

ELM368 – DÖNEM PROJESİ

IMU SENSÖRÜ İLE İNSAN AKTİVİTESİ TESPİTİ

Fouad Aladhami, Ömer Güney, Esra Kirman

210102002808, 200102002070, 200102002028

f.aladhami2021@gtu.edu.tr, o.guney2020@gtu.edu.tr, e.kirman2020@gtu.edu.tr

ABSTRACT (ÖZET)

Aktivite (faaliyet) sınıflandırması sağlıksal durumun takibi gibi pek çok farklı alanlarda kullanılmaktadır. Bu belgede ataletsel ölçü birimi (Inertial measurement unit – IMU) kullanılarak elde edilen ivme ve jiroskop verileriyle insanın yürüme, koşma ve merdiven çıkma aktivitesinden hangisini yaptığının tespitini sağlayan veri toplama ve sinyal işleme çalışmaları yer almaktadır.

ANAHTAR KELİMELELER

FIR Filtre, insan hareketi tespiti, Inertial measurement unit, sinyal işleme

1. Giriş

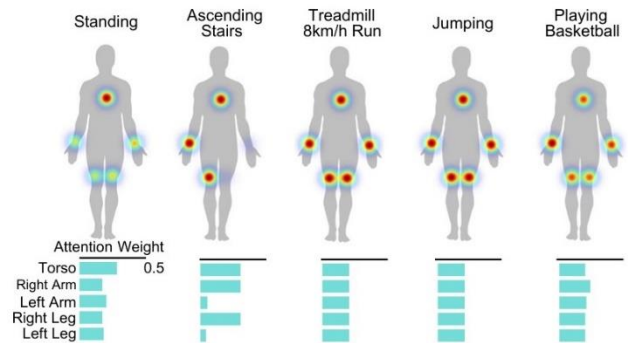
IMU (Inertial measurement unit) sensörleri giyilebilir teknolojilerde ve sağlık izleme sistemlerinde yaygın olarak kullanılan sensörlerdir. Bu sensörler, ivme ve açısal hız gibi verileri sağlayarak insan hareketlerinin izlenmesine ve analiz edilmesine olanak sağlar.

Geleneksel aktivite tanıma sistemleri tipik olarak bir dizi adımdan oluşur. Bu adımlar veri toplama, ön işleme, özellik çıkarma ve sınıflandırmayı içerir. Makalelerde bahsedilen ARC (Activity Recognition Chain) gibi yöntemler bu adımları içeren geleneksel yaklaşımlardan biridir. Ancak derin öğrenme teknikleri, özellikle sinir ağı tabanlı yöntemler, son yıllarda aktivite tanıma konusunda önemli ilerlemeler kaydetmiştir. İnsan Aktivitesi Tanıma (HAR – Human Activity Recognition) konusu içerisinde, yürüme, koşma, merdiven çıkma/inme, spor yapma, yemek pişirme ve düşme gibi çeşitli aktivitenin tespiti üzerine çalışılmaktadır. Bu amaçla sınıflandırmada yüksek doğruluk elde etmek için makine öğrenmesi ve örüntü tanıma teknikleri kullanılmıştır.

Tartışmasız, tüm denetimli HAR öğrenme yöntemleri için en büyük zorluk ve kısıtlayıcı faktör, genellikle sınırlı miktarda etiketli eğitim verisinin olmasıdır. Makine öğreniminin diğer uygulama alanlarının aksine, burada büyük miktarlarda doğru etiketlenmiş veri örnekleri toplamak zordur. Benzer şekilde, bir kişiden günlük aktiviteleri için doğrudan gerçek değer etiketlemesini sağlaması pratik değildir. [1]

İnsan Hareketi Tanımlama işlemini sadece IMU verilerinde sınırlı kalmayıp GPS verileriyle beraber yapanlar bulunmaktadır [2]. Derin öğrenme işlemleriyle ilerleyenler de sayıca fazladır. Convolutional Neural Networks (CNNs)[3], Adaptive Hierarchical Decision Tree (AHDT)[4], Feature Fusion Enrichment Multi-Layer Perceptron (GAM-MLP)[5] gibi derin öğrenme yöntemleri araştırmacılar tarafından kullanılmıştır.

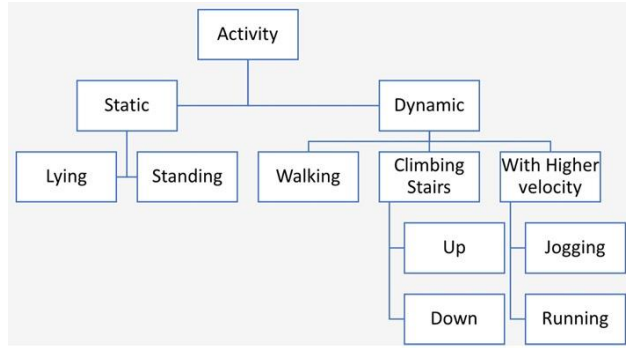
Bu proje içerisinde IMU sensöründen elde edilen ivme ve gyro verileri ile yürüme, koşma ve merdiven çıkma aktivitesi üzerine HAR çalışması yapılmaktadır. Yürüme, koşma ve merdiven çıkma aktivitelerin tespit edilebilmesi için IMU sensöründen gelen verilerin Fourier Dönüşümleri alınarak frekans alanında incelenmektedir. Yürüme hareketinde frekansların düşük olacağı koşma hareketinde daha yüksek frekanslarda hareketlerin gözükceği tahmin edilebilmektedir. Bu deneyde IMU sensör verileri bilekten alınmıştır. Farklı hareketler için IMU verilerinin farklı oranda önemleri vardır. İnsanlar her hareketi yaparken vücutlarının farklı bölgelerini ağırlıkta kullanarak çalıştırır. Birkaç hareketi birbirinden ayırmak için etkili yöntem birkaç bölgeden alınan verilerin incelenmesidir. Bilekten alınan veriler koşma ve yürüme ayırt etmek için yeterliken yürüme ve merdiven çıkma hareketlerindeki kol hareketleri benzer olduğundan bu hareketlerin ayrımı için ayaktan da veri almak daha doğru sonuçlar verecektir. Hareketler özelinde bölgelerin kullanımı Şekil 1’de gösterilmiştir.



Şekil 1: Hareketlere göre IMU verilerinin etken olduğu bölgeler [3].

IMU tabanlı HAR sistemleri, diğer algılama cihazlarına göre daha az maliyetli olma avantajına sahiptir. Ayrıca, bu sistemler kullanıcıların mahremiyetini ihlal etmez ve çevreleriyle etkileşime girmez. Çeşitli insan aktiviteleri benzer sensör desenleri sergilediğinden, geleneksel düz sınıflandırma yöntemleri kullanılarak bunları ayırt etmek zor olabilir. Düz sınıflandırma, tüm kategorilerde aynı anda ve aynı seviyede eğitilen bir sınıflandırıcıyı ifade eder. Sensör desenlerinin karmaşık doğası, bu tür yöntemler için önemli bir zorluk oluşturur ve farklı türdeki insan aktivitelerini daha az karmaşık bir şekilde doğru bir şekilde sınıflandırma ihtiyacını artırır [4].

İnsan aktiviteleri üç ana kategoriye ayrılabilir: statik, düşük dinamik ve dinamik aktiviteler. Statik aktivite, çevrede sabit kalmak gibi minimal hareket veya fiziksel çaba gerektirir. Buna karşılık, düşük dinamik bir aktivite, minimal fiziksel hareket içerir ancak tamamen hareketsiz değildir. Dinamik aktivite ise sürekli değişen bir ortamda hareket içerir. Aktivitelerin kategorilere göre dağılımı Şekil 2’de gösterilmiştir.



Şekil 2: Hareketlerin sınıflandırılması [4].

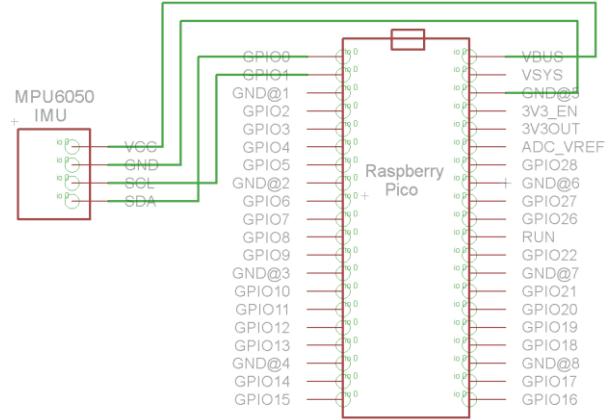
Bu çalışmada dinamik hareketlerden üç farklı durum ayırt edilmeye çalışılmıştır. Şekil 2’deki sınıflandırmada görebiliriz ki bu üç durum sınıflandırma yöntemiyle sınıflandırılacak aktivitelerdir. Ancak dinamik aktivitelere sahip olduğundan aralarındaki farkların daha iyi tespiti için Şekil 1’de de ele alındığı gibi verinin birkaç bölgeden veya seçilen hareketlerdeki farklılıkları açıkça ortaya koyabilen bir noktadan alınması gerekmektedir.

2. Deneyler ve Analiz

2.1. Veri Toplama

IMU verilerinin toplanabilmesi için IMU verilerini okuyacak ve bunu bilgisayar ortamına yazacak tasarım Şekil 3’te verilen devre şemasındaki gibi gerçekleştirildi. IMU olarak piyasada yaygın olarak kullanılan ve maliyeti düşük olan MPU6050, işlemci olarak küçük boyutu ve kolay programlanabilir olması sebebiyle Raspberry Pico W tercih edilmiştir. Tasarlanan ilk senaryoda IMU’dan alınan veriler SD kart modülü aracılığıyla SD kart üzerine yazılacaktı. Böylece sensör deney verileri alınacak kişinin bileğine veya ilgili başka bir bölgesine (göğüs, bacak vb.)

takılabilecek ve bilgisayar bağlantısı olmadan veri toplanabilecekti. Ancak SD karta en hızlı 30 ms aralıklarla veri yazılabildiğinden daha hızlı olması açısından 5ms hızla UART kullanılmıştır.



Şekil 3: IMU Devresi Şeması.

IMU’da veri toplama için kullanılan kod aşağıda verildiği gibidir.

```

import machine
import utime
import sys
import uos
import time
# Configure I2C
i2c = machine.I2C(0, freq=400000, sda=machine.Pin(0),
scl=machine.Pin(1))
#uart = machine.UART(0, baudrate=115200) #
Assuming UART0, adjust the UART number as needed
#uos.dupterm(uart)
#uart.init(baudrate=115200) # Set baud rate to 115200
bps or adjust to your desired value

class MPU6050():
    # MPU6050 I2C address
    MPU6050_ADDR = 0x68

    # Register addresses
    MPU6050_REG_PWR_MGMT_1 = 0x6B
    MPU6050_REG_SMPLRT_DIV = 0x19
    MPU6050_REG_ACCEL_CONFIG = 0x1C
    MPU6050_REG_GYRO_CONFIG = 0x1B
    MPU6050_REG_ACCEL_XOUT_H = 0x3B
    def __init__(self, i2c):
        self.i2c = i2c
        self.addr = MPU6050.MPU6050_ADDR
        self.accel_x = 0
        self.accel_y = 0
        self.accel_z = 0
        self.gyro_x = 0
        self.gyro_y = 0
        self.gyro_z = 0
        # Initialize MPU6050
  
```

```

self.i2c.writeto_mem(MPU6050.MPU6050_ADDR,
MPU6050.MPU6050_REG_PWR_MGMT_1, b'\x00') #
Wake up MPU6050

self.i2c.writeto_mem(MPU6050.MPU6050_ADDR,
MPU6050.MPU6050_REG_SMPLRT_DIV, b'\x00') #
Sample rate divider (1kHz)

self.i2c.writeto_mem(MPU6050.MPU6050_ADDR,
MPU6050.MPU6050_REG_ACCEL_CONFIG, b'\x00') #
Accelerometer scale (+/- 2g)

self.i2c.writeto_mem(MPU6050.MPU6050_ADDR,
MPU6050.MPU6050_REG_GYRO_CONFIG, b'\x00') #
Gyroscope scale (+/- 250deg/s)

def read_sensor_data(self):
    data =
self.i2c.readfrom_mem(MPU6050.MPU6050_ADDR,
MPU6050.MPU6050_REG_ACCEL_XOUT_H, 14)
    self.accel_x = (data[0] << 8) | data[1]
    self.accel_y = (data[2] << 8) | data[3]
    self.accel_z = (data[4] << 8) | data[5]
    self.gyro_x = (data[8] << 8) | data[9]
    self.gyro_y = (data[10] << 8) | data[11]
    self.gyro_z = (data[12] << 8) | data[13]

MPU6050_Sensor = MPU6050(i2c)

# Assign chip select (CS) pin (and start it high)
cs = machine.Pin(5, machine.Pin.OUT)
# Intialize SPI peripheral (start with 1 MHz)
spi = machine.SPI(0,
    baudrate=1000000,
    polarity=0,
    phase=0,
    bits=8,
    firstbit=machine.SPI.MSB,
    sck=machine.Pin(2),
    mosi=machine.Pin(3),
    miso=machine.Pin(4))

# Open the file we just created and read from it
#with open("/sd/test01.txt", "r") as file:
#    data = file.read()
#    print(data)

def timer1_callback(timer):
    #start_time = time.ticks_us()
    MPU6050_Sensor.read_sensor_data()
    timestmap = time.ticks_ms()
    print(str(timestmap)+ '\t'+ 'ax',
MPU6050_Sensor.accel_x, 'ay',
MPU6050_Sensor.accel_y, 'az',
MPU6050_Sensor.accel_z, 'gx',

```

```

MPU6050_Sensor.gyro_x, 'gy',
MPU6050_Sensor.gyro_y, 'gz',
MPU6050_Sensor.gyro_z)

    #try:
        #file.write(str(MPU6050_Sensor.accel_x) + '\t' +
str(MPU6050_Sensor.accel_y) + '\t' +
str(MPU6050_Sensor.accel_z) + '\t' +
str(MPU6050_Sensor.gyro_x) + '\t' +
str(MPU6050_Sensor.gyro_y) + '\t' +
str(MPU6050_Sensor.gyro_z) + '\n')
    #except:
        #print("File not found")
    #measure time
    #stop_time = time.ticks_us()
    #calculate time
    #delta_time = time.ticks_diff(stop_time, start_time)
    #display time in seconds
    #print("Time: ", delta_time/1000, "ms\n")

#def timer2_callback(timer):
#    print('ax', MPU6050_Sensor.accel_x, 'ay',
MPU6050_Sensor.accel_y, 'az',
MPU6050_Sensor.accel_z, 'gx',
MPU6050_Sensor.gyro_x, 'gy',
MPU6050_Sensor.gyro_y, 'gz',
MPU6050_Sensor.gyro_z)

# Set PLL to maximum frequency (250 MHz)
machine.freq(250000000)

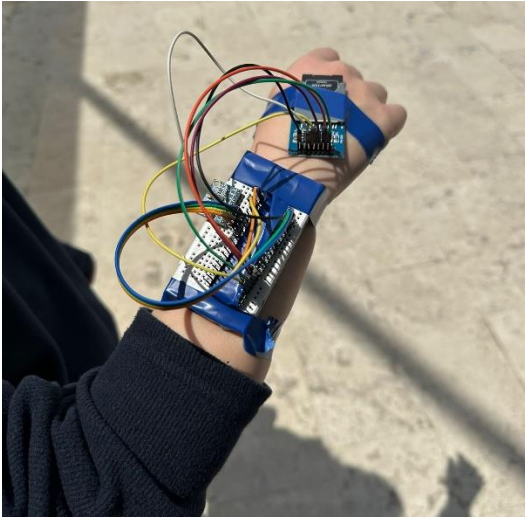
# Print current CPU frequency
cpu_frequency = machine.freq()
print("CPU Frequency:", cpu_frequency, "Hz")
timer1 = machine.Timer()
timer1.init(period=5, mode=machine.Timer.PERIODIC,
callback=timer1_callback)

#timer2 = machine.Timer()
#timer2.init(period=10,
mode=machine.Timer.PERIODIC,
callback=timer2_callback)
# Main loop

while True:
    utime.sleep(1)

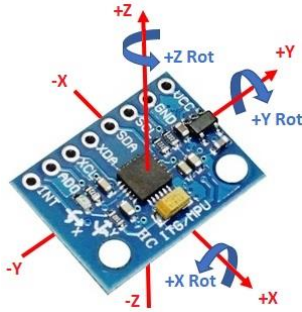
```

Başta gerekli kütüphaneler eklendi ve IMU'nun işlemciyle haberleşeceği I2C protokolünün ve bilgisayarla seri haberleşmenin yapılacağı UART protokolünün konfigürasyonları yapılmıştır. MPU6050 class yapısında register adresleri, data işlemleri tanımlanmıştır. 200 Hz frekansla örnekleme yaparak veriler toplanmıştır.



Şekil 4: Devrenin bilekte konumlandırılması.

Devre bileğe Şekil 4'te görüldüğü gibi bağlanmıştır. Böylelikle hareket esnasındaki x-y-z ivme ve gyro verileri toplanabilmektedir. Hareket esnasında kol ileri geri hareket ettiğinden Şekil 5'te gösterilen MPU6050 eksen bilgisine dikkat edilirse x eksenindeki verilerde daha çok değişim olması beklenmektedir.



Şekil 5: MPU6050 Eksenleri [6].

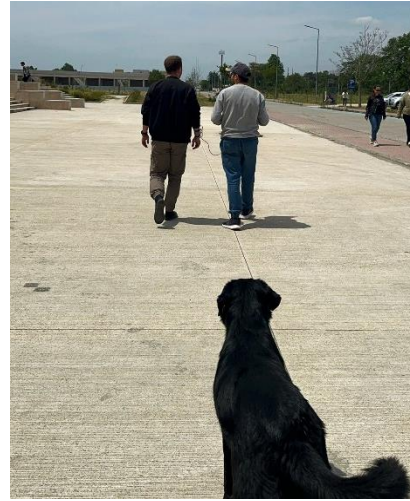
Yürüyerek işleme için kullanılacak verilerin toplandığı anlar Şekil 6,7 ve 8'de verilmiştir.



Şekil 6: Yürüme verisi toplama – 1.



Şekil 7: Yürüme verisi toplama – 2.



Şekil 8: Yürüme verisi toplama – 3.

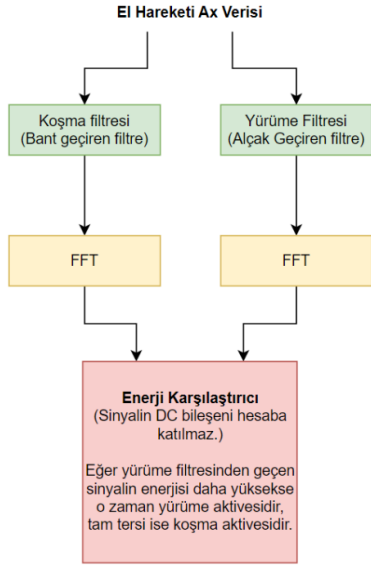
Yürüme esnasındaki ivme koşmadan daha düşük olacağı için koşma esnasında daha yüksek frekanslı hareketler beklenmekteydi. Bu iki hareketi birbirinden ayırırken de bu frekans farkından yararlanıp bant geçiren ve alçak geçiren filtrelerle sinyallerin ayrıştırılabileceği düşünüldü.

2.2. Veri Analizi

Alınan verileri analiz etmek için gelen değerler okunarak ax, ay, az, gx, gy, gz değişkenlerine atanmış ve integer 16 bite çevrilmiştir.

Toplanan verilerin analizi için Şekil 9'da verilen temel algoritma şeması oluşturulmuştur. Algoritma akış şemasından da anlaşılabileceği üzere veriler türüne göre filtrelerden geçirerek baskın oldukları aralıklarda alınıyorlar. Eğer o filtreye sokulan sinyal yürüme veya koşma değil ise sinyalinin ağırlıkta olduğu yerler kesilerek sinyalin enerjisi yok ediliyor ve bu da bir sonraki aşamaya hazırlık oluyor. Sonrasında Fast Fourier Transform (FFT)'ları alınarak enerjilerinin karşılaştırılacağı Enerji Karşılaştırmalı filtreye giriyorlar. Burada, eğer yürüme

filtresinden geçen sinyalin enerjisi karşılaştırma sonucunda daha yüksek geliyorsa o zaman bu sinyalin yürüme sinyali olduğu anlaşıyor. Çünkü yürüme filtresinden gücünün yoğunlukta olduğu kısımların geçmesiyle enerjisi yeterli geliyor. Tam tersi durumda da bu sinyal koşma filtresi olarak kararlaştırılıyor.



Şekil 9: Sinyal İşleme Algoritma Şeması

Birinci yürüme ve koşma verilerindeki sinyallerin zaman eksenindeki çıktıları Şekil 10'daki gibi gelmiştir. Verilerde görülebildiği üzere sinyaller periyodik olarak ilerlemektedir. Koşma verisinin daha sık olduğu görülmektedir. Sensör ivme ölçeri varsayılan ayarlarda kullanıldığından, $\pm 2G$ ivmesine ayarlıdır. Bu yüzden maksimum ivme $2G$ ile, minimum ivme ise $-2G$ ile limitli olmuştur. Bu değerler yer çekim ivmesinden çıkarılmamış halidir. Bu G sınırlamasından dolayı koşarken veya merdiven çıkarken oluşan güçlü ivmelenme durumunun gösterilememesine neden olmuş olabilir.

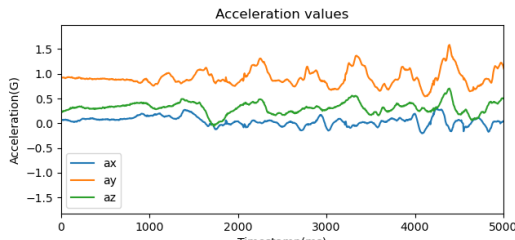
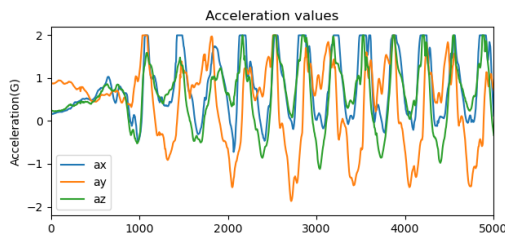


Figure 2



Şekil 10: Zamanda yürüme ve koşma verileri.

Gyro verileri de Şekil 11'deki gibi gelmiştir. Ancak değerlendirme aşamasında gyro verileri dikkate alınmayacaktır. İvme verileriyle değerlendirmeler sağlanacaktır.

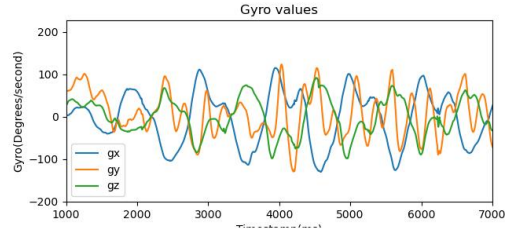
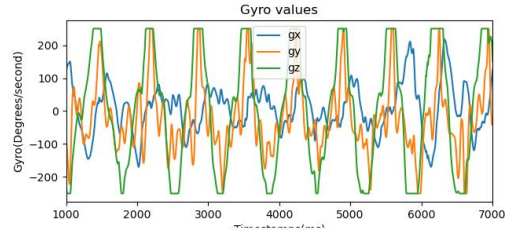


Figure 4



Şekil 11: Yürüme ve koşma gyro verileri.

Aslında alınan veriler 200 Hz ile örneklenmiş dijital işaretlerdir. Bu nedenle çizdirirken Şekil 12'deki gibi ayrıık olarak da çizdirilmiştir.

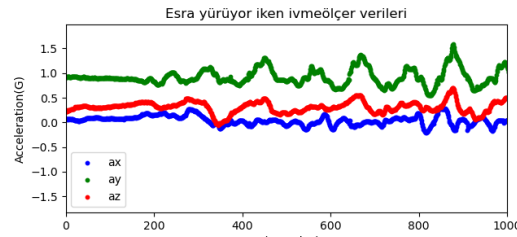
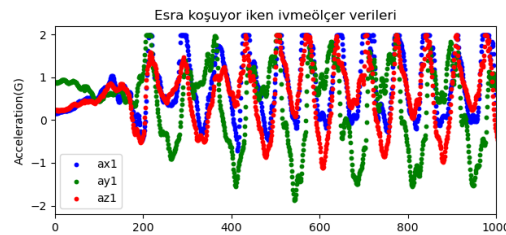
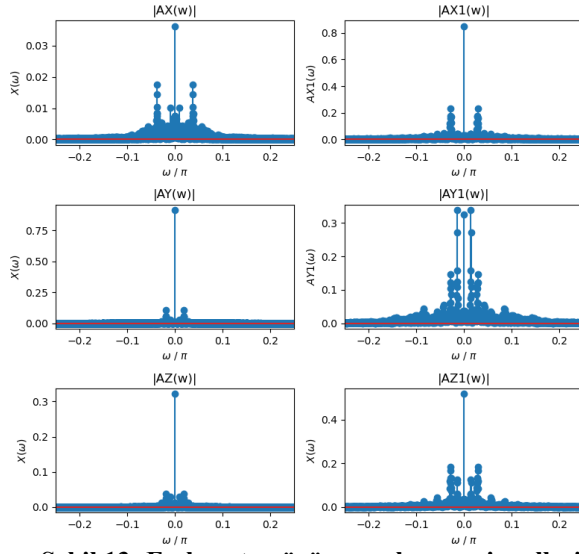


Figure 6



Şekil 12: Ayrıık zamanda yürüme ve koşma verileri.

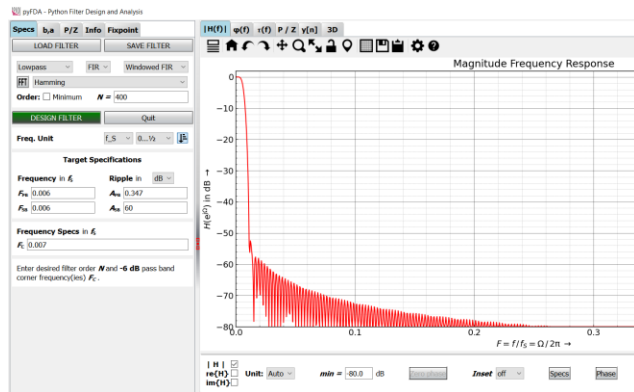
Haraketleri ayırt edebilmek için fourierleri alınmıştır ve Şekil 13'te gözükten x y z eksenlerindeki ivme verilerinin frekanstaki karşılıkları elde edilmiştir. Buradan incelemeler yapılmıştır. Görüldüğü üzere x eksenindeki ivmelerde yürüme sinyali yaklaşık olarak 0.015π frekansa kadar gelirken koşma sinyali yaklaşık olarak 0.010π - 0.032π arasında yoğunlaşmıştır. Bunları ayırt etmek için bant geçiren ve alçak geçiren filtre tasarlanabilmektedir.



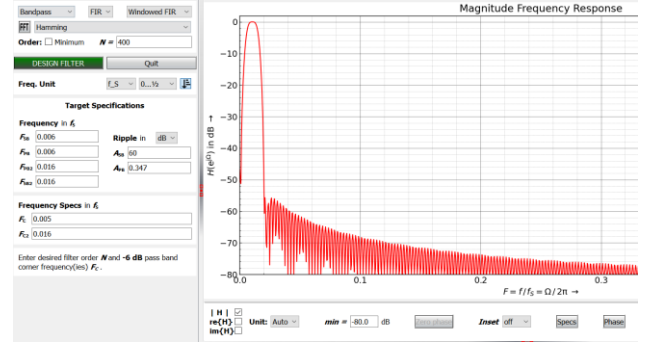
Şekil 13: Frekansta yürüme ve koşma sinyalleri.

Sinyallerde 0 frekansında gelen DC bileşen satüre olan ve durgun olan noktalardan gelmektedir. Yani DC bileşen durgunluğu ifade etmektedir ve sinyalin yapısını bozabilecek kadar güçlüdür. Bu nedenle bu durgunluğun etkisini çıkararak incelemeler yapılmıştır.

Filtreler tasarlanırken pyfda arayüzü kullanılmıştır. Burada frekanslar f cinsinden girildiğinden dolayı $w = 2\pi f$ denkleminde faydalanarak f değerlerinin; yürümeyi filtrelemek için alçak geçiren filtre frekansı olarak yaklaşık $0.006-0.007\pi$, koşmayı filtrelemek için bant geçiren filtre frekans aralığı olarak yaklaşık $0.005\pi-0.0016\pi$ aralığında belirlenmiştir. Filtre penceresi olarak yaygın kullanılan Hamming pencere tercih edilmiştir. Hamming penceresi, nispeten dar ana lob genişliği ve ilk birkaç yan lobun iyi zayıflatılması nedeniyle birçok kişi tarafından tercih edilmektedir. Frekans zirvesine yakın "sızıntının" ilk Sinc yan lobunun boyutunu azaltmaktadır. Yürüme hareketi için tasarlanan alçak geçiren filtre Şekil 14'te, koşma hareketi için tasarlanan bant geçiren filtre Şekil 15'te verilmiştir.

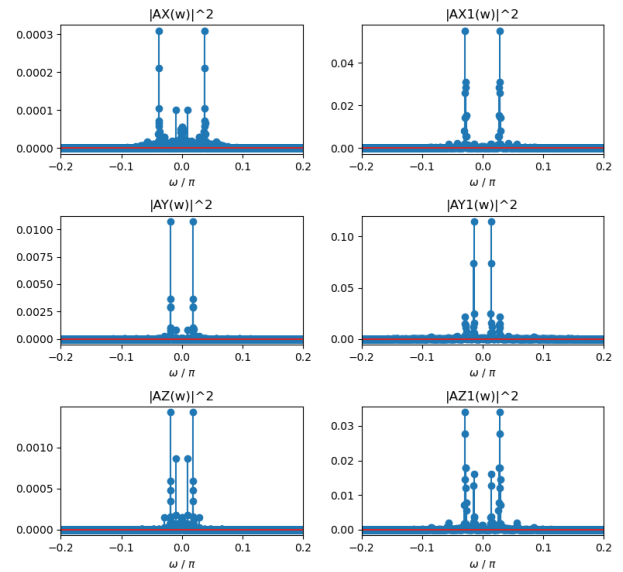


Şekil 14: Alçak Geçiren Filtre.



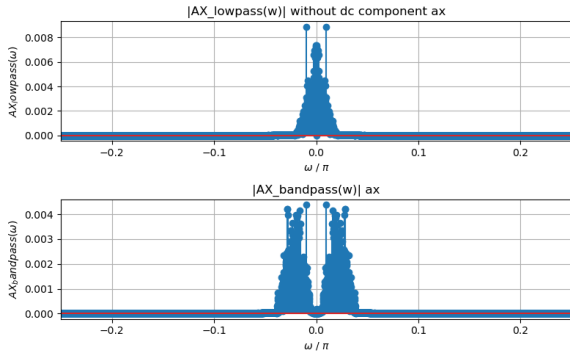
Şekil 15: Bant Geçiren Filtre.

Enerji yoğunluk fonksiyonu gözlemlerin yapılabilmesi için hesaplanmış ve çizdirilmiştir. Çıktılar Şekil 16'da verilmiştir.

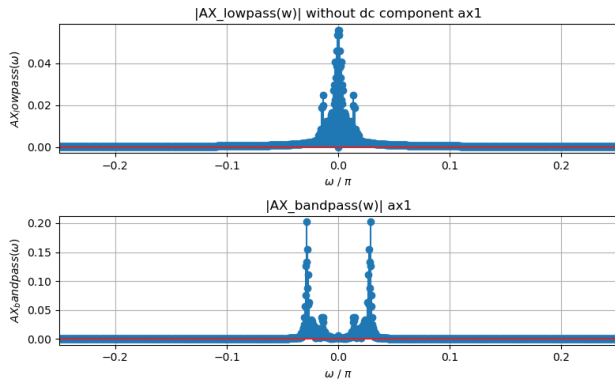


Şekil 16: Enerji Yoğunluk Fonksiyonları

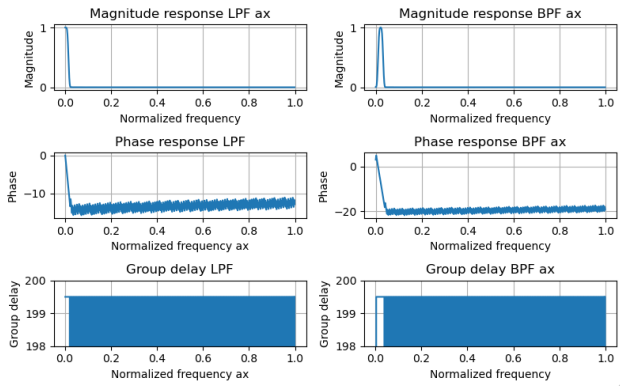
Alçak geçiren ve bant geçiren filtreler yürüme ve koşma için Şekil 17, 18 ve 19'da görüldüğü gibi çizdirilmiştir. Hem zaman hem frekans alanındaki çıktıları ve grup gecikmeleri incelenmiştir. Filtreler tasarlanırken FIR filtre tercih edilmiştir. FIR filtreler kararlılardır ve doğrusal faza sahip olabilmektedirler. Bu da tüm frekanslar için aynı gecikme miktarını sinyale yansıtır ve sinyalin çıkışta sadece faz kaymalı bir şekilde gelmesine, sinyalin istenilen gibi elde edilmesine olanak tanır.



Şekil 17: Frekans alanında filtreler – yürüme.



Şekil 18: Frekans alanında filtreler – koşma.



Şekil 19: Yürüme ve Koşma için filtreler.

STFT çıktısında görüldüğü üzere x eksenindeki veriler biraz gürültülüdür ama yine de beklenildiği gibi yürüme hareketi genlik-frekans grefiğinde görüldüğü gibidir. Aynı şekilde Şekil 21’de verilen koşma STFT çıktıları da tutarlıdır. Burada daha yüksek frekansta yüksek genlik görülmektedir.

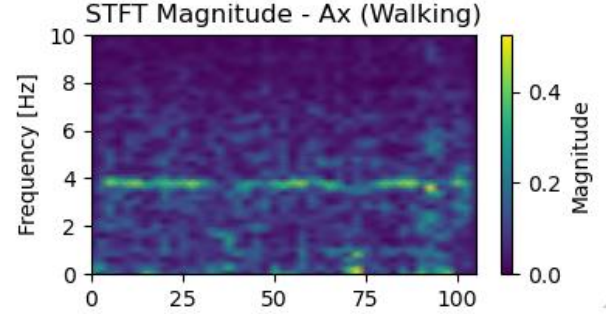


Figure 17

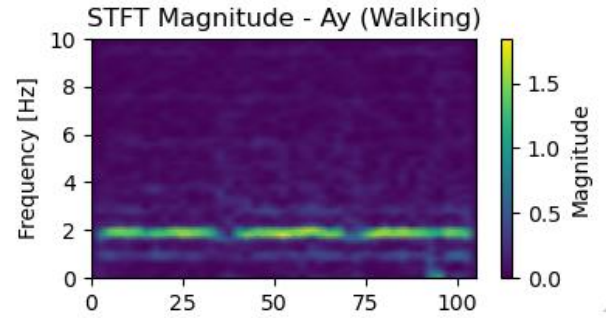
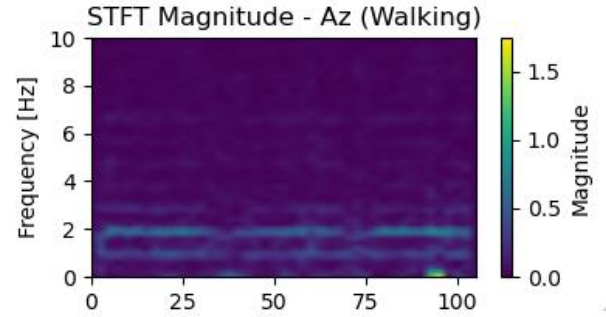


Figure 18



Şekil 20: Yürüme STFT çıktısı.

Bu verilerin STFT (Short-Time Fourier Transform)’leri çizdirilerek frekans çıktılarının zaman eksenine göre durumu incelenmiştir. Çıktılarda önceki sonuçları destekler nitelikte frekans hareketlilikleri görülmektedir. STFT çıktıları Şekil 20 ve 21’de verilmiştir. Burada verilen görseller bir kişiden alınan veriler üzerinedir. Ancak kod kısmında tüm veriler çizdirilmiştir. Şekil 20’de yürüme

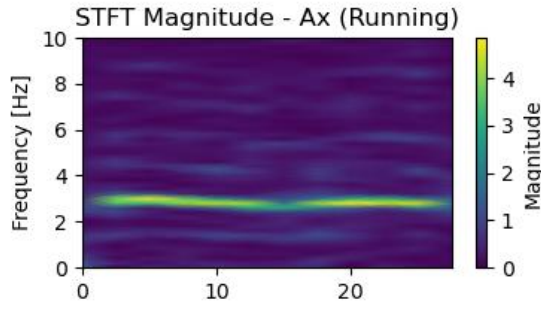
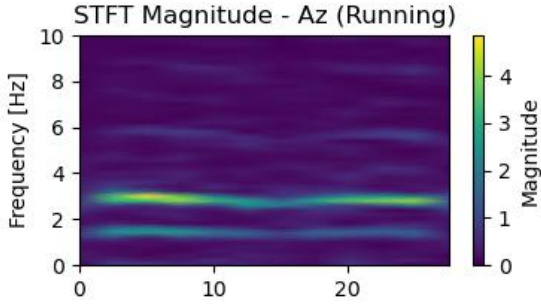
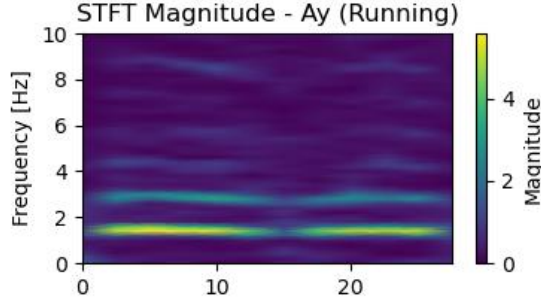
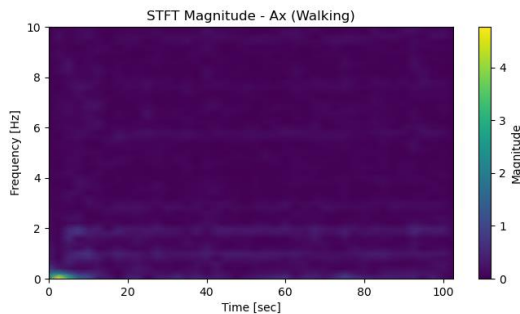


Figure 20



Şekil 21: Koşma STFT çıktısı.

İkinci kişiden alınan verilerde yürümedeki durgunluk daha net bir şekilde görülmektedir. Bu kişiler arasındaki yürüme hızı ve tarzlarındaki farklılıktan kaynaklanmaktadır. Kıyasın yapılabilmesi için ikinci veri yürüme STFT çıktısı Şekil 22’de verilmiştir.



Şekil 22. İkinci veri yürüme x eksen STFT çıktısı.

Koşma ve yürüme hareketleri başarılı bir şekilde ayırt edilebilmiştir. Ancak merdiven çıkma verisi bilekten alındığı zaman istenen sonucu verememektedir. Çünkü yürürken ve merdiven çıkarken yapılan kol hareketleri benzerlik taşımaktadır, hatta neredeyse aynıdır. Giriş kısmında da bu konu üzerinde durulmuştur. Bu nedenle merdiven çıkma verisinin ayırt edilebilmesi için ayakta da veri ölçümü alınmıştır. Ancak bu ölçümde de alınan veriler pek istenildiği gibi gelmemiştir. IMU sensörü ayağa bağlandığında merdivende her bir basamak çıkıldığında z ekseninde dirac işaretine benzer bir işaret oluşur. Dirac işaretinin fourier transformu birdir ve tüm frekanslarda bileşeni vardır. Bundan dolayı frekanslara yayılım gözükmemektedir. İşaret belli bir noktada toplanmamaktadır. Durum Şekil 23’te gözükmemektedir.

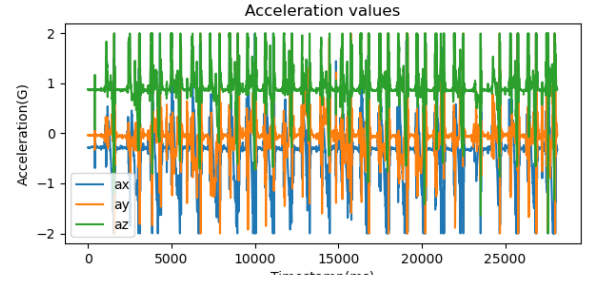
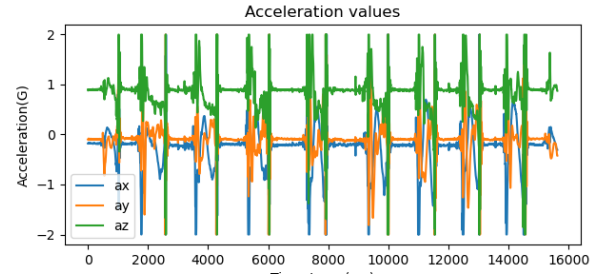


Figure 148



Şekil 23: IMU ayağa bağlı iken zamanda merdiven çıkma hareketi.

STFT’leri alındığında da frekansın genliğinin zamanda ve frekansta yayılım gösterdiği açıkça görülmektedir. STFT çıktıları Şekil 24’te verilmiştir. Tüm kodlar yollanan kod dosyasında yer almaktadır. Uzunluğu sebebiyle burada paylaşılmamıştır.

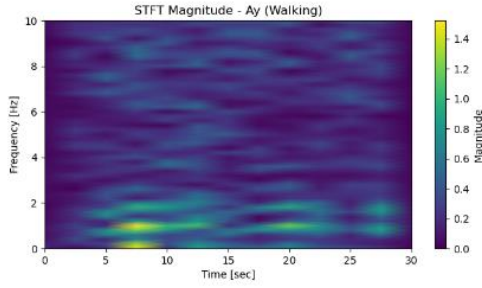


Figure 225

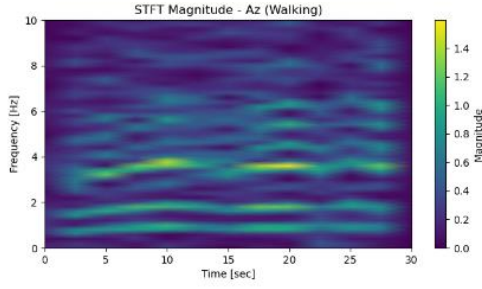
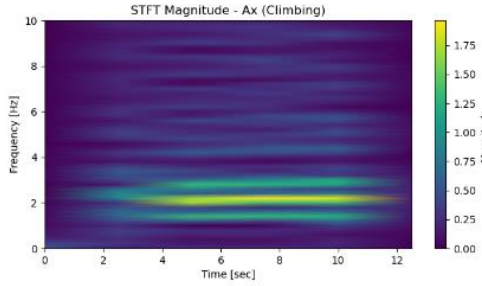


Figure 226



Şekil 24: Merdiven çıkma STFT çıktısı.

Merdiven çıkma eylemi hem bilek hem de ayaktan alınan verilerle zaman ve frekans alanında incelenmiş ancak ayırt edilebilecek bir fark oluşmamıştır. Bu farkı bulabilmek adına enerjisi, varyansı ve ortalama değerine bakılmış ancak buralarda da sonuç elde edilememiştir. Bu konuda makaleler incelendiğinde bu ayırt etme işlemini Destek Vektör Makinesi (Support vector machine (SVM)) ile yapanlar mevcuttur [7]. Bu çalışmada Şekil 25'teki grafik elde edilmiştir ve ayırt etme işlemlerinin SVC ile daha rahat yapılabildiği görülmektedir.

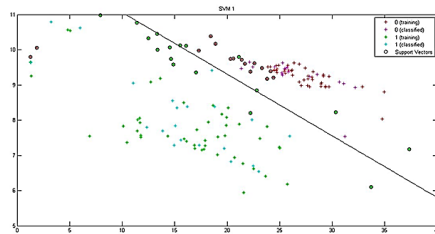


Fig. 4 SVM classification results for walking forward (red and pink) and walking in stairs (cyan and green)

Şekil 25: Yürüme ve Merdiven çıkmanın SVC ile ayırımı.

Koşma ve yürüme ayırt ederken beklenen sonuçlar elde edilmiştir. Ancak merdiven çıkma verilerinde yürümeden farklı olarak yukarı yönlü bir ivmelenme olacağı beklenirken yukarıya düz bir şekilde çıkmadığı aynı zamanda yatayda da ivmelenildiği için beklenildiği gibi sonuç elde edilmemiştir ama elde edilen bu sonuç mantıklıdır. İncelemelerde yapılan uğraşlar sonucunda merdiven çıkma tespiti için daha farklı bölgelerden veriler alınması gerektiği farkedilmiş, pek çok farklı yöntem denenmiştir. Sonuç olarak merdiven çıkma tespiti istenilen gibi yapılamamıştır.

3. Sonuç ve Yorum

Bu çalışma içerisinde toplanan sensör verilerinin istenen herhangi bir durumun (insan hareketi, ses ayrımı vs.) kullanılabileceğini ve bunu frekans alanında inceleme yapıp bu inceleme sonucunda tasarlanabilecek alçak geçiren, bant geçiren veya yüksek geçiren gibi filtrelerle basit uygulamalarla yapabileceğimizi gördük. Kullanılan filtrelerde pencerenin ve filtre türünün önemini uygulamalı olarak öğrendik. İstenen sistem karmaşıklaştıkça ve birbirine yakın verilerle yapıldığında işlerin daha zorlaştığı ve farklı analiz tekniklerinin geliştirilmesi gerektiğini farkettilik.

IMU sensörleri gürültüye karşı hassas olduklarından elde edilen çıktılar bazı noktalarda beklenen daha fazla gürültülü gelmektedir. Bunun için daha kaliteli bir IMU kullanılabilir. Veriler derste öğrendiğimiz yöntemlerle filtrelenemediğinden elimize IMU'dan gelen veriler kullanım için aşırı uygun olmayabilir. Ayrıca farklı bölgelerden alınan verilerin beraber değerlendirilmesi giriş kısmında da bahsedildiği üzere daha olumlu sonuçlar getirebilirdi.

İnsan Hareketi Tanıma (HAR) sistemleri üzerine pek çok farklı çalışma yer almaktadır. Ayrıca insanların sağlık durumlarının tespiti gibi uygulamalarda kullanılabileceği için önem de arz etmektedir. Burada yaptığımız hata ve eksikliklerimizle beraber böyle bir çalışma içerisinde daha derinlemesine yer aldığımızda olaya nasıl yaklaşabileceğimizi öğrenmiş olduk. Elde edilen sinyallerin frekans ve zaman alanlarında nasıl incelenmesi gerektiğini öğrendik. Ayrıca ihtiyaca uygun filtre tasarlama üzerine uğraştık. Uygun filtreyi elde etmek için grafikleri okuduk ve uygun pencereyi seçebilmek için araştırmalar yaptık.

Kaynaklar

- [1] H. Kwon., G. D. Abowd, T. Plötz, Complex Deep Neural Networks from Large Scale Virtual IMU Data for Effective Human Activity Recognition Using Wearables, *Sensors*, 2021, 21, 8337.
- [2] S. C. Navisa, M. N. Cahyadi, T. Asfihani, Analysis of GNSS and IMU Sensor Data Fusion Using the Unscented Kalman Filter Method on Medical Drones in Open Air

- [3] W. Tao, H. Chen, Md Moniruzzaman, Ming C. Leu, Zhaozheng Yin, Ruwen Qin, Attention-Based Sensor Fusion for Human Activity Recognition Using IMU Signals, 20 Dec 2021
- [4] H. Nematallah, S. Rajan, Adaptive Hierarchical Classification for Human Activity Recognition Using Inertial Measurement Unit (IMU) Time-series Data, IEEE /ACCESS.2024.3386351
- [5] Yaxin Mao, Lamei Yan, Hongyu Guo, Yujie Hong, Xiaocheng Huang, Youwei Yuan, AHybrid HumanActivity Recognition Method Using an MLPNeural Network and Euler Angle Extraction Based on IMUSensors
- [6] RoboSync, MPU-6050 3-Axis Accelerometer and Gyroscope Sensor, <https://robosynckits.in/product/mpu-6050-3-axis-accelerometer-and-gyroscope-sensor/>
- [7] R. Alvarez, E. Pulido, D.A. Sierra, Climbing/Descending Stairs Detection Using Inertial Sensors and Implementing PCAandaSVMClassifier, 28 Oct 2016