

Çağın AĞIRDEMİR

SORULAR

5. Elde ettiğiniz yansıtıcı işaretlerin görüntü koordinat sisteminden dünya koordinat sistemine geciriniz (**2 Boyutlu Kalibrasyon**). Kısaca ünitemiz **piksel [px]** den **metre [m]** ye çevirelecek. Hatırlatma: **5 Hafta** ders notlarına bakınız. Kalibrasyon düzlemi değerlerini **BAG_Kalibrasyon.pdf** dosyasında bulabilirsiniz. (%15)
Kalibrasyon Görüntüsü: *fettah_kalibrasyon.jpg*
6. **Ayak Ucu, Topuk, Ayak Bileği, Diz ve Kalçanın** yerleştirilen yansıtıcı işaretlerden elde edilen konum değerlerinin hız ve ivme değerlerini hesaplayınız, gürültüyü filtre kullanarak ortadan kaldırınız ve zamana göre grafiğini çiziniz. (%15)
7. Dizin açısal hız değişimin hesaplayınız ve zamana göre grafiğini çiziniz. (%10)

CEVAP 5. Dünya koordinat sistemine geiř

```
S = [ 2.5 30;  
      2.5 60;  
      2.5 130;  
      2.5 190;  
      108 10;  
      108 80;  
      108 120;  
      108 180];  
load calib_im.txt;  
x = calculate_conformal(calib_im, S, 1);
```

BAG_kalibrasyon.pdf dosyasından alınan dikey
alışma koordinatları.



calib_im.txt - Notepad

File Edit Format View Help

```
405.97,197.05  
405.01,295.11  
403.03,525.44  
400.00,722.62  
753.73,130.39  
752.95,359.78  
751.95,492.35  
749.16,691.07
```

Kalibrasyon.jpg dosyasından alınan koordinatlar
ile oluşturulmuş calib_im.txt dosyası

Variables - centroids_data			
H centroids_data			
8x2 double			
	1	2	3
1	405.9429	197.0429	
2	405.1096	295.2329	
3	402.9733	525.4400	
4	400.0882	722.8382	
5	753.8286	130.3857	
6	752.9412	359.9706	
7	751.9394	492.1515	
8	749.1231	691.1231	
9			
10			
11			

Variables - H			
H centroids_data			
8x2 double			
	1	2	3
1	0.0277	0.3061	
2	0.0268	0.6040	
3	0.0241	1.3022	
4	0.0185	1.9010	
5	1.0818	0.0983	
6	1.0829	0.7947	
7	1.0820	1.1956	
8	1.0767	1.7992	
9			
10			

Koordinat sırası BAG_kalibrasyon.pdf dosyasındaki sıralama ile aynıdır.

Bu koordinatlar sınav sorularında bulunan koordinatlarla karşılaştırılmıştır.

```

RGB = imread(sprintf('C:\\fettah_sut2_C001H001S00010000%d.jpg',i));
mask=255*zeros(size(RGB));
mask=insertShape(mask,'FilledRectangle',[0 0 1024 250],'Color','white');
mask=uint8(mask);
RGB(mask ~= 0) = mask(mask ~= 0);

```

Bu soruda imcrop yerine insertShape fonksiyonu kullanılarak diğer yansıtıcı nesneler çıkartılmıştır.

```

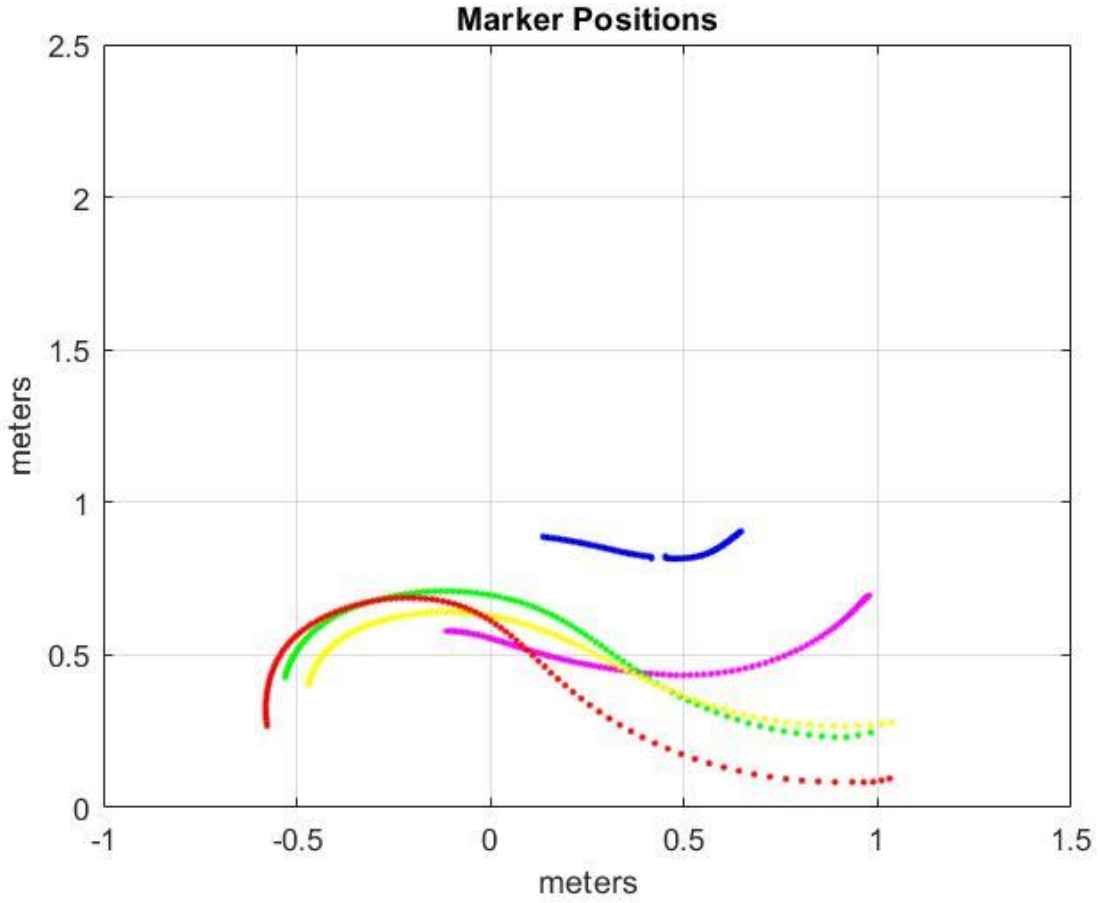
for i=1:105
centroids_data(:,2,i) = 1024 - centroids_data(:,2,i);
hip(i,:) = centroids_data(1,:,i);
knee(i,:) = centroids_data(2,:,i);
heel(i,:) = centroids_data(3,:,i);
ankle(i,:) = centroids_data(4,:,i);
toe(i,:) = centroids_data(5,:,i);
end

```

centroids_data değişkenindeki veriler marker pozisyon adları ile isimlendirilmiş değişkenlere ayrılmıştır.

y koordinatları frame boyutu olan 1024 ten çıkartılarak yukarıdan aşağı doğru olan koordinat hesaplanmıştır.

Bu değişkenler calculate_reconformal fonksiyonu ile cm olarak tekrar hesaplanmıştır.



Bu deęişkenlerin (hip,heel,ankle,knee,toe) plot edilmiş hali.

CEVAP 6. Hız ve ivme deęerlerinin grafikleri

Örnek olarak diz hesaplama kodları anlatıldıktan sonra direct marker hız, ivme grafikleri eklenmiştir.

Hız hesaplama

$$V_i = \frac{S_{i+1} - S_{i-1}}{2\Delta t}$$

Video 500 fps olduğundan dolayı $\Delta t = \frac{1}{500}$ alınmıştır.

İvme hesaplama

$$a_i = \frac{V_{i+1} - V_{i-1}}{2\Delta t}$$

Örnek Filtre komutu

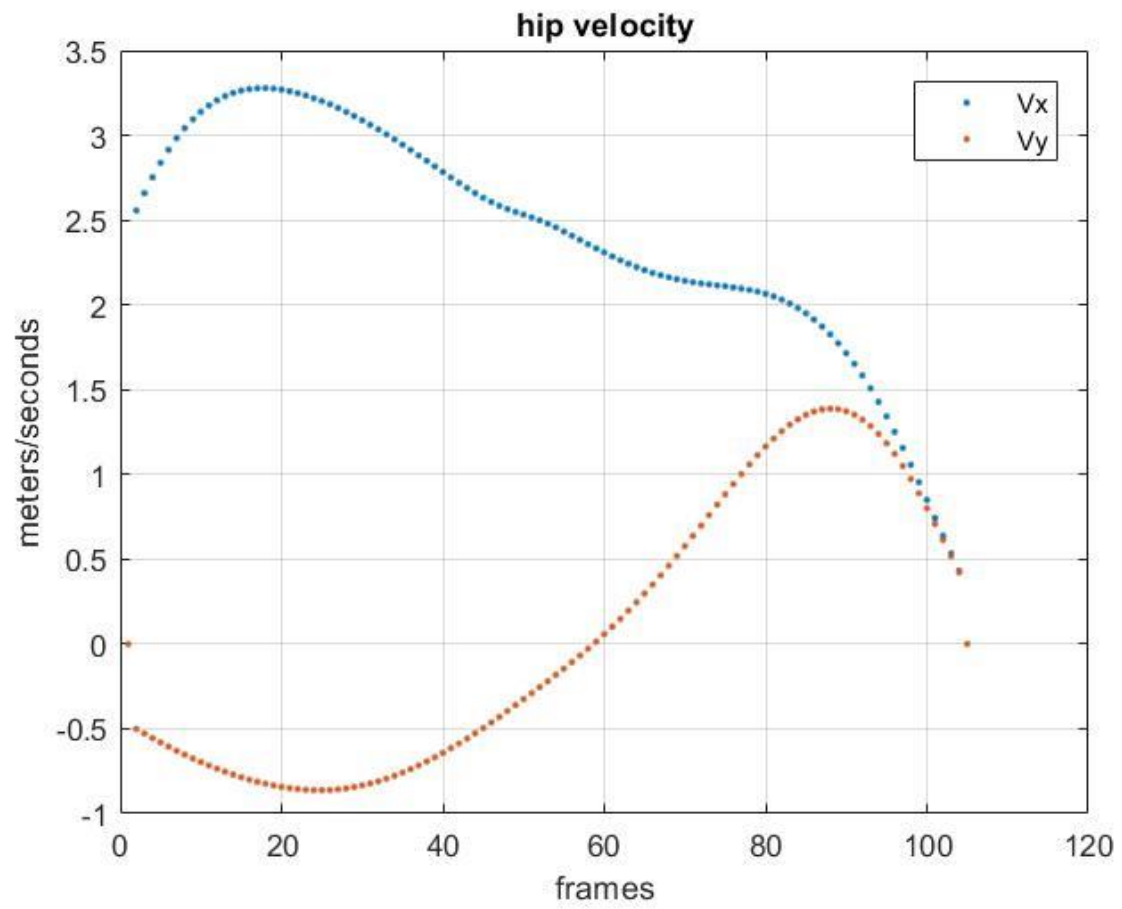
```
[b,a]= butter(2,10/250,'low');  
%filtrelenecek marker degiskenini giriniz  
filtering_marker = toe_cm;  
filterData(:,1) = filtfilt(b,a,(filtering_marker(:,1)));  
filterData(:,2) = filtfilt(b,a,(filtering_marker(:,2)));
```

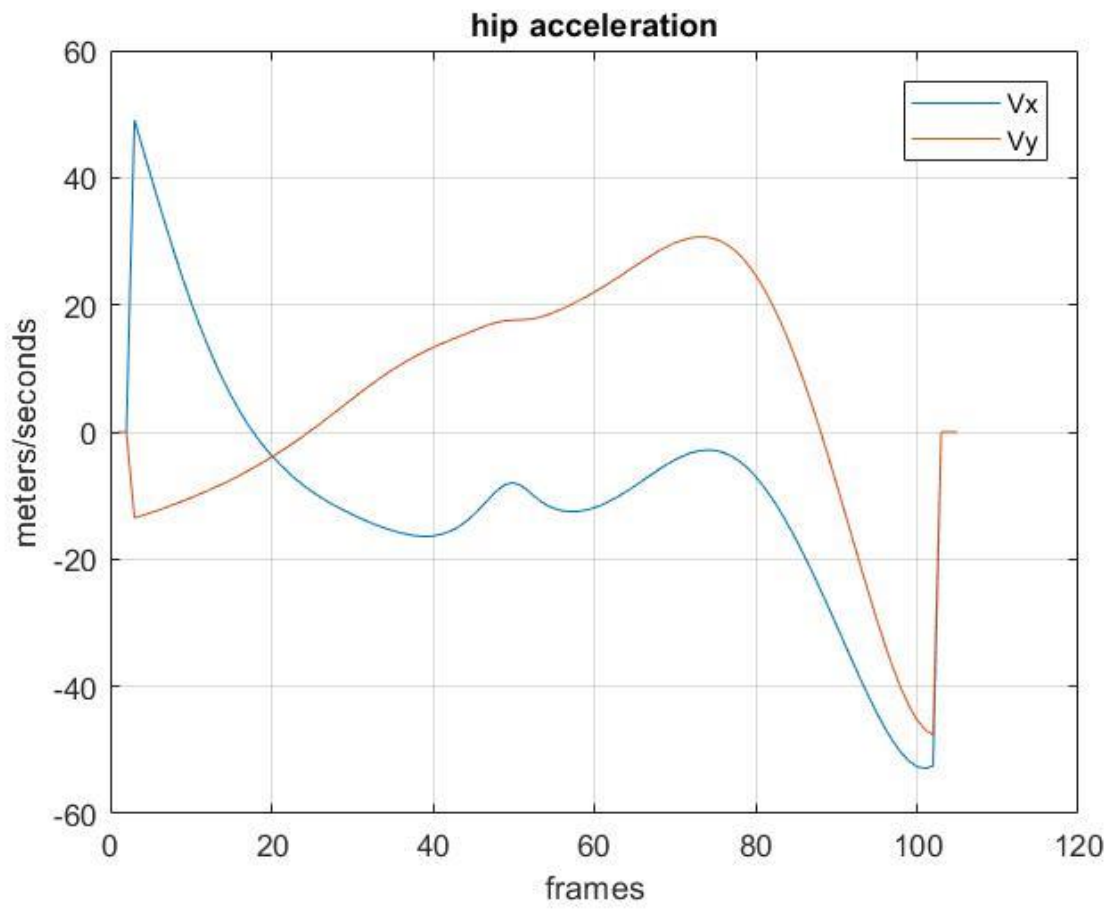
Örnek Hız hesaplama

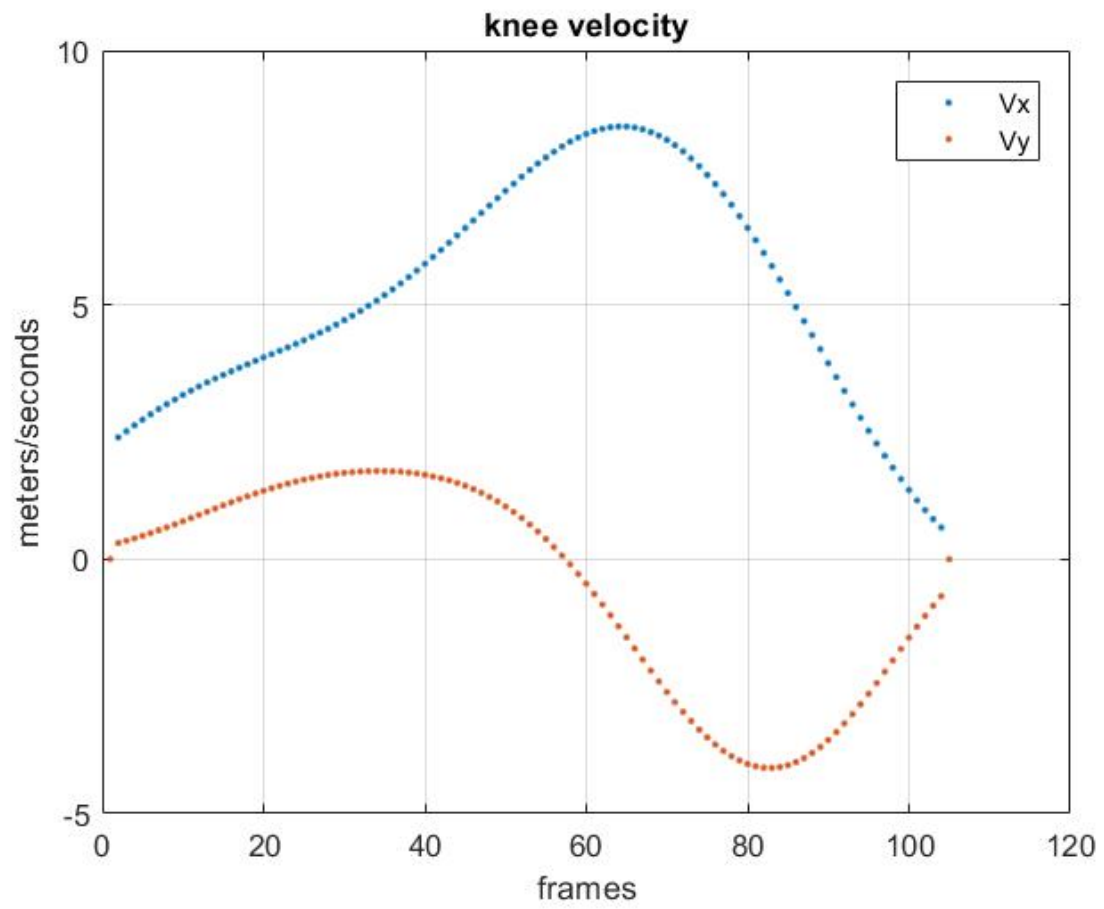
```
for p=1:length(filterData)  
if p <= 103  
filterData(p+1,3)=(filterData(p+2,1)-filterData(p,1))/(2/500);  
filterData(p+1,4)=(filterData(p+2,2)-filterData(p,2))/(2/500);  
end  
end  
figure(1), plot(filterData(:,3),'-');axis([0 120 -1 3.5]);grid on;  
xlabel('frames');  
ylabel('meters/seconds');  
hold on  
figure(1), plot(filterData(:,4),'-');axis([0 120 -1 3.5]);grid on;  
legend('Vx','Vy');
```

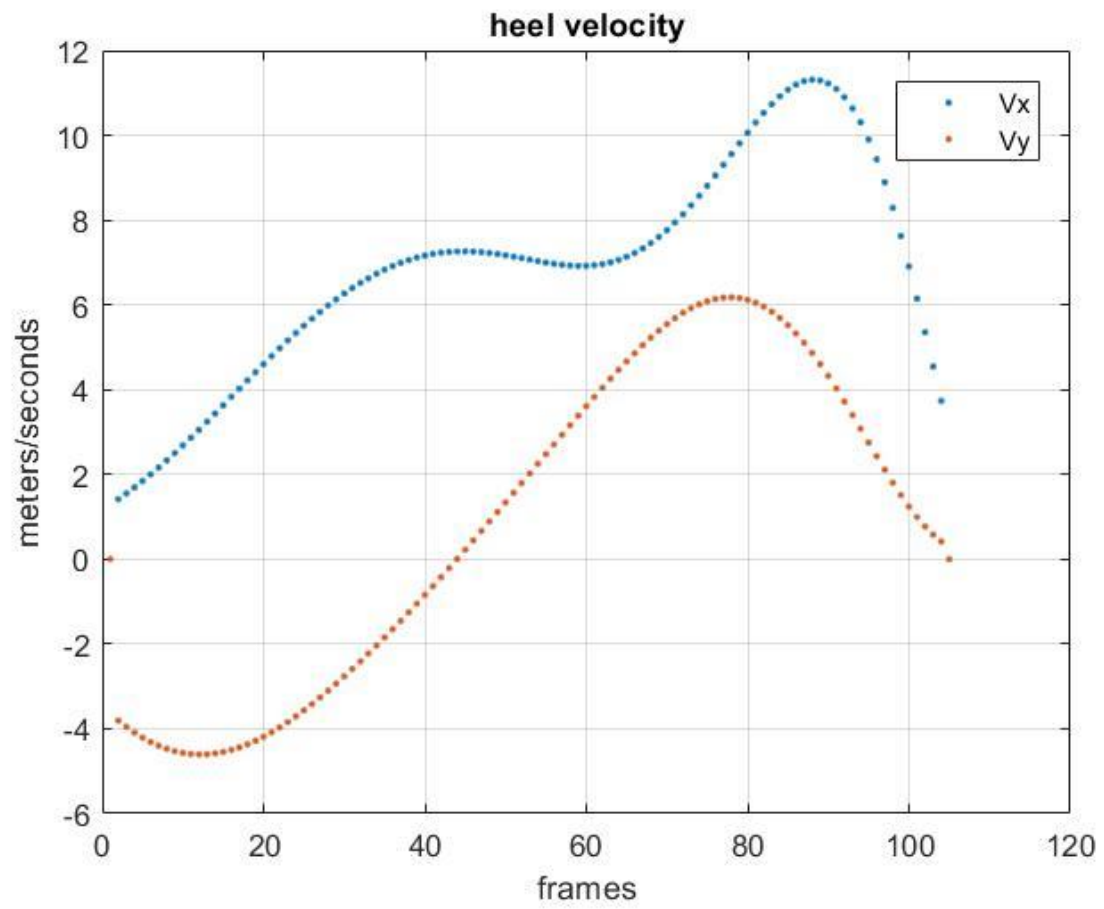
Örnek İvme hesaplama

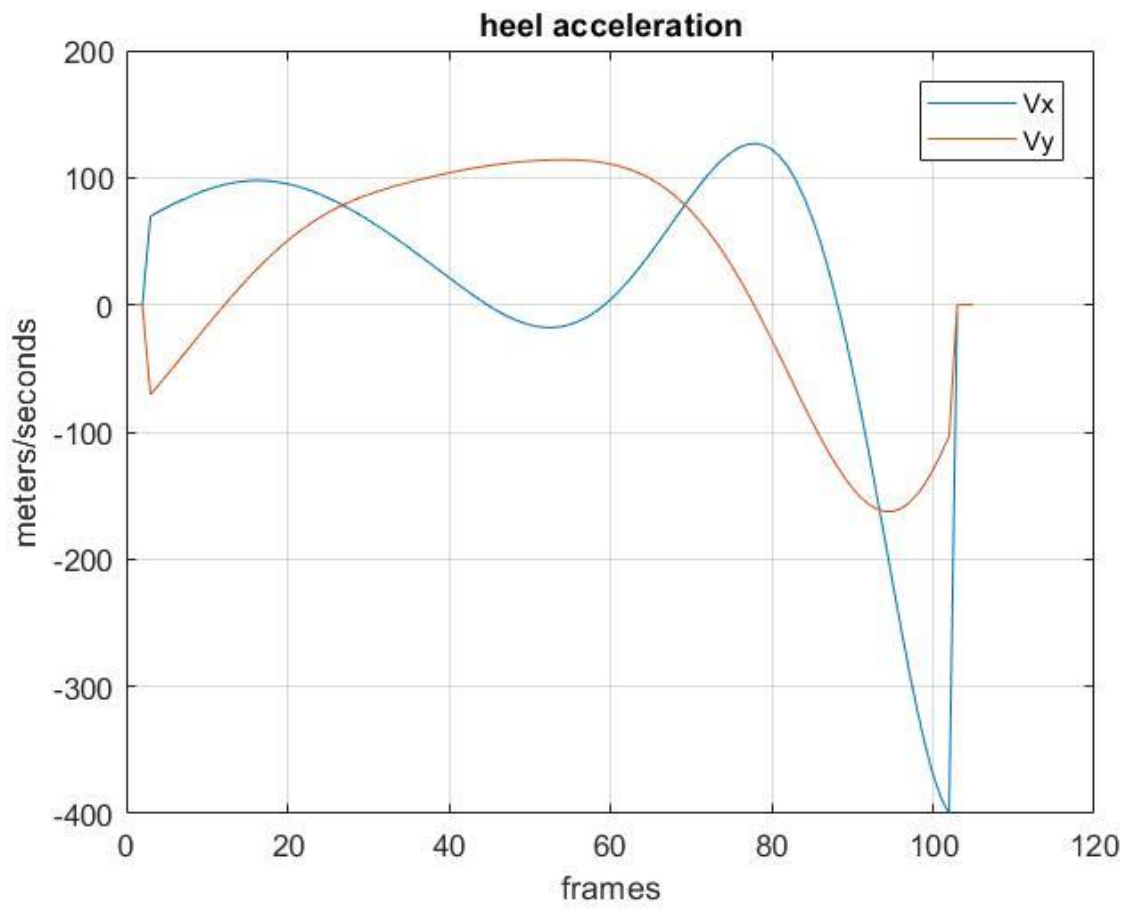
```
for p=3:length(filterData)  
if p <= 102  
filterData(p,5)=(filterData(p+1,3)-filterData(p-1,3))/(2/500);  
filterData(p,6)=(filterData(p+1,4)-filterData(p-1,4))/(2/500);  
end  
end  
figure(2), plot(filterData(:,5),'-');axis([0 100 -60 60]);grid on;  
xlabel('frame');  
ylabel('meters/seconds');  
hold on  
figure(2), plot(filterData(:,6),'-');axis([0 100 -60 60]);grid on;  
legend('Vx','Vy');
```

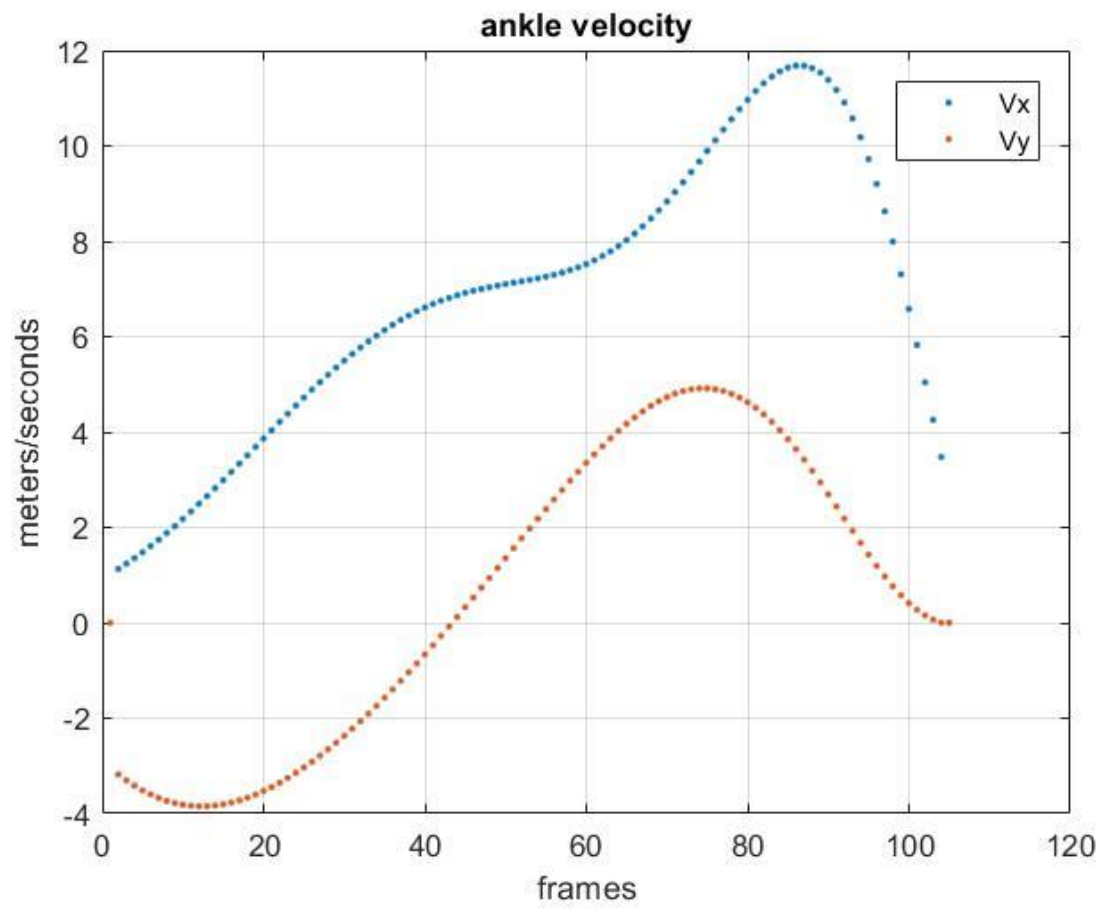


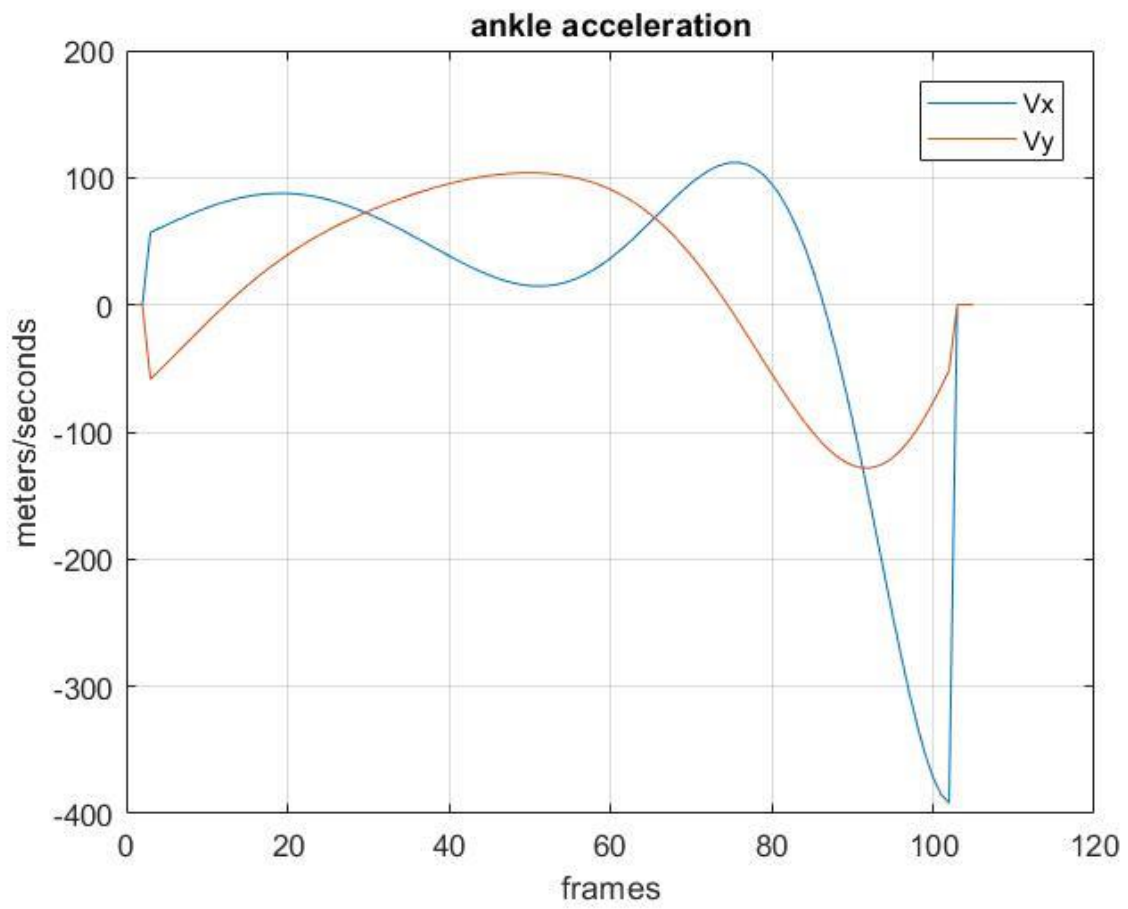


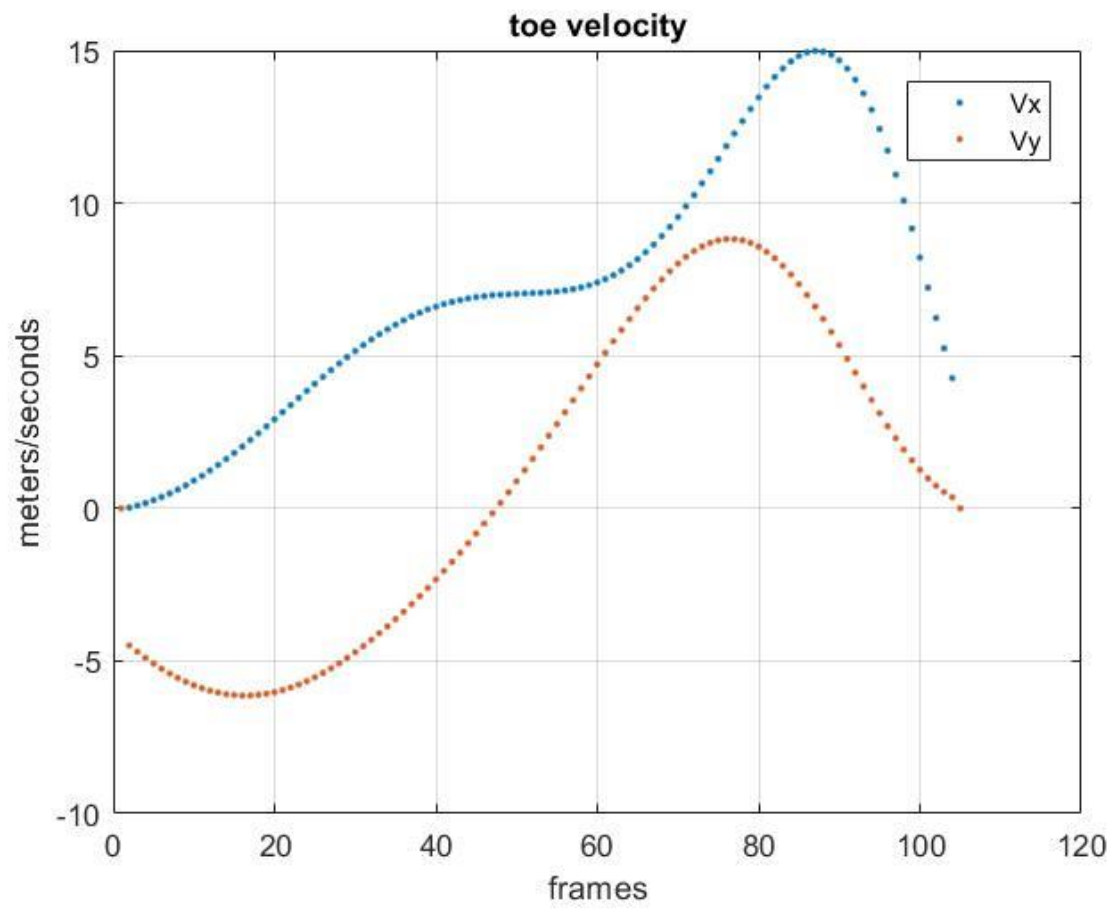


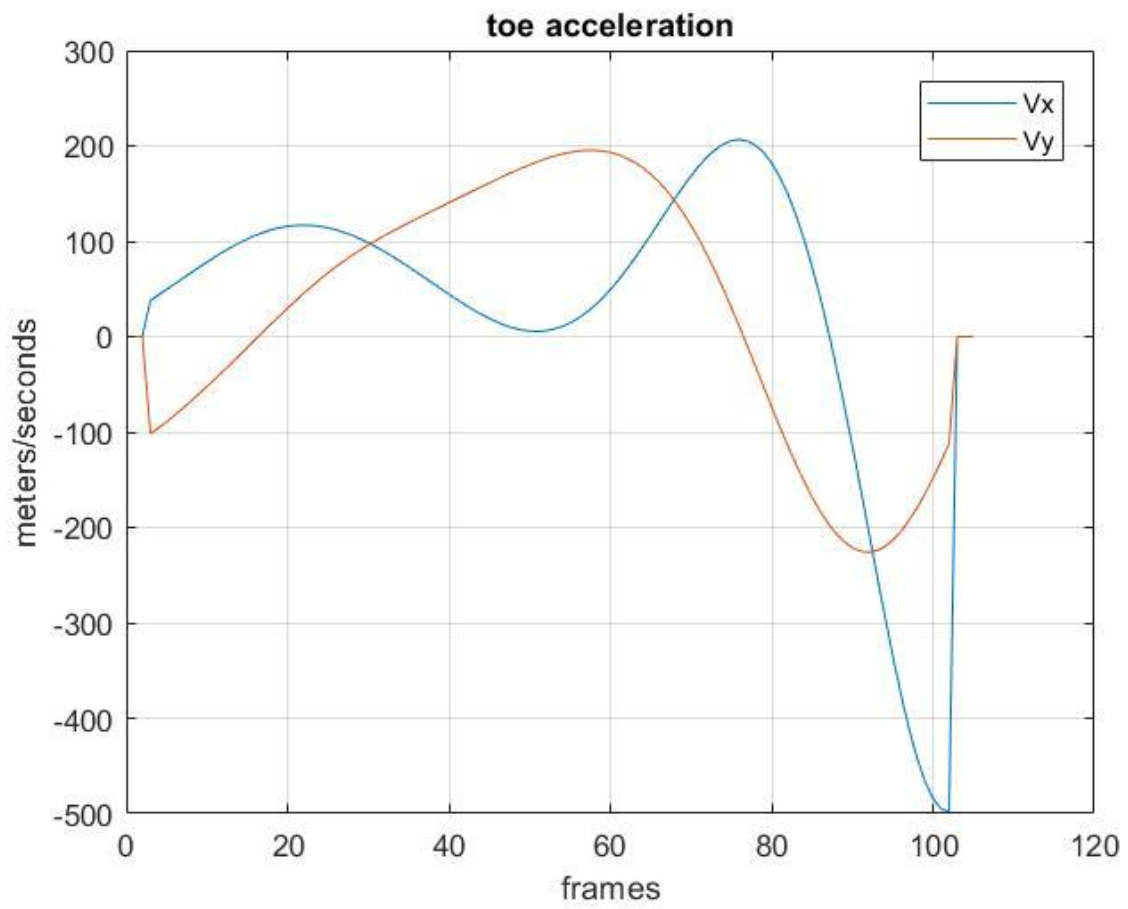


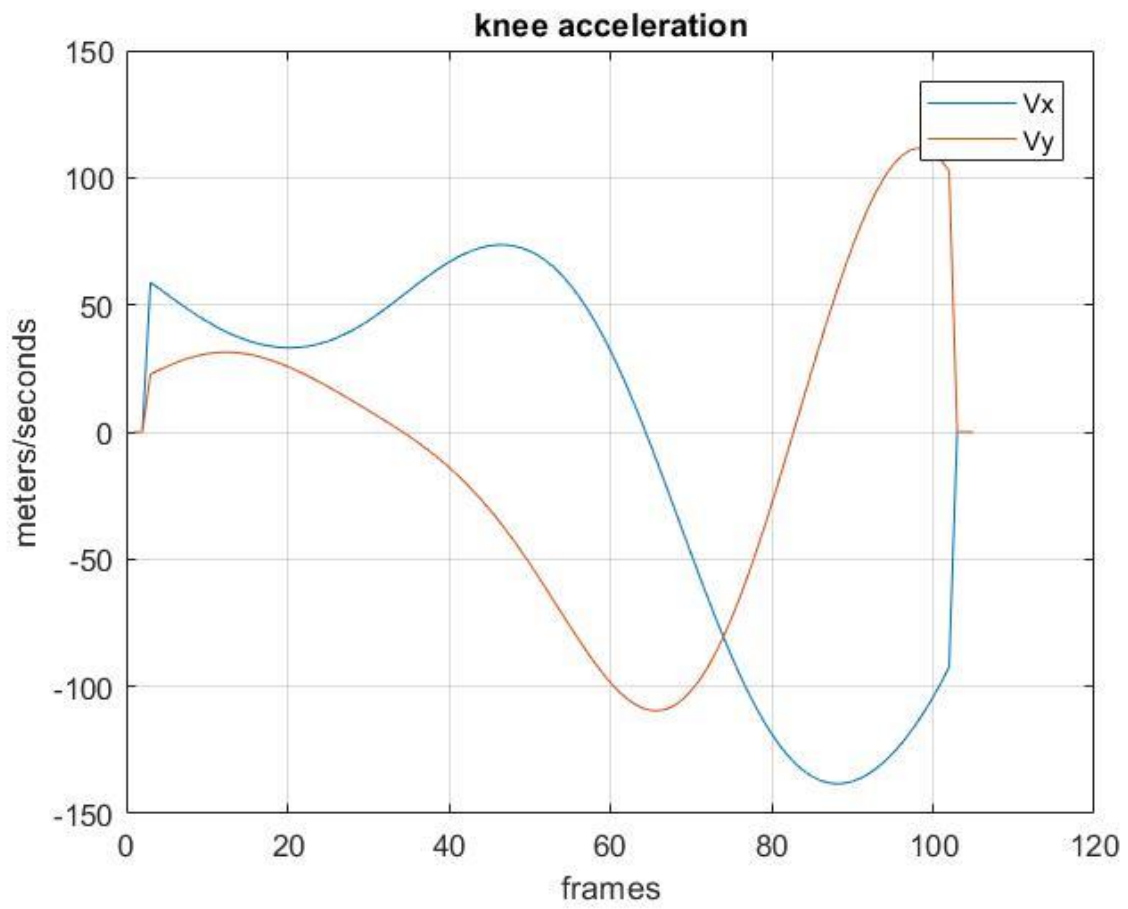








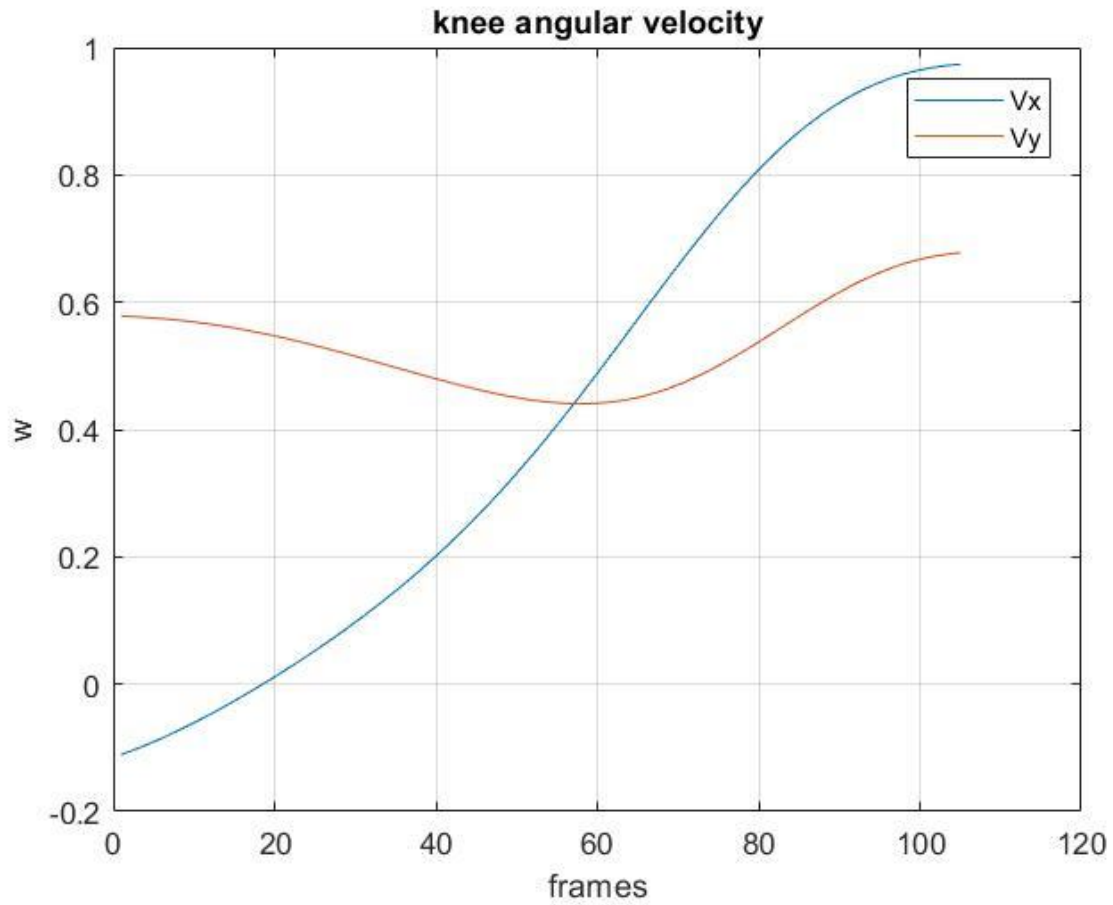




CEVAP 7. Açısal hız hesaplama

```
[h ,w] = freqz(b,a,filterData(:,1));  
figure(3),plot(w);grid on;  
hold on  
title('knee angular velocity');  
xlabel('frames');  
ylabel('w');  
[h ,w] = freqz(b,a,filterData(:,2));  
figure(3),plot(w);grid on;  
legend('Vx','Vy');
```

fps bilindiği için saniye frame dönüşümü yapılabilir.



Ek

arasinav_5_7.m