

BCA 607 Hareket Analizi Sistemleri

ARA SINAV

Tayfun GÜRLERİK
N19139647
02 Aralık 2019

İçindekiler

1. Yansıtıcı İşaretlerin Geometrik Merkezlerinin (Centroid) koordinatlarının hesaplanması	2
2. Hesaplanan Geometrik Merkezlerin Üzerlerine Farklı Renklerde Yuvarlak İşaretler Yerleştirilmesi, Koordinatlarının Yazılması.....	6
3. Markerların arasına beyaz çizgi çizilmesi ve Videonun oluşturulması	6
4. Kalibrasyon işlemi ve Noktaların Dünya koordinat sistemine aktarılması	8
5. Grafiklerin Elde Edilmesi	10
6. Dizin Açısıl Hız Değişiminin Hesaplanması ve Grafiğinin Elde Edilmesi	19

Resim 1 RGB resim.....	2
Resim 2 Gri tonlamalı resim	2
Resim 3 background resmi	3
Resim 4 Gri tonlamalı resimden background çıkartılınca elde edilen sonuç	3
Resim 5 Kontrast eklenmiş resim	4
Resim 6 Markerların pozisyon bilgileri ve markerları birleştiren çizgiler	7
Resim 7 Kalibrasyon resminde markerların işaretlenmesi.....	8

Grafik 1 Marker Pozisyonları	13
Grafik 2 Ayak Ucu Hız Değişimi	13
Grafik 3 Topuk Hız Değişimi.....	14
Grafik 4 Bilek Hız Değişimi	15
Grafik 5 Diz Hız Değişimi	15
Grafik 6 Kalca Hız Değişimi	16
Grafik 7 Ayak Ucu İvme Değişimi.....	16
Grafik 8 Topuk İvme Değişimi	17
Grafik 9 Bilek İvme Değişimi	17
Grafik 10 Diz İvme Değişimi	18
Grafik 11 Kalca İvme Değişimi	18
Grafik 12 Dizin Açısıl Hızının Değişimi	19

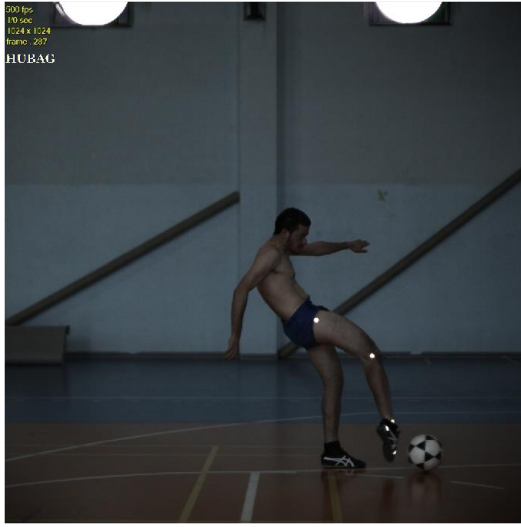
1. Yansıtıcı İşaretlerin Geometrik Merkezlerinin (Centroid) koordinatlarının hesaplanması

İlk olarak dosya adı değişkeni, for döngüsü içerisinde sıralı olacak şekilde oluşturulabilmesi amacıyla formatlı bir string oluşturulmuştur.

```
dosya_on_ad='fettah_sut2_C001H001S0001';  
dosya_uzanti='.jpg';  
dosya_adi=strcat(dosya_on_ad,sprintf( '%06d', n ) ,dosya_uzanti);
```

Burada n o anda işlenen karenin sayısıdır.

Daha sonra imread komutu ile dosya renkli olarak okunup RGB değişkenine atanmıştır.



Resim 1 RGB resim

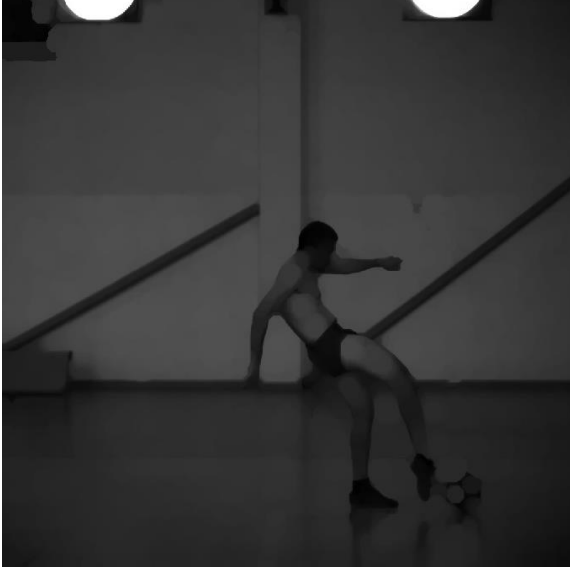
rgb2gray komutu ile resim gri ölçekli olarak elde edilmiş ve I değişkenine atanmıştır.

```
RGB=imread(dosya_adi);  
I=rgb2gray(RGB);
```

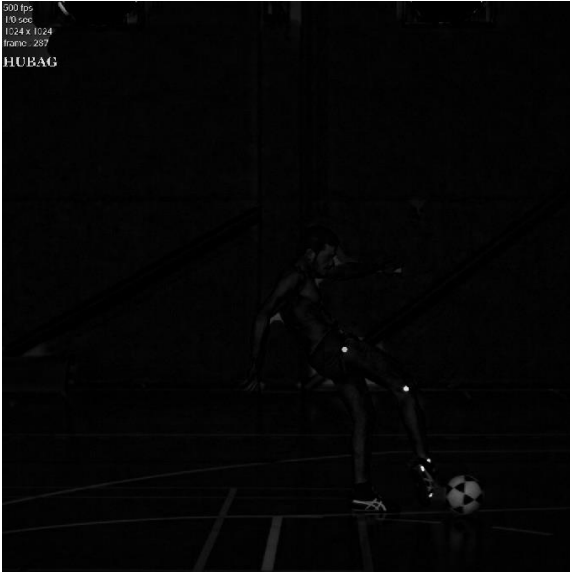


Resim 2 Gri tonlamalı resim

Aydınlatmadan kaynaklanan parlamaları resimden çıkarmak amacıyla ikl önce strel('disk',15) komutu ile 15 pixel yarıçapında disk yapısal elemanı oluşturulmuştur. Daha sonra gri ölçekli resim bu yapısal elemanla açılarak arkaplan ışık görüntüsü elde edilmiştir(background). Gri ölçekli resimden background çıkartılarak daha az ışık olan ikinci bir gri resim elde edilmiştir(I2).



Resim 3 background resmi



Resim 4 Gri tonlamalı resimden background çıkartılınca elde edilen sonuç



Resim 5 Kontrast eklenmiş resim

I₂ görüntüsü üzerinde imadjust komutu ile kontrast oluşturularak I₃ görüntüsü elde edilmiştir. Binary görüntü oluşturmak için I₃ görüntüsü kullanılmıştır.

```
se=strel('disk',15);  
background=imopen(I,se);  
I2=I-background;  
I3=imadjust(I2,[0.3 0.7],[ ]);
```

I₃ resminin threshold değeri kullanılarak resim binary formata dönüştürülmüştür. Y<=135 değeri için maskeleme yapılarak sol üstte bulunan beyaz yazıların siyah beyaz resmin içine girmesi engellenmiştir.

```
T=graythresh(I3);  
BW=imbinarize(I3,T);  
%maskeleme islemi icin Y si 135 den kucuk olan tum pixelleri 0 yaptim.  
BW(:,1:135)=0;  
BW=medfilt2(BW,[3 3]);  
BW = bwareaopen(BW,15);
```

Bwboundaries komutu ile markerların labellama işlemi yapılmış, elde edilen label bilgileri ile regionprops komutuyla centroid bilgileri elde edilmiştir.

```
stats=regionprops(L,I,'Centroid');  
cg_centroids=cat(1,stats.Centroid);
```

Görüntüler incelendiğinde, ilk 60 framede varsayılan olarak gelen soldan sağa sıralamada label sıralamasında herhangi bir sorun gözükmemektedir. 60'ıncı ve 86'ncı framele arasında diz ve kalçadaki labelların sıralaması yer değiştirmektedir. 87'nci ve 89'uncu framele arasında bilek, diz ve kalçadaki labelların sıralaması yer değiştirmektedir. Bunu için önce 3. ve 5. labellar arasında, daha sonra 3. ve 4. Labellar arasında swap işlemi yapılmıştır. 90'ıncı frameden sonra sıralama işlemi y yönünde azalan olarak yapılmış, sadece 90. frame'de bilek ve topuk labelları yer değiştirilmiştir.

```
if n>=60 && n<=86
```

```
[cg_centroids(4,:),cg_centroids(5,)] = deal(cg_centroids(5,:),cg_centroids(4,:));
```

```
elseif n>=87 && n<=89
```

```

[cg_centroids(3,:),cg_centroids(5,.)]=deal(cg_centroids(5,:),cg_centroids(3,));

[cg_centroids(3,:),cg_centroids(4,.)]=deal(cg_centroids(4,:),cg_centroids(3,));
elseif n>=90
    cg_centroids=sortrows(cg_centroids,2,'descend');
    if n==90

[cg_centroids(2,:),cg_centroids(3,.)]=deal(cg_centroids(3,:),cg_centroids(2,));
    end
end

```

noktalar isminde bir struct oluşturularak elde edilen centroid bilgileri bu structın içinde ilgili marker noktası adıyla kaydedilmiştir.

```

noktalar.ayak_uclu(n,:)=cg_centroids(1,:);
noktalar.topuk(n,:)=cg_centroids(2,:);
noktalar.bilek(n,:)=cg_centroids(3,:);
noktalar.diz(n,:)=cg_centroids(4,:);
if n<48 || n>51
    noktalar.kalca(n,:)=cg_centroids(5,:);
end

```

48. ve 51. Frameler arasında kalça markerının önünden sporcunun elinin geçmesi nedeniyle markerın centroid bilgisi bulunamamaktadır. Bunun için bu frameler dışında elde edilen centroid bilgileri vasıtasıyla bir cubic spline uydurulmuş ve 48-51 frameler arasında kalça markerının pozisyonu bu spline vasıtasıyla hesaplanmıştır.

```

%kalcanin 48-51 frameleri arasindaki pozisyonlarini cubic spline
yontemi
%ile bulacagiz
noktalar.kalca(48:51,:)=[];
xq=[48,49,50,51];
x=ones(105,1);
x(:,1)=1:105;
x(48:51,:)=[];
sx=spline(x,noktalar.kalca(:,1),xq);
sy=spline(x,noktalar.kalca(:,2),xq);
eklenecekler=zeros(4,2);
for i=1:4
    eklenecekler(i,1)=sx(i);
    eklenecekler(i,2)=sy(i);
end
temp=zeros(105,2);
temp(1:47,:)=noktalar.kalca(1:47,:);
temp(48:51,:)=eklenecekler;
temp(52:end,:)=noktalar.kalca(48:end,:);
noktalar.kalca=temp;

```

2. Hesaplanan Geometrik Merkezlerin Üzerlerine Farklı Renklerde Yuvarlak İşaretler Yerleştirilmesi, Koordinatlarının Yazılması

Renk paleti olarak 'sarı,yeşil,mavi,magenta,cyan' renkleri seçilmiştir.

Ayak Ucu =Sarı

Topuk= Yeşil

Bilek=Mavi

Diz=Magenta

Kalça=Cyan

Olacak şekilde renklendirilecektir. Sprintf komutu ile kullanabilmek amacıyla noktalar struct'ı cell objesine dönüştürülmüştür.

```
renk='ygbmc';
cell_noktalar=struct2cell(noktalar);
for i=1:length(cell_noktalar)
    centroid=cell_noktalar{i};
    x=centroid(n,1);
    y=centroid(n,2);
    plot(x,y,'o','Color',renk(i));
    text(x+5,y,sprintf('%3.3f,%3.3f',x,y),'Color',renk(i),'FontSize',12);
end
```

3. Markerların arasına beyaz çizgi çizilmesi ve Videonun oluşturulması

Cell_noktalar hücre yapısı içerisinde hücrenin uzunluğunun bir eksiği kadar döngü oluşturulmuştur(bir sonraki noktanın indeksi kullanıldığı için). Line komutu ile bu noktalar arasında kesikli çizgi çizilmiştir.

```
for i=1:length(cell_noktalar)-1
    centroid=cell_noktalar{i};
    centroid2=cell_noktalar{i+1};
    line([centroid(n,1),centroid2(n,1)],[centroid(n,2),centroid2(n,2)],'Color','white','LineStyle','--');
end
```

Videonun oluşturulması için mp4 formatı kullanılmıştır. FrameRate 25 olarak seçilmiştir.

Yine bütün resimler tek tek bir for döngüsünün içerisinde açılarak üzerlerine marker bilgileri ve markerları birleştiren çizgiler çizilerek kaydedilmiştir. Elde edilen video sut.mp4 ismiyle ekteedir.

```
%% Video görüntülerinin olusturulmasi islemi
video=false;
if video
    aviobj=VideoWriter('sut.mp4','MPEG-4');
    aviobj.FrameRate=25;
    aviobj.Quality=100;
    open(aviobj);
    renk='ygbmc';
```

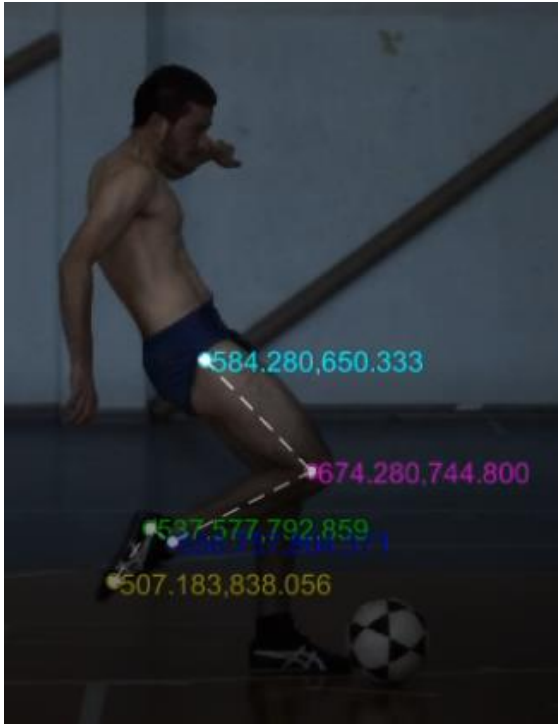
```

cell_noktalar=struct2cell(noktalar);
for n=1:105
    dosya_adi=strcat(dosya_on_ad,sprintf( '%06d', n )
,dosya_uzanti);
    RGB=imread(dosya_adi);
    imshow(RGB);
    hold on
    for i=1:length(cell_noktalar)
        centroid=cell_noktalar{i};
        x=centroid(n,1);
        y=centroid(n,2);
        plot(x,y,'o','Color',renk(i));

        text(x+5,y,sprintf('%3.3f,%3.3f',x,y),'Color',renk(i),'FontSize',12);
    end
    for i=1:length(cell_noktalar)-1
        centroid=cell_noktalar{i};
        centroid2=cell_noktalar{i+1};

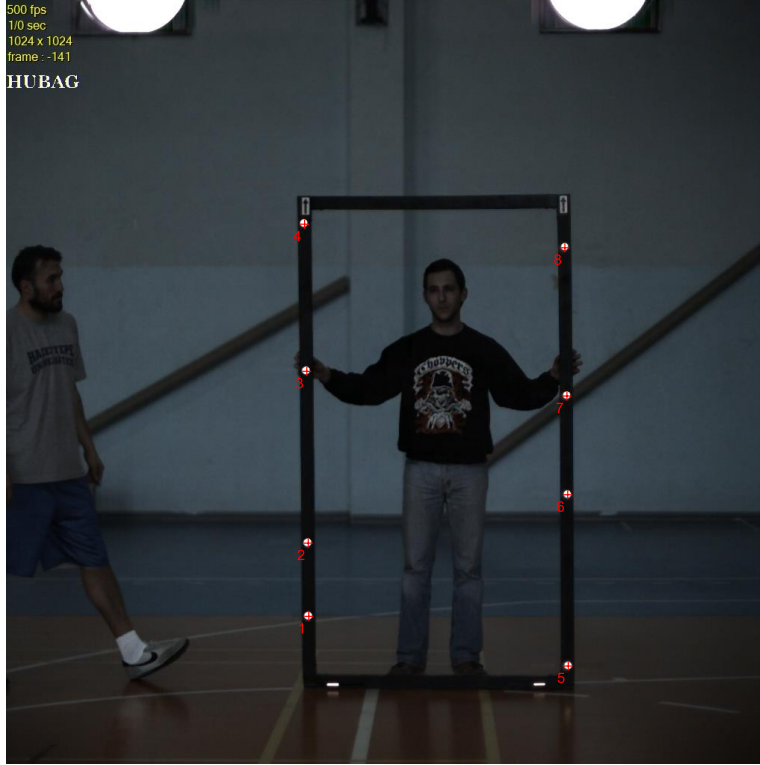
        line([centroid(n,1),centroid2(n,1)],[centroid(n,2),centroid2(n,2)],'Color','white','LineStyle','--');
    end
    frame=getframe(gcf);
    writeVideo(aviobj,frame);
    hold off
end
end
if video
    close(aviobj);
end

```



Resim 6 Markerların pozisyon bilgileri ve markerları birleştiren çizgiler

4. Kalibrasyon işlemi ve Noktaların Dünya koordinat sistemine aktarılması



Resim 7 Kalibrasyon resminde markerların işaretlenmesi

Kalibrasyon için Kalibrasyon.jpg dosyasında 1. Adımdaki işlemler tekrarlanarak markerların centroid konumları kalib_centroids matrisine aktarılmıştır.

```
%% Kalibrasyon islemi
kalibRGB=imread('kalibrasyon.jpg');
I=rgb2gray(kalibRGB);
se=strel('disk',15);
background=imopen(I,se);
I2=I-background;
I3=imadjust(I2,[0.3 0.7],[0]);
[T EM]=graythresh(I3);
BW=imbinarize(I3,T);

BW(:,1:135)=0;
BW=medfilt2(BW,[3 3]);

BW = bwareaopen(BW,50);
imshow(kalibRGB)
hold on
[B,L]=bwboundaries(BW,'noholes');
kalib_stats=regionprops(BW,'Centroid');
kalib_centroids=cat(1,kalib_stats.Centroid);
```

```

kalib_centroids=sortrows(kalib_centroids,[1 2],{'descend','descend'});
[kalib_centroids(1,:),kalib_centroids(5,:)]=deal(kalib_centroids(5,:),
...
    kalib_centroids(1,:));
[kalib_centroids(2,:),kalib_centroids(6,:)]=deal(kalib_centroids(6,:),
...
    kalib_centroids(2,:));
[kalib_centroids(3,:),kalib_centroids(7,:)]=deal(kalib_centroids(7,:),
...
    kalib_centroids(3,:));
[kalib_centroids(4,:),kalib_centroids(8,:)]=deal(kalib_centroids(8,:),
...
    kalib_centroids(4,:));
for i=1:length(kalib_centroids)
    centroid=kalib_centroids(i,:);
    plot(centroid(1),centroid(2),'r+');
    text(centroid(1)-15,centroid(2)+15,num2str(i),'Color','r');
end
hold off

```

Dünya koordinat sistemine geçiş için ilk önce noktaların y bileşenleri, resimlerin pixel olarak yüksekliği olan 1024'ten çıkartılarak y ekseninin yönü değiştirilmiştir.

Calculate_conformal() fonksiyonu ile birim dönüştürme işlemi için gerekli olan parametreler hesaplanarak teta,scale,Tx,Ty değerlerine atanmıştır.

Daha sonra 1. Adımda bulunan pozisyon bilgileri filtreleme işleminden geçirilerek gürültünün azaltılması hedeflenmiştir. Bunun için butter metodu kullanılmıştır. Filtre sayısı 2 gidiş, 2 geliş olmak üzere toplamda 4 seçilmiştir. Veri toplama hızı saniyede 500 kare olduğundan ve gürültüyü önlemek istediğimiz frekansın hızlı bir el hareketine benzerliğinden dolayı 15/250 parametresi kullanılmıştır. Filtremizin geçirgenlik parametresi 'low' olarak tanımlanmıştır.

```

kalib_centroids(:,2)=1024-kalib_centroids(:,2);
noktalar.kalca(:,2)=1024-noktalar.kalca(:,2);
noktalar.diz(:,2)=1024-noktalar.diz(:,2);
noktalar.bilek(:,2)=1024-noktalar.bilek(:,2);
noktalar.topuk(:,2)=1024-noktalar.topuk(:,2);
noktalar.ayak_uclu(:,2)=1024-noktalar.ayak_uclu(:,2);
S=[2.5 30;
    2.5 60;
    2.5 130;
    2.5 190;
    108 10;
    108 80;
    108 120;
    108 180];
I=kalib_centroids;
x=calculate_conformal(I,S,1);
teta=atand(x(2)/x(1));
scale=x(1)/cosd(teta);
Tx=x(3);
Ty=x(4);
delta_t=1/500;
[b,a]=butter(2,15/250,'low');
filtered_kalca=filtfilt(b,a,noktalar.kalca);
filtered_diz=filtfilt(b,a,noktalar.diz);

```

```

filtered_bilek=filtfilt(b,a,noktalar.bilek);
filtered_topuk=filtfilt(b,a,noktalar.topuk);
filtered_ayak_uclu=filtfilt(b,a,noktalar.ayak_uclu);
gercek_konumlar.ayakucu=calculate_reconformal(x,filtered_ayak_uclu);
gercek_konumlar.topuk=calculate_reconformal(x,filtered_topuk);
gercek_konumlar.bilek=calculate_reconformal(x,filtered_bilek);
gercek_konumlar.diz=calculate_reconformal(x,filtered_diz);
gercek_konumlar.kalca=calculate_reconformal(x,filtered_kalca);

```

Filtreden geçirilerek hesaplanan pozisyon bilgileri kullanılarak dünya koordinat sistemine aktarilarak gercek_konumlar isimli bir struct yapısının içine aktarılmıştır.

velocity_central_diff ve accl4s_central_diff fonksiyonları vasıtasıyla hızlar ve ivmeler hesaplanmıştır.

```

hizlar.ayakucu=velocity_central_diff(gercek_konumlar.ayakucu/100,delta_t);
hizlar.topuk=velocity_central_diff(gercek_konumlar.topuk/100,delta_t);
hizlar.bilek=velocity_central_diff(gercek_konumlar.bilek/100,delta_t);
hizlar.diz=velocity_central_diff(gercek_konumlar.diz/100,delta_t);
hizlar.kalca=velocity_central_diff(gercek_konumlar.kalca/100,delta_t);
ivmeler.ayakucu=accl4s_central_diff(gercek_konumlar.ayakucu/100,delta_t);
ivmeler.topuk=accl4s_central_diff(gercek_konumlar.topuk/100,delta_t);
ivmeler.bilek=accl4s_central_diff(gercek_konumlar.bilek/100,delta_t);
ivmeler.diz=accl4s_central_diff(gercek_konumlar.diz/100,delta_t);
ivmeler.kalca=accl4s_central_diff(gercek_konumlar.kalca/100,delta_t);

```

5. Grafiklerin Elde Edilmesi

Plot komutu vasıtasıyla elde edilen pozisyon, hız ve ivmelerin grafikleri çıkartılmıştır.

```

%% grafikler
%pozisyon grafigi
plot(gercek_konumlar.ayakucu(:,1),gercek_konumlar.ayakucu(:,2));
hold on
plot(gercek_konumlar.topuk(:,1),gercek_konumlar.topuk(:,2));
plot(gercek_konumlar.bilek(:,1),gercek_konumlar.bilek(:,2));
plot(gercek_konumlar.diz(:,1),gercek_konumlar.diz(:,2));
plot(gercek_konumlar.kalca(:,1),gercek_konumlar.kalca(:,2));
title('Sut Marker Pozisyonlari');
xlabel('x(cm)');
ylabel('y(cm)');
legend('Ayak Ucu','Topuk','Bilek','Diz','Kalca');
axis([-150 200 0 200]);
grid on;
saveas(gcf,'Pozisyonlar.png');
hold off
%Ayak Ucu Hiz grafigi
plot(1:104,hizlar.ayakucu(:,1));
hold on
plot(1:104,hizlar.ayakucu(:,2));
title('Ayak Ucu Hizi (m/s)');
xlabel('Frames');

```

```

ylabel('m/s');
legend('Ayak Ucu Vx', 'Ayak Ucu Vy');
grid on;
saveas(gcf, 'AyakUcuHiz.png');
hold off
%Topuk Hiz grafigi
plot(1:104,hizlar.topuk(:,1));
hold on
plot(1:104,hizlar.topuk(:,2));
title('Topuk Hizi (m/s)');
xlabel('Frames');
ylabel('m/s');
legend('Topuk Vx', 'Topuk Vy');
grid on;
saveas(gcf, 'TopukHiz.png');
hold off
%Bilek Hiz grafigi
plot(1:104,hizlar.bilek(:,1));
hold on
plot(1:104,hizlar.bilek(:,2));
title('Bilek Hizi (m/s)');
xlabel('Frames');
ylabel('m/s');
legend('Bilek Vx', 'Bilek Vy');
grid on;
saveas(gcf, 'BilekHiz.png');
hold off
%Diz Hiz grafigi
plot(1:104,hizlar.diz(:,1));
hold on
plot(1:104,hizlar.diz(:,2));
title('Diz Hizi (m/s)');
xlabel('Frames');
ylabel('m/s');
legend('Diz Vx', 'Diz Vy');
grid on;
saveas(gcf, 'DizHiz.png');
hold off
%Kalca Hiz grafigi
plot(1:104,hizlar.kalca(:,1));
hold on
plot(1:104,hizlar.kalca(:,2));
title('Kalca Hizi (m/s)');
xlabel('Frames');
ylabel('m/s');
legend('Kalca Vx', 'Kalca Vy');
grid on;
saveas(gcf, 'KalcaHiz.png');
hold off
%AyakUcu ivme grafigi
plot(3:length(ivmeler.ayakucu),ivmeler.ayakucu(3:end,1));
hold on
plot(3:length(ivmeler.ayakucu),ivmeler.ayakucu(3:end,2));
title('Ivme AyakUcu (m/s/s)');
xlabel('Frames');
ylabel('m/s/s');
legend('AyakUcu Ax', 'AyakUcu Ay');
grid on;
saveas(gcf, 'AyakUcuIvme.png');

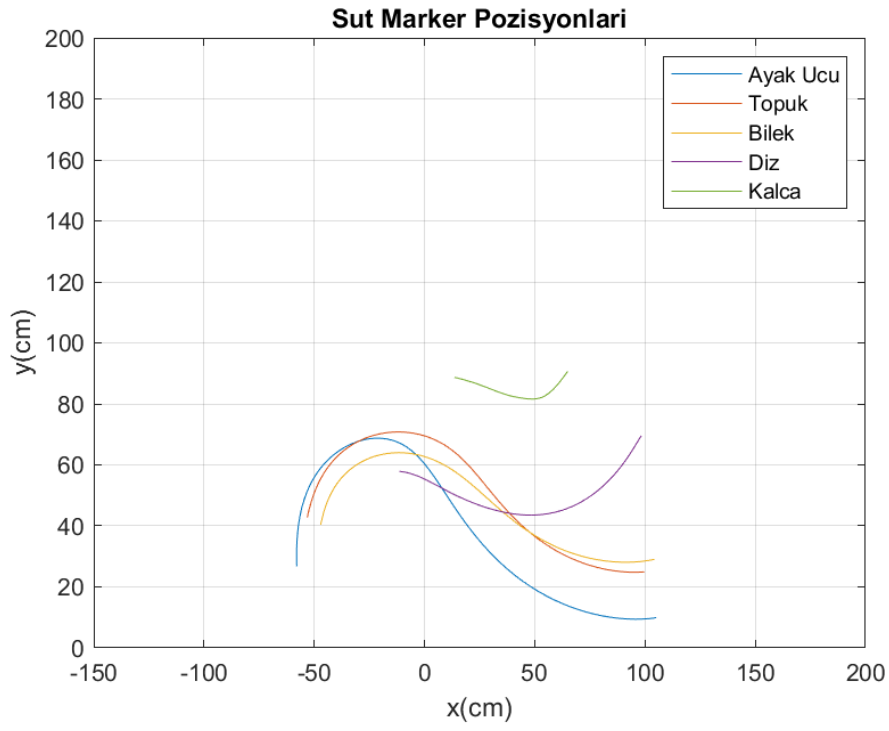
```

```

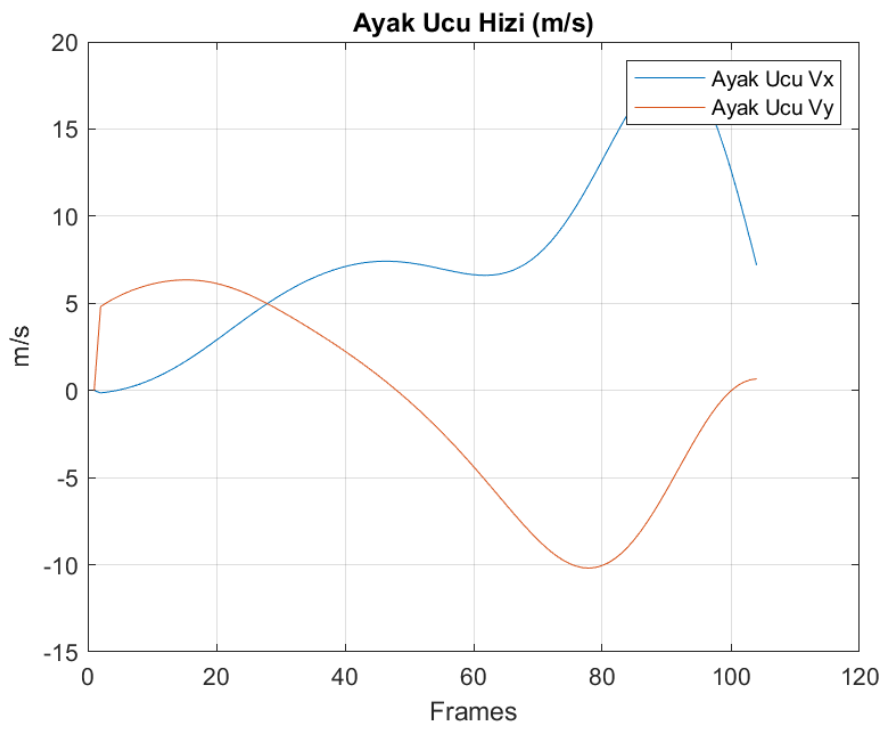
hold off
%Topuk ivme grafigi
plot(3:length(ivmeler.topuk),ivmeler.topuk(3:end,1));
hold on
plot(3:length(ivmeler.topuk),ivmeler.topuk(3:end,2));
title('Ivme Topuk (m/s/s)');
xlabel('Frames');
ylabel('m/s/s');
legend('Topuk Ax', 'Topuk Ay');
grid on;
saveas(gcf, 'TopukIvme.png');
hold off
%Bilek ivme grafigi
plot(3:length(ivmeler.bilek),ivmeler.bilek(3:end,1));
hold on
plot(3:length(ivmeler.bilek),ivmeler.bilek(3:end,2));
title('Ivme Bilek (m/s/s)');
xlabel('Frames');
ylabel('m/s/s');
legend('Bilek Ax', 'Bilek Ay');
grid on;
saveas(gcf, 'BilekIvme.png');
hold off
%Diz ivme grafigi
plot(3:length(ivmeler.diz),ivmeler.diz(3:end,1));
hold on
plot(3:length(ivmeler.diz),ivmeler.diz(3:end,2));
title('Ivme Diz (m/s/s)');
xlabel('Frames');
ylabel('m/s/s');
legend('Diz Ax', 'Diz Ay');
grid on;
saveas(gcf, 'DizIvme.png');
hold off
%Kalca ivme grafigi
plot(3:length(ivmeler.kalca),ivmeler.kalca(3:end,1));
hold on
plot(3:length(ivmeler.kalca),ivmeler.kalca(3:end,2));
title('Ivme Kalca (m/s/s)');
xlabel('Frames');
ylabel('m/s/s');
legend('Kalca Ax', 'Kalca Ay');
grid on;
saveas(gcf, 'KalcaIvme.png');
hold off

```

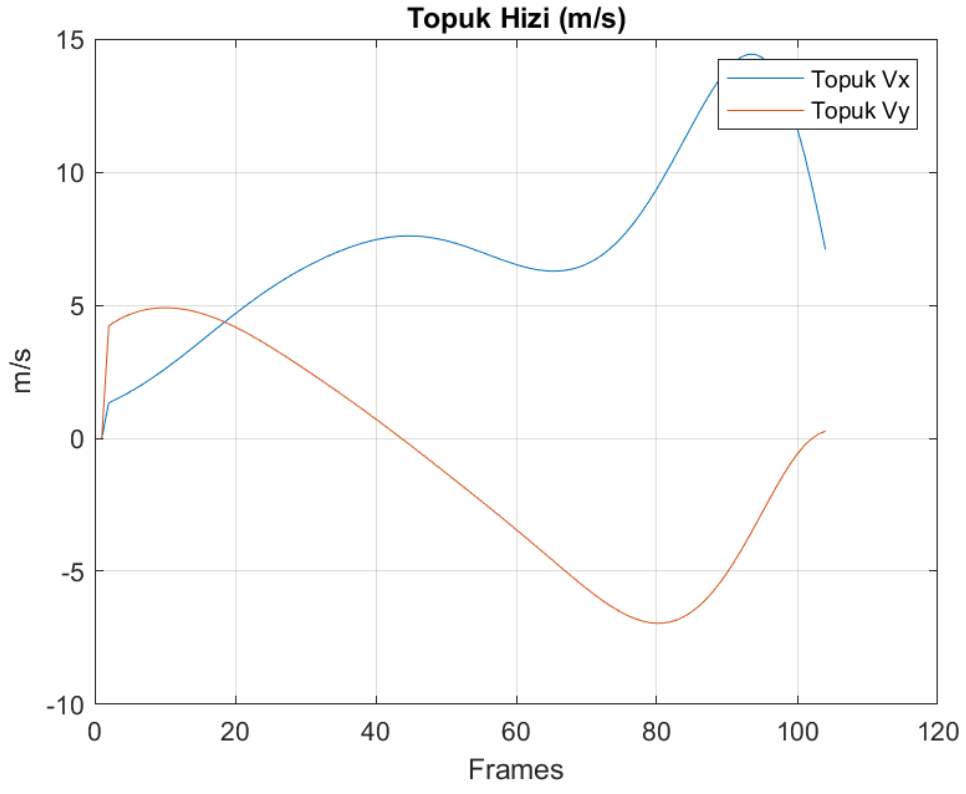
Elde edilen grafikler aşağıdaki gibidir.



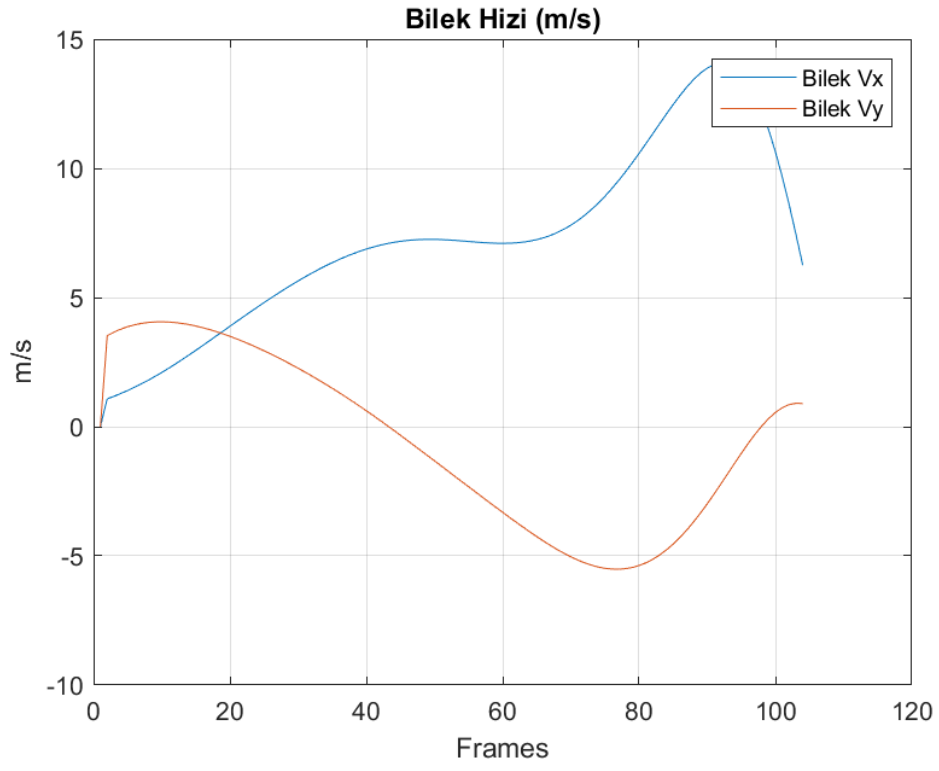
Grafik 1 Marker Pozisyonlari



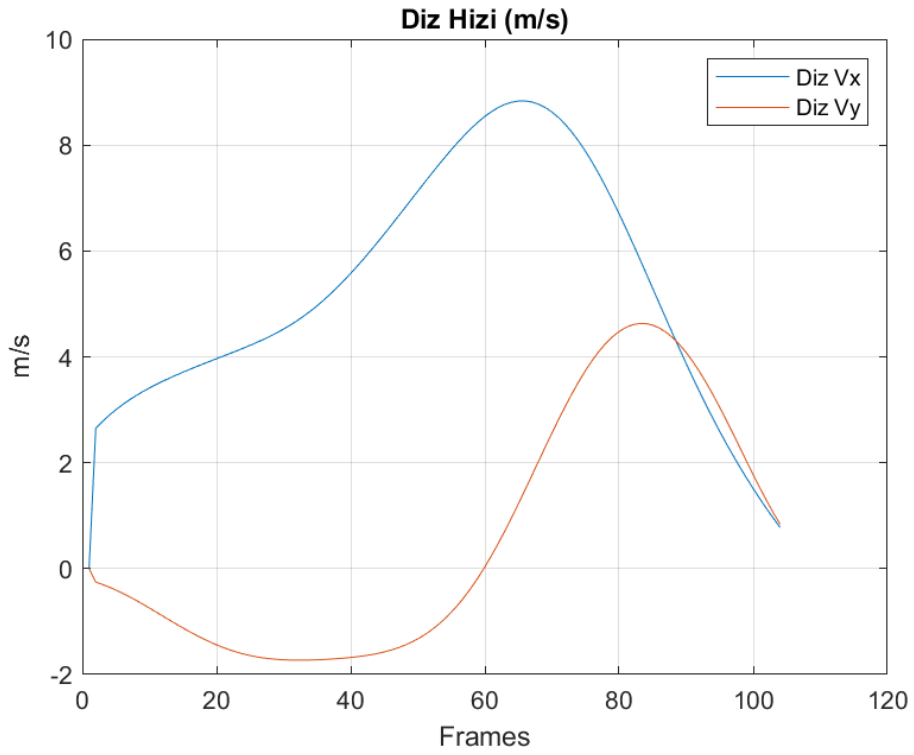
Grafik 2 Ayak Ucu Hız Değişimi



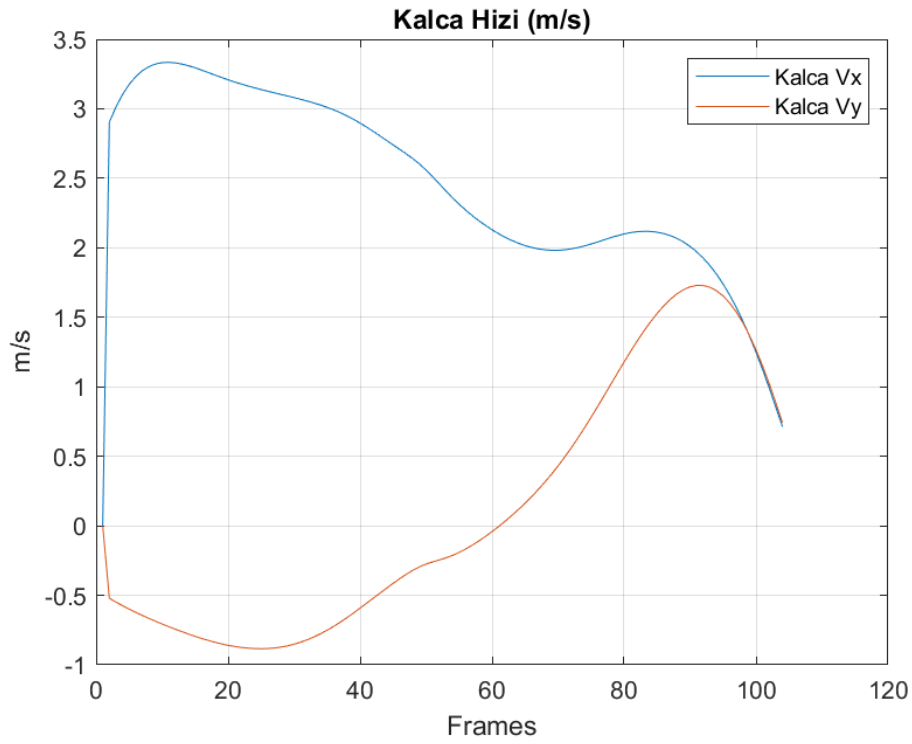
Grafik 3 Topuk Hız Değişimi



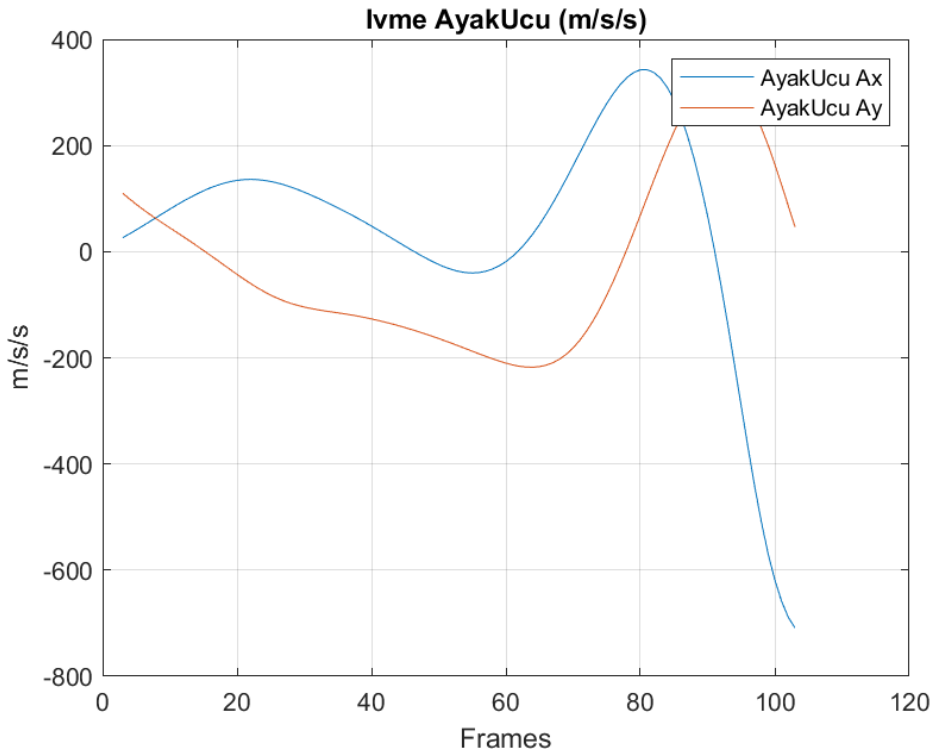
Grafik 4 Bilek Hız Değişimi



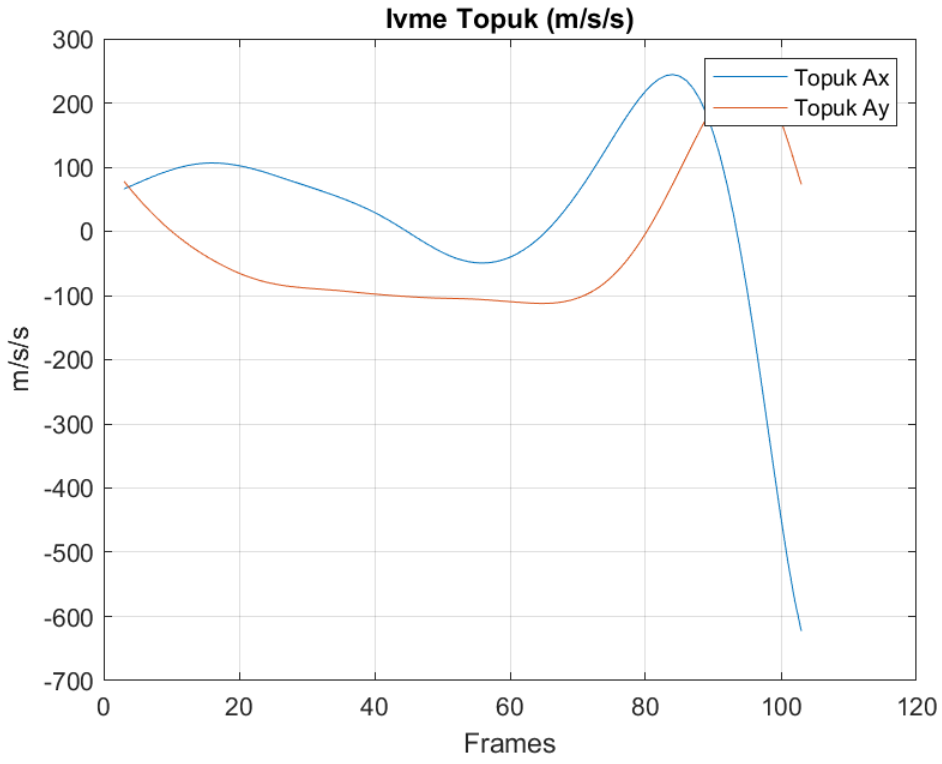
Grafik 5 Diz Hız Değişimi



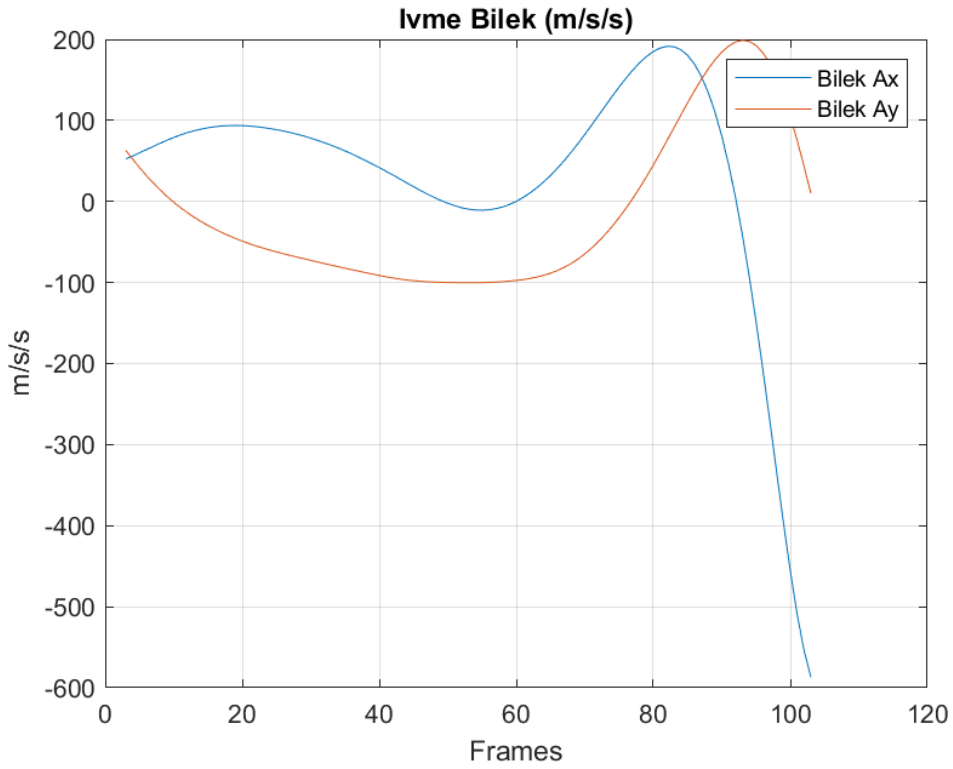
Grafik 6 Kalca Hiz Değişimi



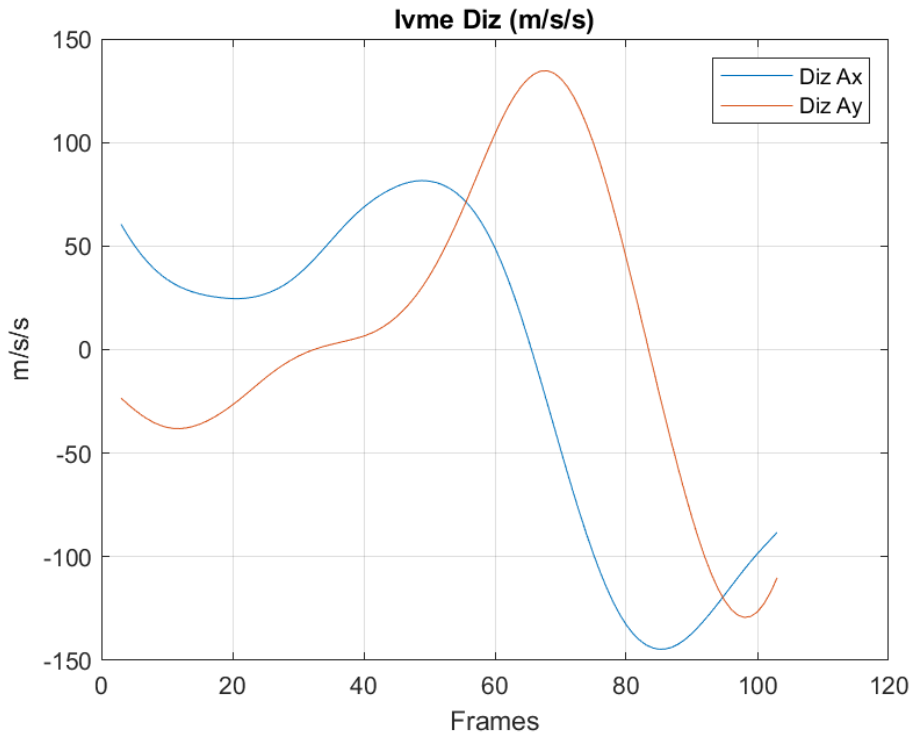
Grafik 7 Ayak Ucu İvme Değişimi



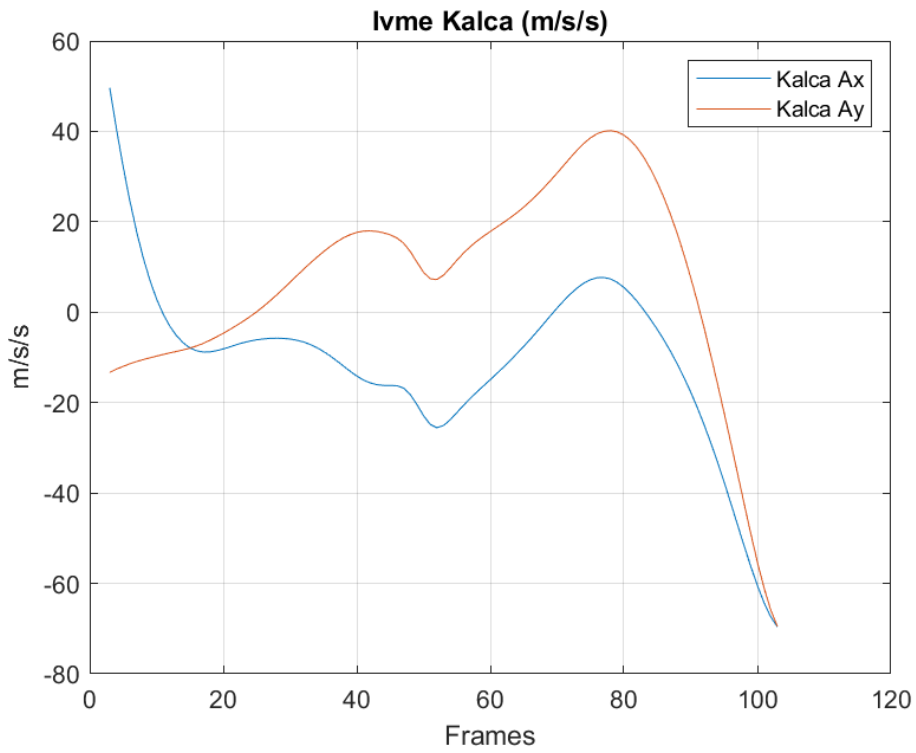
Grafik 8 Topuk İvme Değişimi



Grafik 9 Bilek İvme Değişimi



Grafik 10 Diz İvme Değişimi



Grafik 11 Kalca İvme Değişimi

6. Dizin Açısıl Hız Değişiminin Hesaplanması ve Grafiğinin Elde Edilmesi

Açısıl hızın tanımı: $\omega = \Delta\theta / \Delta t$ (birim zamandaki açı değişimi)

Bu tanımdan yola çıkarak her bir karede diz ve kalça pozisyonlarını kullanarak iki marker arasındaki açıyı hesapladım.

```
angle=atan((gercek_konumlar.kalca(:,2)-gercek_konumlar.diz(:,2))/...  
            (gercek_konumlar.kalca(:,1)-gercek_konumlar.diz(:,1)));  
angle=angle(:,105);
```

Bir for döngüsü yardımıyla açısıl hızları hesapladım.

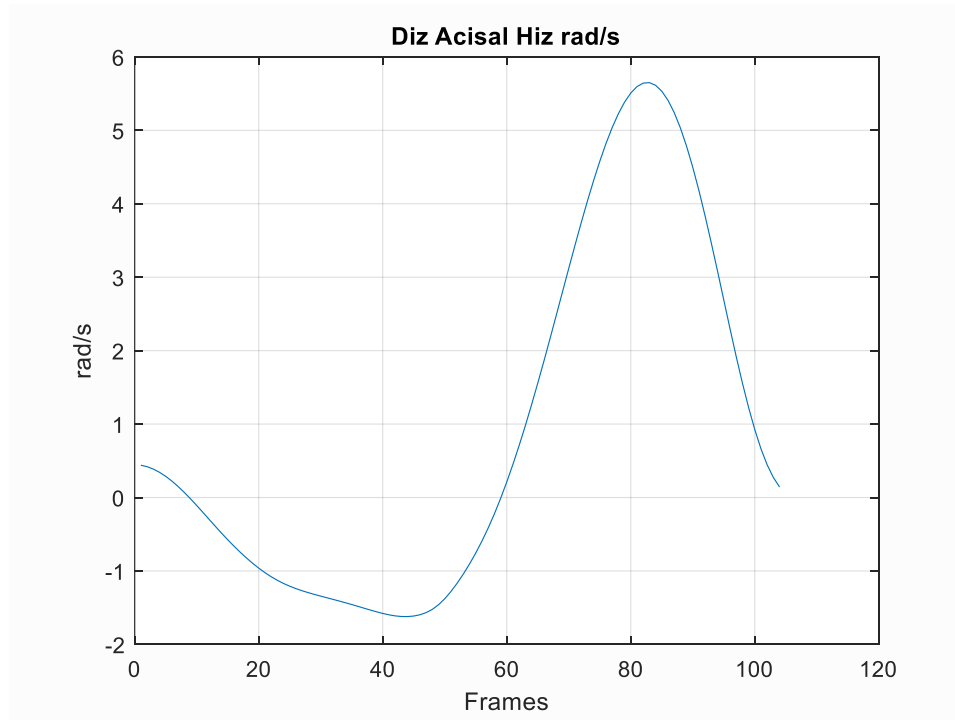
```
for i=1:104  
    acisal_hizlar(i,1)=(angle(i+1,1)-angle(i,1))/delta_t;  
end
```

Daha sonra plot komutuyla açısıl hızın değişimini çizdirdim.

```
plot(1:length(acisal_hizlar)-1,acisal_hizlar(1:end-1,1));  
hold on
```

```
title('Diz Acisal Hiz rad/s');  
xlabel('Frames');  
ylabel('rad/s');
```

```
grid on;  
saveas(gcf,'Diz Acisal Hiz.png');  
hold off
```



Grafik 12 Dizin Açısıl Hızının Değişimi