

# BCA 607 Hareket Analizi Sistemleri Final Projesi

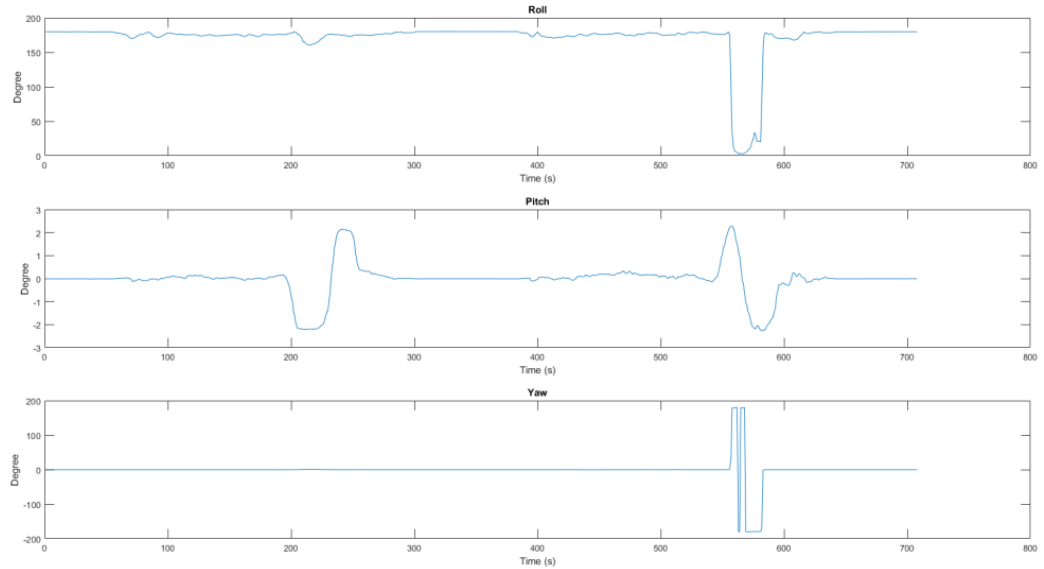
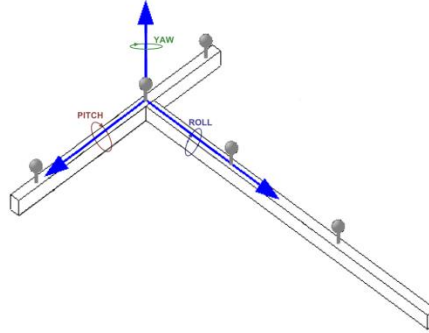
Projeniz 2 bölümden oluşmaktadır

## 1. Bölüm

Hafta 13 den IMU ve VICON veri setini indiriniz.

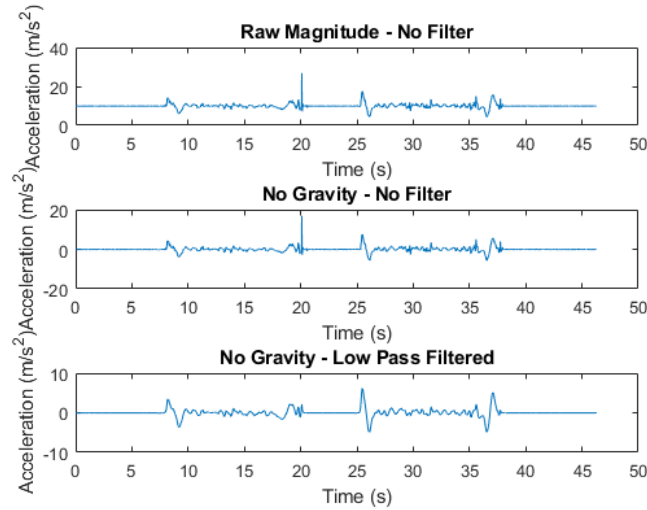
[http://yunus.hacettepe.edu.tr/~saritan/bca607/hafta\\_13.zip](http://yunus.hacettepe.edu.tr/~saritan/bca607/hafta_13.zip)

- Zip dosya içerisinde bulunan Etiketlenmiş Vicon (*vicon\_all\_labelled.txt*) veri setini kullanarak IMU (Inertial **M**esurament **U**nit) sensörünün bağlandığı kalibrasyon cubuğunun yörüngesini çiziniz.
- 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.

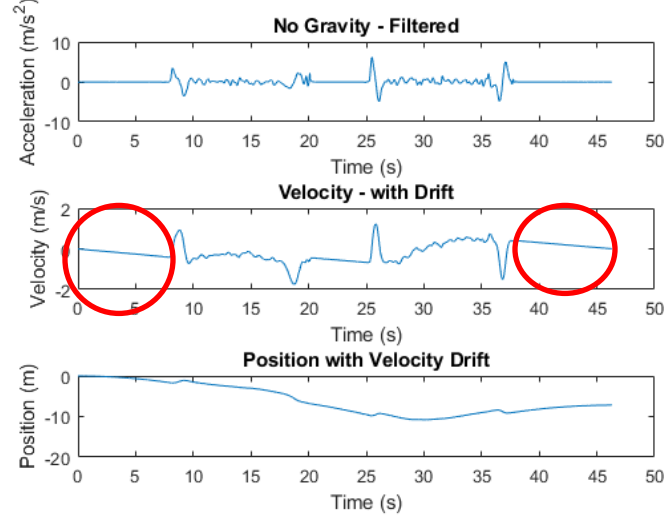


## 2. Bölüm

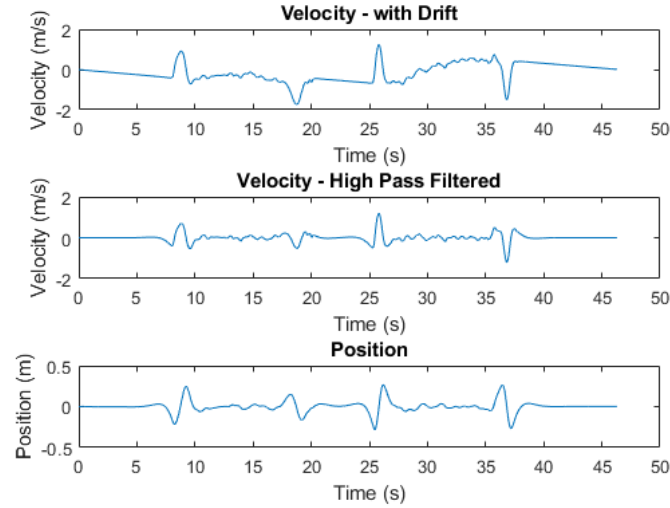
- Kısa Bilgi Notu: Bir nesnenin uzaydaki yönelimini bulmak için 3 temel sensör kullanılır. Bunlar ivmeölçer, gyro ve magnetredir. Bu üç sensörün bir arada kullanılmaktadır. İvme ölçer ile Pitch ve Roll açısı belirlenebilir, ancak hareket sırasında oluşan gürültü hassas bilgi edinilmesini zorlaştırır. Gyroscope ise 3 eksenindeki açısal hızları ölçer. Eğer başkangıçdaki açı değeri bilinmiyorsa gyroscope ile anlık açı hesaplanamaz, sadece açısal hız üzerinden açı değişimi hesaplanabilir. Gyroscope un diğer sorunuda “Drift” özelliğidir. Gyroscope sabit duran bir cisimde bile küçük de olsa açısal hızlar gösterir. Bu nedenle sadece Gyroscope bağlı sistemlerde hesaplanan açılar yavaş yavaş değişir. Bunu engellemek için bazı filtreler kullanılır. Bu filtrelerin amacı Gyroscope üzerinden verileri alıp ivme ölçer üzerinden bu verileri düzeltmektir. Böylece drift engellenmiş olur ve sadece ivmeölçere bağlı olunmadığı için hareket halindeki gürültü sonuçlardan kurtulmuş olur.
- Hafta 13 den IMU ve VICON veri setini indiriniz.  
[http://yunus.hacettepe.edu.tr/~saritan/bca607/hafta\\_13.zip](http://yunus.hacettepe.edu.tr/~saritan/bca607/hafta_13.zip)
- a) Zip dosya içerisinde bulunan IMU (*phoneIMU.mat*) veri setini kullanarak bileşke ivmeyi hesaplayınız, bileşke ivmeden yer çekim ivmesini çıkarınız ve hareketten oluşan ivmeyi filtreleyiniz. Grafik olarak gösteriniz.



- b) Filtrelenmiş ivme değerini kullanarak hızı ve yerdeğiřtirmeyi hesaplayınız ve grafik olarak ciziniz. Hareketsiz durumda oluřan hız deęiřimine **dikkat**.



- c) Hızı deęerini filtreleyerek kaymayı ortadan kaldırınız ve yerdeęiřtirmeyi tekrar hesaplayınız.



Hesaplamalarınızı ve grafiklerinizi raporlandırarak **PDF** olarak

[serdar.aritan@gmail.com](mailto:serdar.aritan@gmail.com) ve [serdar.aritan@hacettepe.edu.tr](mailto:serdar.aritan@hacettepe.edu.tr) adreslerine **BCA607**

**Final Projesi** konu bařlıęıyla **22 Ocak 2020 arřamba** gnne kadar gnderiniz.

Bařarılar dilerim.

**İpucu: Unutmayın bu sorunla ilk siz karşılaşmıyorsunuzdur!!! Araştırın....**

[https://www.researchgate.net/post/How\\_can\\_I\\_avoid\\_data\\_drifting\\_when\\_integrating\\_acceleration\\_signal2](https://www.researchgate.net/post/How_can_I_avoid_data_drifting_when_integrating_acceleration_signal2)



**Mahesh J Rao**

Indian Institute of Technology Madras

3 years ago

I faced the same problem.

Drift in your velocity and displacement or position comes from the numerical integration, which is an approximation, even when you use better techniques like Simpson's 1-4-1 integration rule instead of trapezoidal rule.

Noise are high frequency signals where as drifts are low or very low frequency signals.

If drift is not desirable use high pass filter for your result (velocity and displacement) with appropriate cutoff frequencies to avoid it

If noise is also present in your result along with drift, use band-pass filter with appropriate pass and stop frequencies.

Instead of using band-pass filter, using low pass filter for measurement (acceleration) to remove noise and then high pass filter for integrated signals (velocity and position) yields better integrated signals.

10 Recommendations



**Robert Owsiński**

Opole University of Technology

2 years ago

This is a part of my function in Matlab.

I hope, that it will be helpfull for someone.

```
[b,a]=butter(2,f/Fs,'low');
```

```
[d,c]=butter(2,f/Fs,'high');
```

```
a=filter(b,a,signal);
```

```
v=cumtrapz(t,a);
```

```
v=filter(d,c,v);
```

```
s=cumtrapz(t,v);
```

where: