
plot(yaw)  
ylabel('yaw')
```

Ekran Çıktısı

