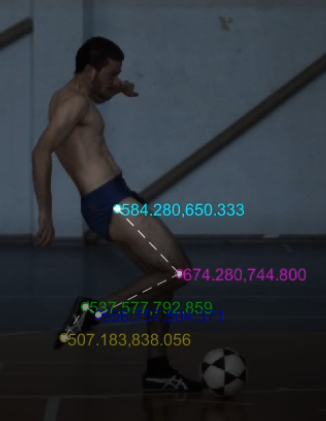
Tayfun GÜRLEVİK

N19139647

02 Aralık 2019

BCA 607 Hareket Analizi Sistemleri

ARA SINAV



İçindekiler

[1. Yansıtıcı İşaretlerin Geometrik Merkezlerinin (Centroid) koordinatlarının hesaplanması 2](#_Toc26263938)

[2. Hesaplanan Geometrik Merkezlerin Üzerlerine Farklı Renklerde Yuvarlak İşaretler Yerleştirilmesi, Koordinatlarının Yazılması 6](#_Toc26263939)

[3. Markerların arasına beyaz çizgi çizilmesi ve Videonun oluşturulması 6](#_Toc26263940)

[4. Kalibrasyon işlemi ve Noktaların Dünya koordinat sistemine aktarılması 8](#_Toc26263941)

[5. Grafiklerin Elde Edilmesi 10](#_Toc26263942)

[6. Dizin Açısal Hız Değişiminin Hesaplanması ve Grafiğinin Elde Edilmesi 19](#_Toc26263943)

[Resim 1 RGB resim 2](#_Toc26263966)

[Resim 2 Gri tonlamalı resim 2](#_Toc26263967)

[Resim 3 background resmi 3](#_Toc26263968)

[Resim 4 Gri tonlamalı resimden background çıkartılınca elde edilen sonuç 3](#_Toc26263969)

[Resim 5 Kontrost eklenmiş resim 4](#_Toc26263970)

[Resim 6 Markerların pozisyon bilgileri ve markerları birleştiren çizgiler 7](#_Toc26263971)

[Resim 7 Kalibrasyon resminde markerların işaretlenmesi 8](#_Toc26263972)

[Grafik 1 Marker Pozisyonları 13](#_Toc26263981)

[Grafik 2 Ayak Ucu Hız Degişimi 13](#_Toc26263982)

[Grafik 3 Topuk Hız Değişimi 14](#_Toc26263983)

[Grafik 4 Bilek Hız Değişimi 15](#_Toc26263984)

[Grafik 5 Diz Hız Değişimi 15](#_Toc26263985)

[Grafik 6 Kalca Hiz Değişimi 16](#_Toc26263986)

[Grafik 7 Ayak Ucu İvme Değişimi 16](#_Toc26263987)

[Grafik 8 Topuk İvme Değişimi 17](#_Toc26263988)

[Grafik 9 Bilek İvme Değişimi 17](#_Toc26263989)

[Grafik 10 Diz İvme Değişimi 18](#_Toc26263990)

[Grafik 11 Kalca İvme Değişimi 18](#_Toc26263991)

[Grafik 12 Dizin Açısal Hızının Değişimi 19](#_Toc26263992)

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 RGB resim

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

RGB=imread(dosya\_adi);

I=rgb2gray(RGB);



Resim Gri tonlamalı resim

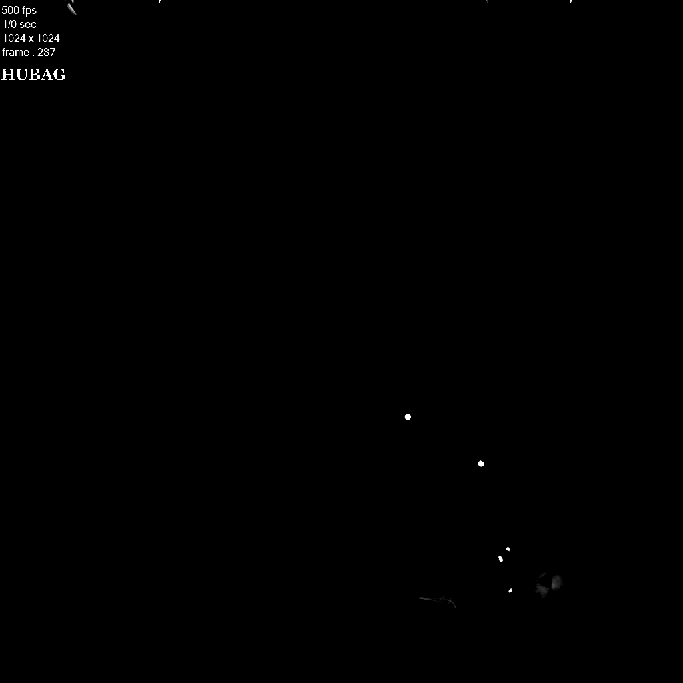
Aydınlatmadan kaynaklanan parlamaları resimden çıkarmak amacıyla ikl önce strel(‘disk’,15) komutu ile 15 pixel yarıçapında disk yapısal elemanı oluşturulmuştur. Daha sonra gri ölçekli resim bu yapısal elemanla açılarak arkaplan ışık görüntüsü elde edilmiştir(background). Gri ölçekli resimden background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



Resim background resmi



Resim Gri tonlamalı resimden background çıkartılınca elde edilen sonuç



Resim Kontrost 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 frameler arasında diz ve kalçadaki labelların sıralaması yer değiştirmektedir. 87’nci ve 89’uncu frameler 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\_ucu(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;

1. 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

1. 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 ektedir.

%% Video goruntulerinin 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

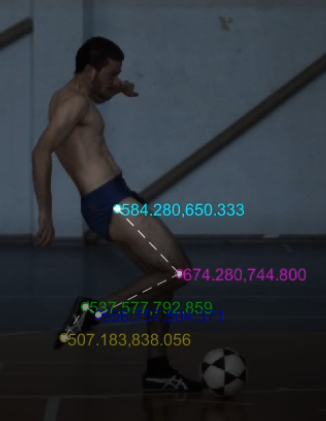
end

end

if video

close(aviobj);

end



Resim Markerların pozisyon bilgileri ve markerları birleştiren çizgiler

1. Kalibrasyon işlemi ve Noktaların Dünya koordinat sistemine aktarılması



Resim 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],[]);

[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() fonkisyonu 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\_ucu(:,2)=1024-noktalar.ayak\_ucu(:,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\_ucu=filtfilt(b,a,noktalar.ayak\_ucu);

gercek\_konumlar.ayakucu=calculate\_reconformal(x,filtered\_ayak\_ucu);

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 aktarılarak gerçek\_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);

1. 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');

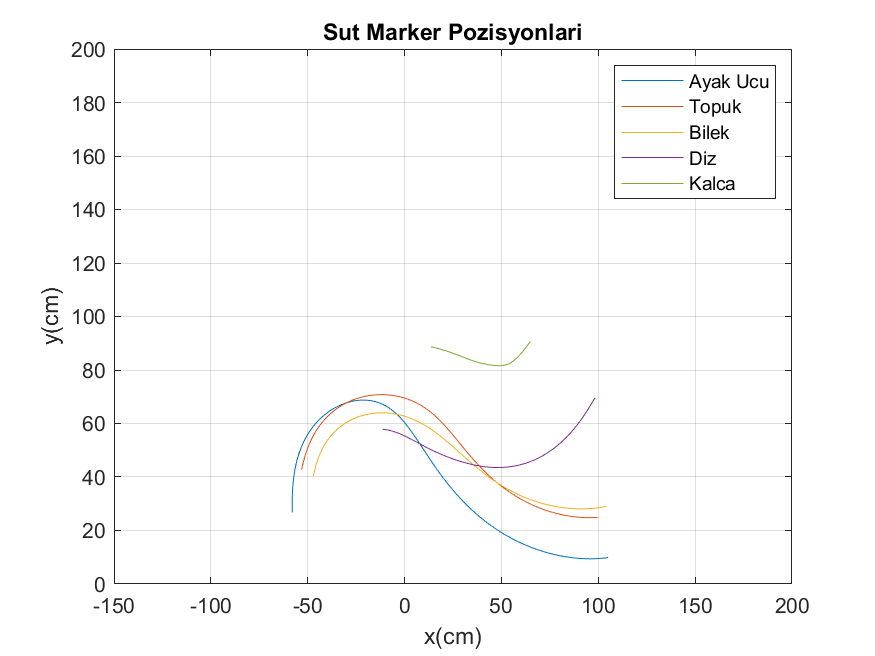
legend('Kalca Ax','Kalca Ay');

grid on;

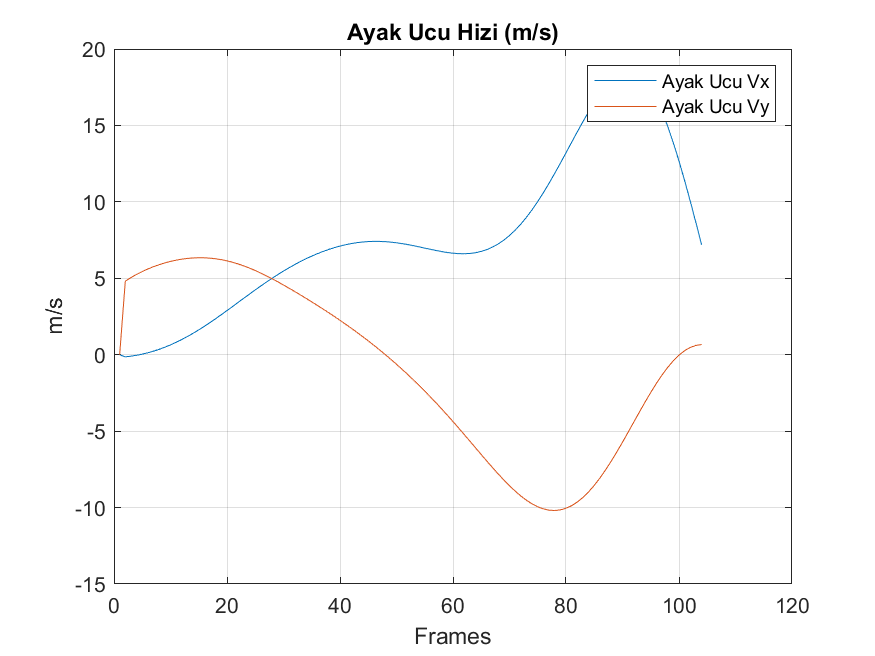
saveas(gcf,'KalcaIvme.png');

hold off

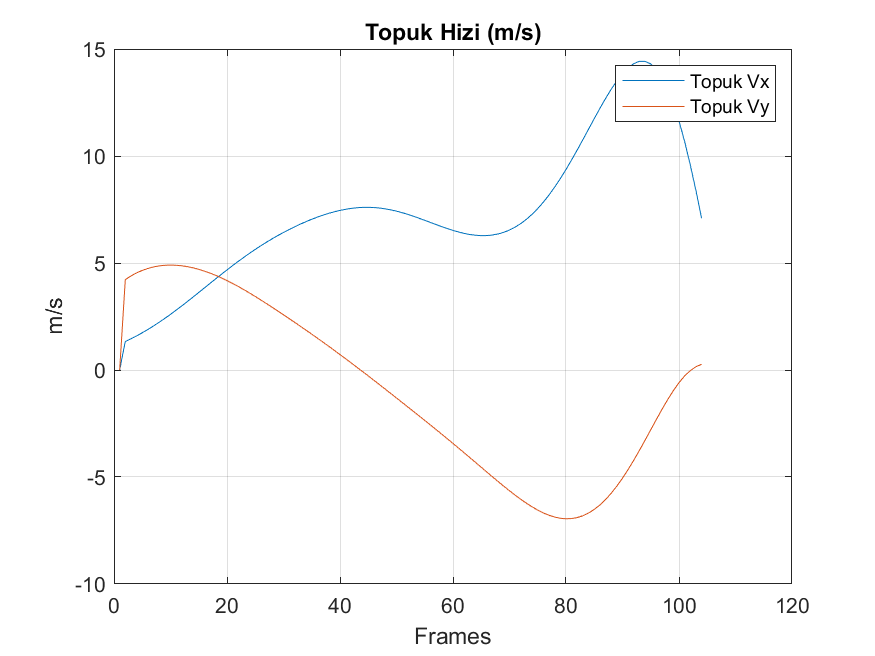
Elde edilen grafikler aşağıdaki gibidir.



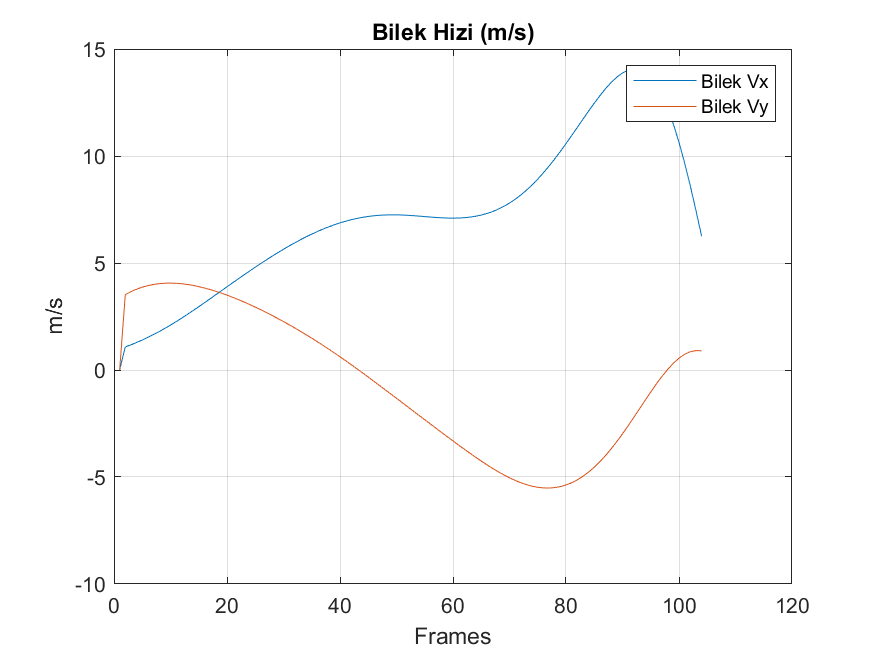
Grafik Marker Pozisyonları

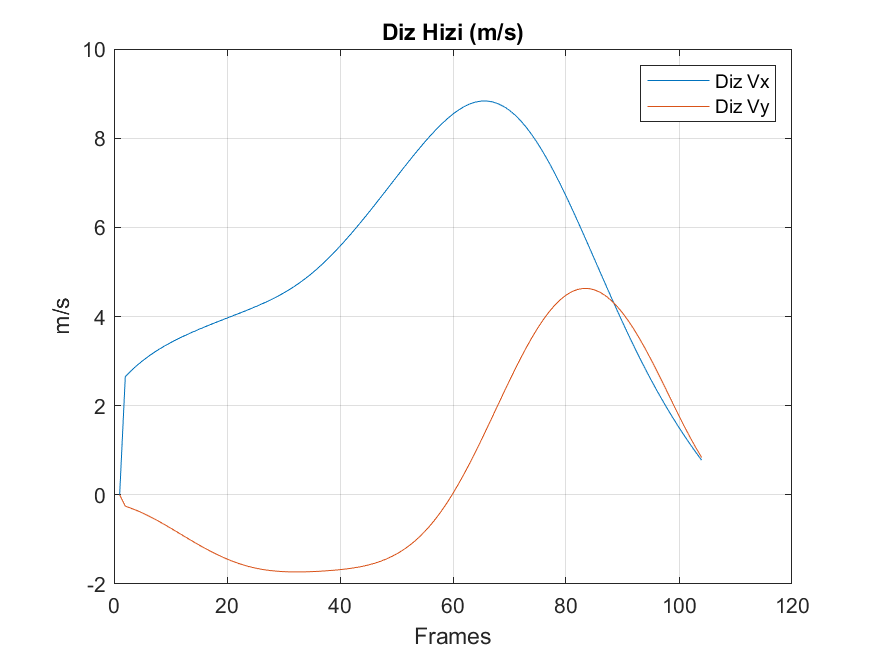


Grafik Ayak Ucu Hız Degişimi

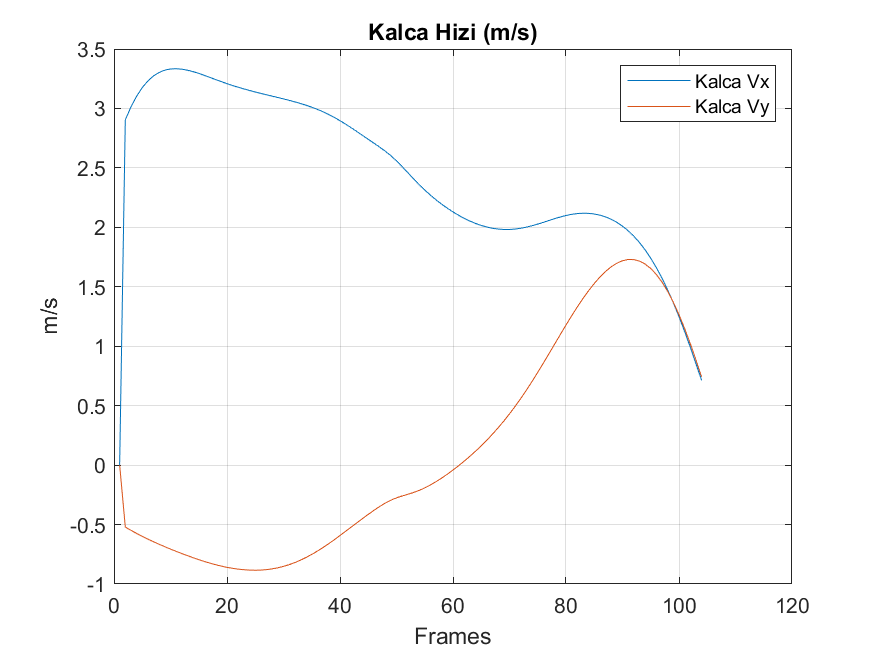


Grafik Topuk Hız Değişimi

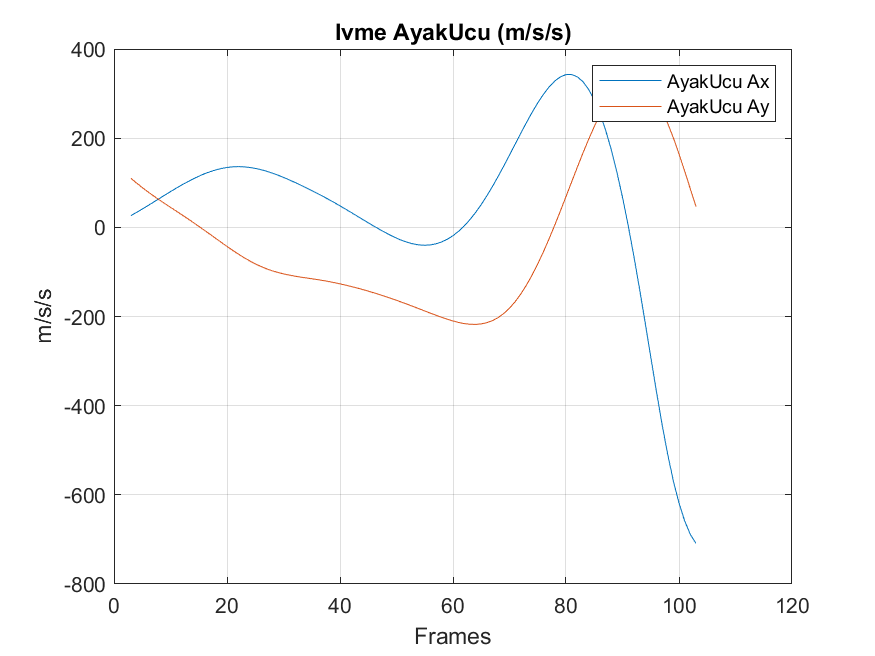


Grafik Bilek Hız Değişimi

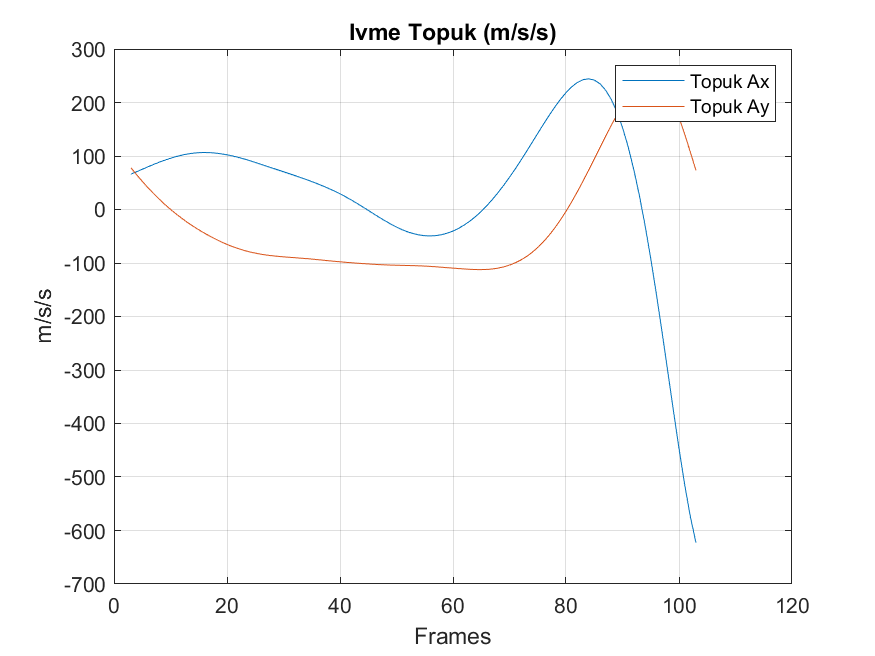
Grafik Diz Hız Değişimi



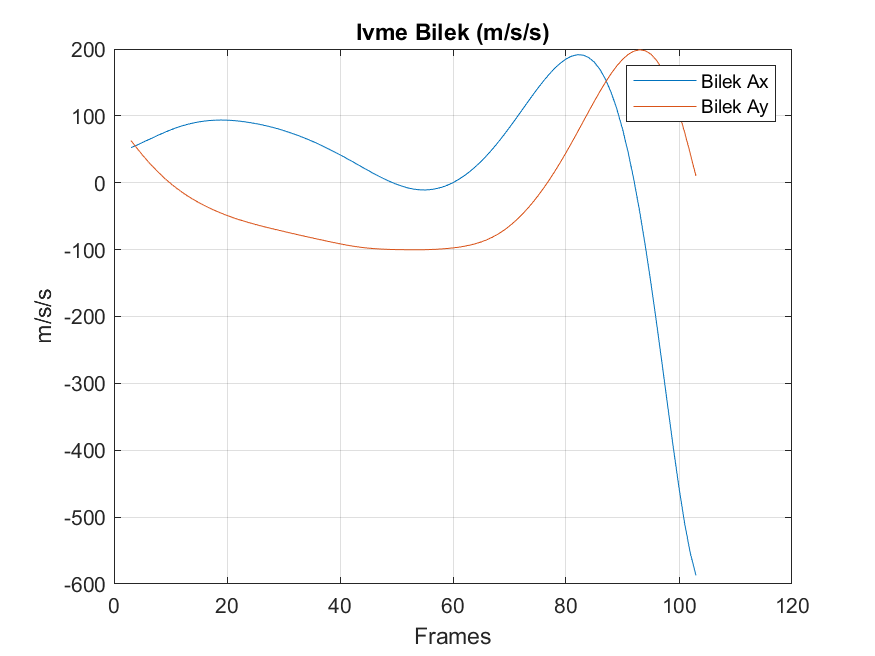
Grafik Kalca Hiz Değişimi



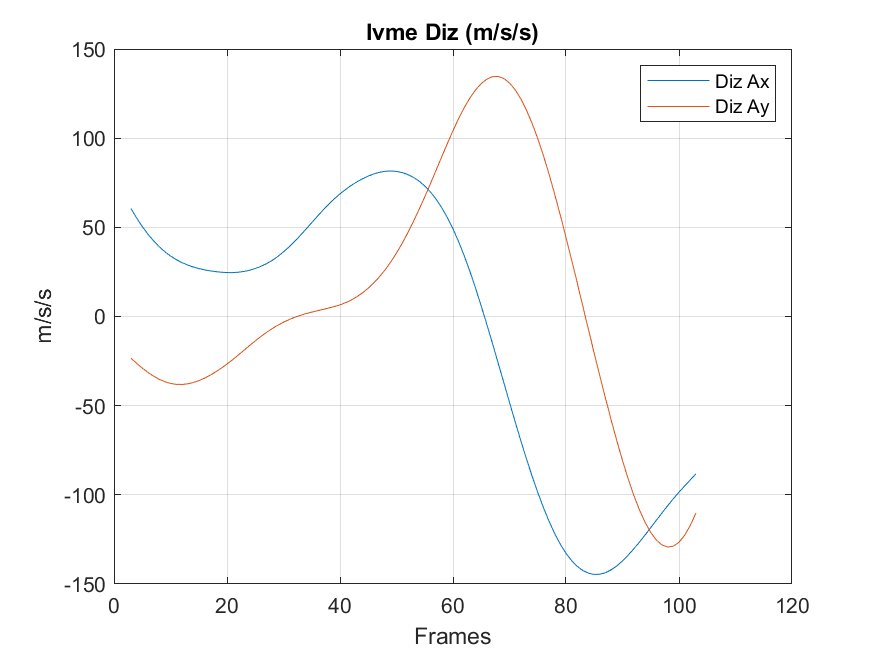
Grafik Ayak Ucu İvme Değişimi



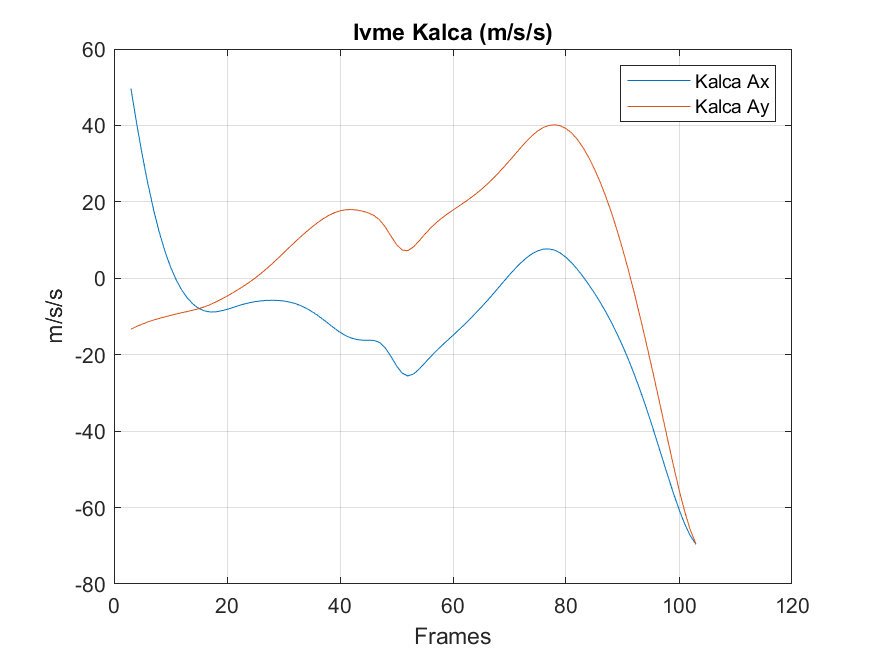
Grafik Topuk İvme Değişimi



Grafik Bilek İvme Değişimi



Grafik Diz İvme Değişimi



Grafik Kalca İvme Değişimi

1. Dizin Açısal Hız Değişiminin Hesaplanması ve Grafiğinin Elde Edilmesi

Açısal hızın tanımı: *ω*=∆θ/∆t (birim zamandaki açı değişimi)

Bu tanımdan yola çıkarak her bir karede diz ve kalça posiyonlarını 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 açısal hızları hesapladım.

for i=1:104

acisal\_hizlar(i,1)=(angle(i+1,1)-angle(i,1))/delta\_t;

end

Daha sonra plot komutuyla açısal 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 Dizin Açısal Hızının Değişimi