BCA607 Hareket Analizi Sistemleri

Final Raporu

Tayfun GÜRLEVİK

N19139647

# Bölüm

İlk olarak 13. Haftanın dosyalarında bulunan vicon\_all\_labelled.txt dosyasının matlab programında kullanbilecek şekilde virgülle ayrılmış formata dönüştürülmesi gerekmektedir. Bu amaçla C# yazılım dili kullanarak ilgili dosyadaki M1,M2,M3,M4 ve M5 isimleri ile etiketlenmiş markerların pozisyon bilgisini başka bir dosyaya yazdıracak program geliştirilmiştir.

Marker bilgilerini virgülle ayrılmış formatta parse eden programın kodu aşağıdaki gibidir:

using System;

using System.IO;

namespace ViconParser

{

class Program

{

static void Main(string[] args)

{

string filepath = "vicon\_all\_labelled.txt";

using (StreamReader reader = new StreamReader(filepath))

{

using (StreamWriter writer = new StreamWriter("vicon\_parsed.txt"))

{

do

{

string line = reader.ReadLine();

if (line.Contains("Frame Number:"))

{

int frameNumber = int.Parse(line.Split(':')[1].Trim());

Console.WriteLine(frameNumber);

writer.Write(frameNumber + ",");

}

if (line.Contains("Markers (5):"))

{

string marker1Line = reader.ReadLine();

int startIndex = marker1Line.IndexOf('(');

int endIndex = marker1Line.IndexOf(')');

string marker1Pozitions = marker1Line.Substring(startIndex + 1, endIndex - startIndex - 1);

Console.WriteLine(marker1Pozitions);

string marker2Line = reader.ReadLine();

startIndex = marker2Line.IndexOf('(');

endIndex = marker2Line.IndexOf(')');

string marker2Pozitions = marker2Line.Substring(startIndex + 1, endIndex - startIndex - 1);

Console.WriteLine(marker2Pozitions);

string marker3Line = reader.ReadLine();

startIndex = marker3Line.IndexOf('(');

endIndex = marker3Line.IndexOf(')');

string marker3Pozitions = marker3Line.Substring(startIndex + 1, endIndex - startIndex - 1);

Console.WriteLine(marker3Pozitions);

string marker4Line = reader.ReadLine();

startIndex = marker4Line.IndexOf('(');

endIndex = marker4Line.IndexOf(')');

string marker4Pozitions = marker4Line.Substring(startIndex + 1, endIndex - startIndex - 1);

Console.WriteLine(marker4Pozitions);

string marker5Line = reader.ReadLine();

startIndex = marker5Line.IndexOf('(');

endIndex = marker5Line.IndexOf(')');

string marker5Pozitions = marker5Line.Substring(startIndex + 1, endIndex - startIndex - 1);

Console.WriteLine(marker5Pozitions);

writer.WriteLine("{0},{1},{2},{3},{4}", marker1Pozitions, marker2Pozitions,

marker3Pozitions, marker4Pozitions, marker5Pozitions);

}

} while (!reader.EndOfStream);

}

}

}

}

}

Program dosyası ViconParser.zip içerisinde bulunabilir.

Daha sonra elde edilen virgülle ayrılmış formattaki dosya kullanılarak matlabda ilgili değişkenlere markerların pozisyon bilgileri aktarılmıştır. Elde edilen sonuç Grafik 1’de gösterilmiştir.

load 'vicon\_parsed.txt';

frames=vicon\_parsed(:,1);

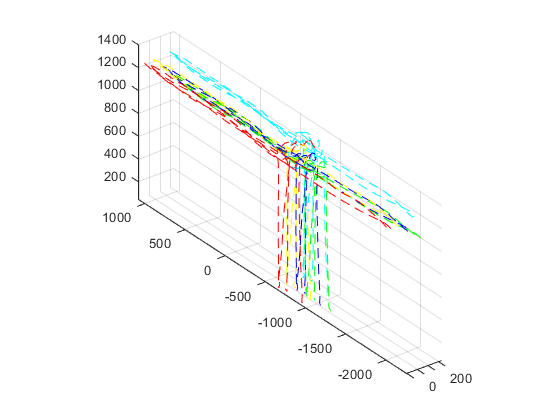
M1=vicon\_parsed(:,2:4);

M2=vicon\_parsed(:,5:7);

M3=vicon\_parsed(:,8:10);

M4=vicon\_parsed(:,11:13);

M5=vicon\_parsed(:,14:16);



Grafik 1 Markerların yörüngesi

M1, M3, M5 markerlarının pozisyon bilgisi kullanılarak fill3 komutu yardımıyla bir poligon (Grafik 2) oluşturulmuş ve animasyonu final.mp4 dosyasına kaydedilmiştir.

%% Video olusturma islemi

video=true;

if video

aviObj=VideoWriter('final.mp4','MPEG-4');

aviObj.FrameRate=25;

open(aviObj);

end

for i=1:length(frames)

plot3(M1(:,1),M1(:,3),M1(:,2),'--r')

grid on

axis equal

hold on

plot3(M2(:,1),M2(:,3),M2(:,2),'--g')

plot3(M3(:,1),M3(:,3),M3(:,2),'--b')

plot3(M4(:,1),M4(:,3),M4(:,2),'--y')

plot3(M5(:,1),M5(:,3),M5(:,2),'--c')

X(1,1)=M1(i,1);

X(1,2)=M3(i,1);

X(1,3)=M5(i,1);

Y(1,1)=M1(i,3);

Y(1,2)=M3(i,3);

Y(1,3)=M5(i,3);

Z(1,1)=M1(i,2);

Z(1,2)=M3(i,2);

Z(1,3)=M5(i,2);

fill3(X,Y,Z,'red');

if video

frame=getframe(gcf);

writeVideo(aviObj,frame);

end

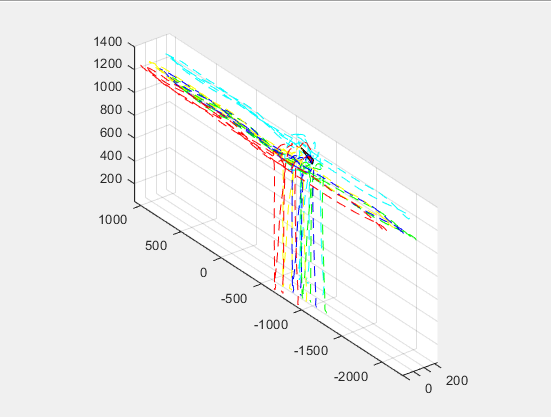
hold off

end

if video

close(aviObj);

end



Grafik 2 M1, M3, ve M5 markerları ile oluşturulan rigidbody

M1, M3 ve M5 markerlarının poziyonlarını kullanarak bir rigid body oluşturmak amacıyla P1, P2 ve P3 noktaları tanımlanmış olup bu noktalar vasıtasıyla rigidbodynin yerel koordinat sisteminin birim vektörleri bulunmuştur.

Hafta 10’da için Research Methods.pdf dosyasında anlatılan adımlar takip edilerek, Kalibrasyon çubuğunun hareketsiz durduğu ilk frame kalibrasyon karesi olarak kabul edilerek, CalCS( Kalibrasyon koordinat sistemi) ve bu matrisin tersinin , birim matris ile çarpımıyla da RTM(Rotational Transformation Matrix) hesaplanmıştır.

Her bir kare için PCS (Provisional Coordinate System) hesaplanarak RTM ile çarpımıyla SCS(segment coordinat system) hesaplanmıştır. Rotm2eul komutu yardımıyla SCS için euler açıları hesaplanmış, elde edilen vektörün bileşenleri ile de roll(Grafik 3), pitch(Grafik 4) ve yaw(Grafik 5) değerleri hesaplanarak grafikleri elde edilmiştir.

%% Roll,pitch ve yaw degerlerinin hesaplanmasi

roll=zeros(1,length(frames));

pitch=zeros(1,length(frames));

yaw=zeros(1,length(frames));

for n=1:length(frames)

P1=M1(n,:);

P2=M3(n,:);

P3=M5(n,:);

[P1(1,2),P1(1,3)]=deal(P1(1,3),P1(1,2));

[P2(1,2),P2(1,3)]=deal(P2(1,3),P2(1,2));

[P3(1,2),P3(1,3)]=deal(P3(1,3),P3(1,2));

v1=P2-P1;

v2=cross(P3-P1,v1);

i=v1/norm(v1);

j=v2/norm(v2);

k=cross(i,j);

PCS=[i(1),j(1),k(1);i(2),j(2),k(2);i(3),j(3),k(3)];

SCS=PCS\*RTM;

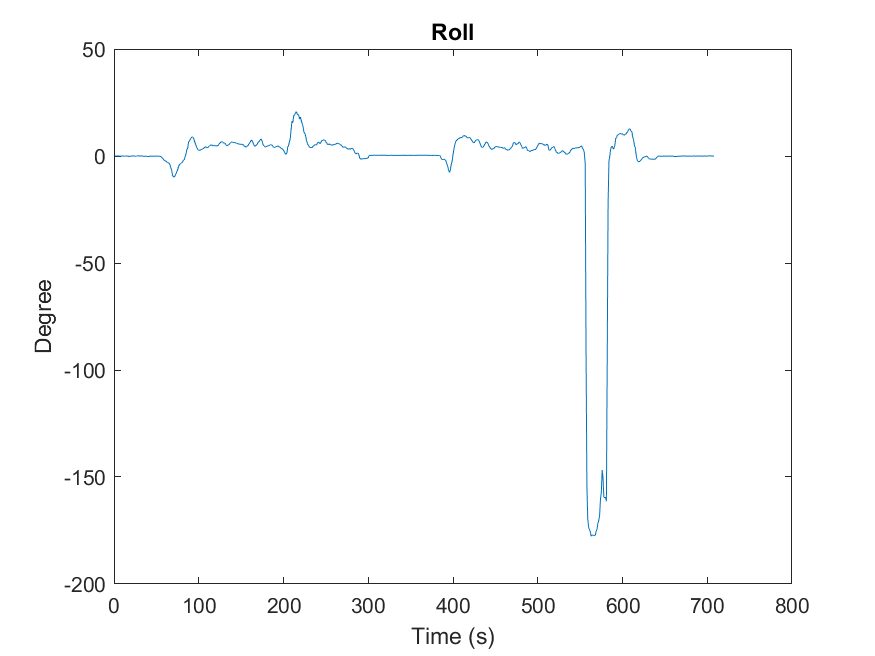
eul=rotm2eul(SCS,'XYZ');

roll(n)=eul(1);

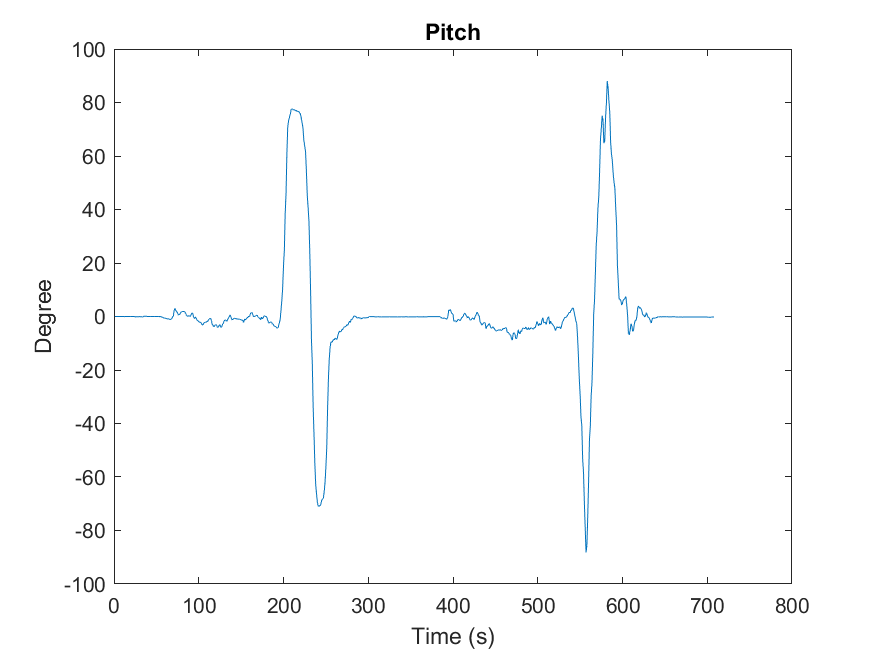
pitch(n)=eul(2);

yaw(n)=eul(3);

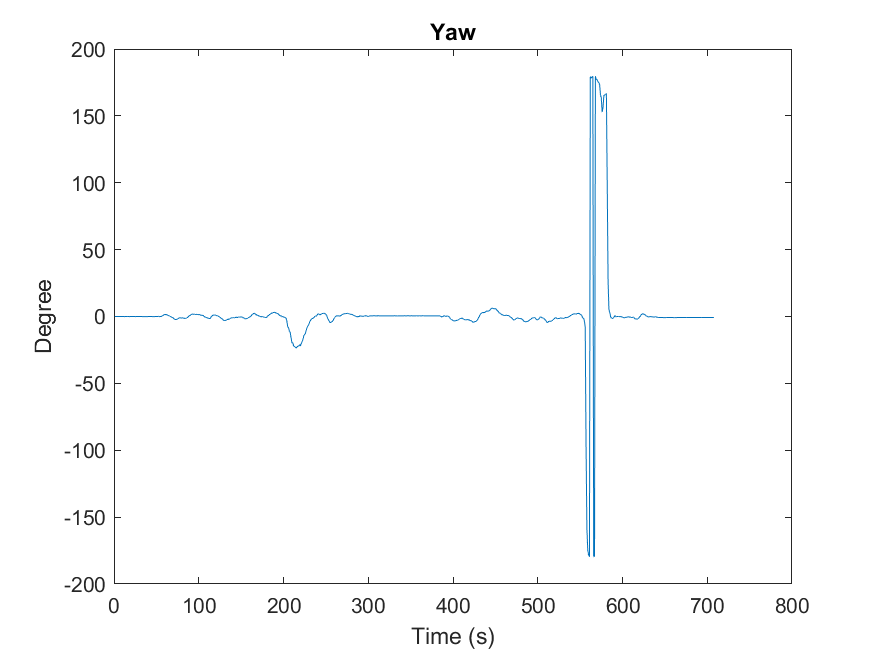
end



Grafik 3 Rigidbody Roll açısı



Grafik 4 Rigidbody Pitch Açısı



Grafik 5 Rigidbody Yaw Açısı

# Bölüm

İlk olarak phoneIMU.mat dosyası matlab alanına yüklenerek a(accelerometer) verileri alınmıştır.

Bileşke ivme,

|a|= formülü ile hesaplanmıştır.

Yerçekimi dünya üzerinde her yerde aynı olmadığından Hacettepe Üniversitesi, Matematik Bölümünün enlem ve boylam değerleri yardımıyla(39.8695973, 32.7367129) yerçekimi <https://www.sensorsone.com/local-gravity-calculator/> adresindeki hesaplama aracıyla 9.80148 m/s2 olarak bulunmuştur. Aşağıdaki kod yardımıyla bileşke ivme ve yerçekiminden arındırılmış bileşke ivme hesaplanmış ve grafikleri elde edilmiştir(Grafik 6, Grafik 7).

load('hafta13\phoneIMU.mat');

bileske\_ivme=sqrt(a(:,1).^2+a(:,2).^2+a(:,3).^2);

%bileske ivme

plot(t\_a,bileske\_ivme);

title('Raw Magnitude - No filter');

xlabel('Time(s)');

ylabel('Acceleration (m/s2)');

saveas(gcf,'bileske\_ivme.png');

g=9.80148;

a\_no\_g=bileske\_ivme-g;

%bileske ivme(yercekimsiz)

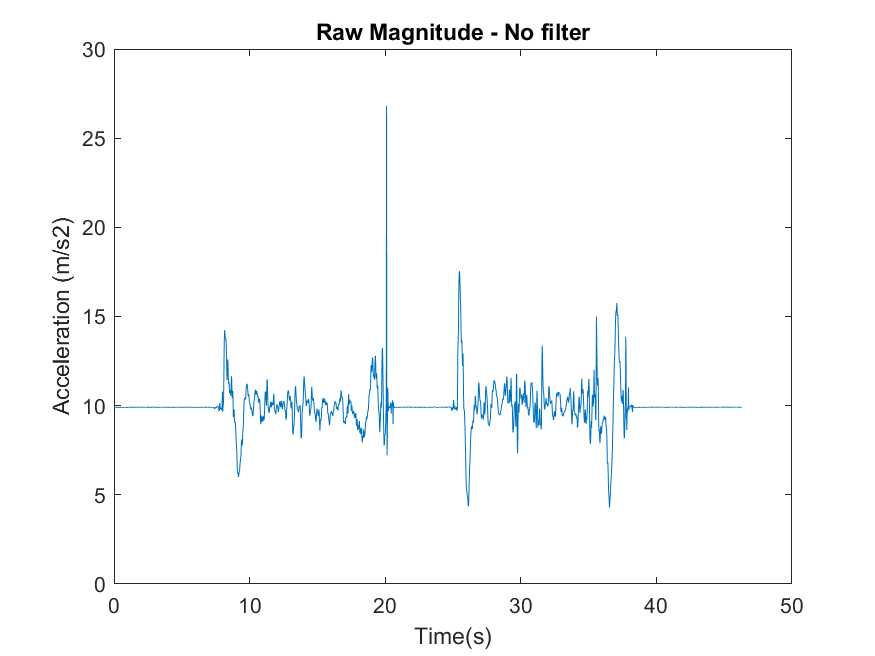
plot(t\_a,a\_no\_g);

title('No Gravity - No filter');

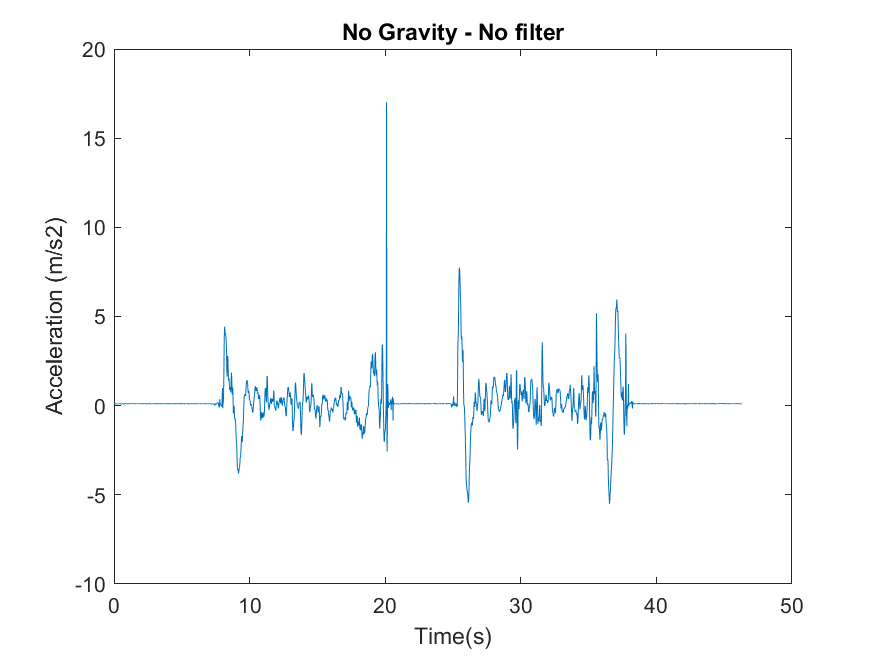
xlabel('Time(s)');

ylabel('Acceleration (m/s2)');

saveas(gcf,'bileske\_ivme\_yercekimsiz.png');



Grafik 6 Bileşke ivme



Grafik 7 Bileşke ivme(Yerçekimsiz)

İvmenin filtrelenmesi amacıyla öncelikle FFT(Fast Fourier transform) analizi yapılmıştır.

%% FFT incelemesi

Fs = 50; % Sampling frequency

T = 1/Fs; % Sampling period

L = length(t\_a); % Length of signal

t = t\_a; % Time vector

Y=fft(a);

P2 = abs(Y/L);

P1 = P2(1:L/2+1);

P1(2:end-1) = 2\*P1(2:end-1);

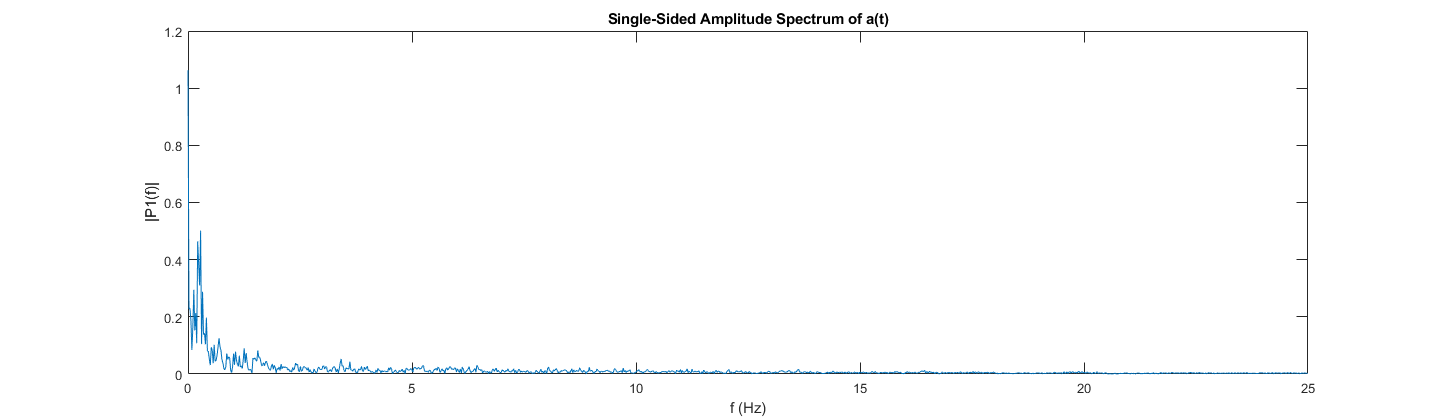
f = Fs\*(0:(L/2))/L;

plot(f,P1)

title('Single-Sided Amplitude Spectrum of a(t)')

xlabel('f (Hz)')

ylabel('|P1(f)|')



Grafik 8 İvmenin FFT specturumu

Grafik 8’de görüldüğü üzere gürültüyü azaltmak için kullanılacak filtrede cut-off frekansı olarak 2-2.5 Hz arasında bir frekans seçilmesi uygun olacaktır.

%% ivmenin filtrelenmesi

fc=2;

fs=1/0.02;

[c,d]=butter(1,fc/(fs/2),'low');

filtrelenmis\_ivme=filtfilt(c,d,a);

bileske\_ivme\_filtered=sqrt(filtrelenmis\_ivme(:,1).^2+filtrelenmis\_ivme(:,2).^2+filtrelenmis\_ivme(:,3).^2);

a\_no\_g\_filtered=bileske\_ivme\_filtered-g;

%filtrelenmis yercekimsiz ivme

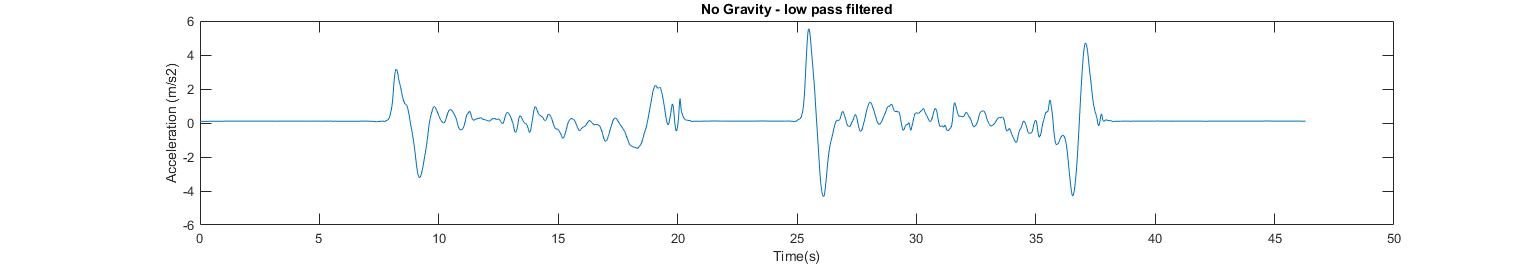
plot(t\_a,a\_no\_g\_filtered);

title('No Gravity - low pass filtered');

xlabel('Time(s)');

ylabel('Acceleration (m/s2)');

saveas(gcf,'filtrelenmis\_ivme.png');



Grafik 9 Filtrelenmis ivme(Yercekimsiz)