



Műholdas helyzetmeghatározó rendszerek II.

Intelligens Rendszerek
Gyakorlat



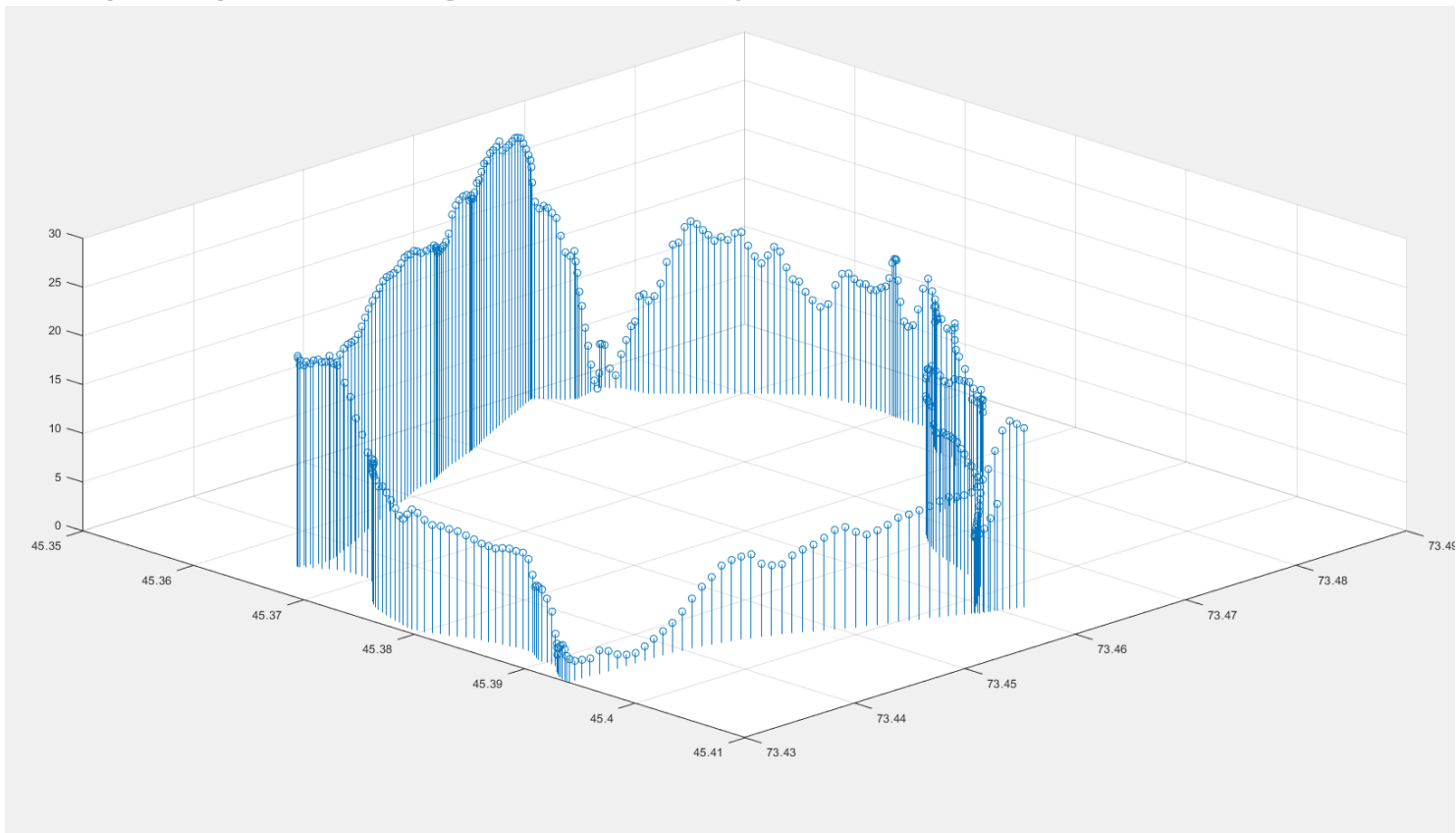
GPS

NMEA log és paramétere



Magassági profil megjelenítése

- Feladat:
 - Adott egy NMEA log
 - Rajzolja ki a gép 3D nyomvonalát!





GPGGA NMEA mondat felépítése

\$GPGGA,542.600,3314.8961,N,11142.2714,W,1,10,.86,395.2,M,-26.6,M,,*52

- 1 = UTC of Position
- 2 = Latitude
- 3 = N or S
- 4 = Longitude
- 5 = E or W
- 6 = GPS quality indicator (0=invalid; 1=GPS fix; 2=Diff. GPS fix)
- 7 = Number of satellites in use [not those in view]
- 8 = Horizontal dilution of position
- 9 = Antenna altitude above/below mean sea level (geoid)
- 10 = Meters (Antenna height unit)
- 11 = Geoidal separation (Diff. between WGS-84 earth ellipsoid and mean sea level. -=geoid is below WGS-84 ellipsoid)
- 12 = Meters (Units of geoidal separation)
- 13 = Age in seconds since last update from diff. reference station
- 14 = Diff. reference station ID#
- 15 = Checksum



Checksum számítása

\$GPGGA,542.600,3314.8961,N,11142.2714,W,1,10,.86,395.2,M,-26.6,M,,*52

- Checksum: 0x52 (maximális értéke 0xFF)

Ciklus (i=2-től mondathossza-3-ig)

{

chk=bitenkénti_xor(chk, mondat(i));

}

Ciklus vége

//vesszük a karakterek decimális kódjainak a xor kapcsolatát!

G xor **P** xor **G** xor **G** xor **A** xor , xor **5** xor **4** xor **2** xor . xor **6** xor **0** xor **0** xor ,
xor xor **M** xor , xor ,



Matlab parancsok

- help parancsnev – gugli mellett az egyik legjobb barátunk
- clc – képernyő törlése
- clear all – összes változó törlése
- cell_tipusu_tomb = importdata('filename.txt');
- string = char(cell_tipusu_tomb(i)); //cell típusból string
- strcmp(a,b) //két string összehasonlítása, visszatérési érték 0 vagy 1
- dec_tipusu_valtozo = hex2dec(hex_valtozo);
- dec_valtozo = unicode2native(character)//karakter-ből szám
- bitszintu_xor_eredmeny = bitxor(a,b); //a és b változó bitszintű xor kapcsolata
- double_valtozo = str2double(string_valtozo); //stringből double típusú változó
- Plot(x), Plot(x,y), Plot(x,y,z) //grafikon megjelenítés



%Hasonlóan az előző órai anyaghoz:

```
format long
```

```
clc
```

```
clear all
```

```
%GPS log betöltése
```

```
filename='gps_2.txt'; %filenév
```

```
gps = importdata(filename); %betöltés után cell típusú tömböt  
kapunk!
```

```
                                %bármilyen típusú adatot  
tartalmazhat. Számoláskor kasztolni kell!
```

```
delimiter=','; %elválasztó
```

```
s=0; %segéd változó
```

```
idx=1; %altitude tömb indexe
```

```
altitude=0; %magassági adatokat tartalmazó tömb
```

```
lat=0;
```

```
lon=0;
```

```
chk=0; %Checksum
```

```
nmea_type='$GPGGA'; %keresett NMEA mondat típusa
```

```
calculated_chk=0; %számolt checksum
```



```
for i=1:length(gps)
    s=char(gps(i));
    splitted=strsplit(s,delimiter);%egy GPS mondat feldarabolása az
    elválasztó karakter mentén
    if strcmp(splitted(1),nmea_type)==1 %keresett mondat ellenőrzése
        chk=char(splitted(length(splitted)));%Checksum kinyerése a mondat
        végéről
        chk=chk(length(chk)-1:length(chk)); %chk max FF, a * levágása
        chk=hex2dec(chk); %hexa szám a checksum, így át kell
        konvertálni decimálisba

        %Checksum ellenőrzése, a checksum előállítása a $ után
        karaktertől a
        %a * előtti karakterig tart, XOR kapcsolat (bitszintű)
        calculated_chk=0;
        for j=2:length(s)-3
            num = unicode2native(s(j)); %a karakterből számot számot
            kell kreálni
            calculated_chk=bitxor(calculated_chk,num); %XOR művelet
        end
    end
end
```




```
%az eredeti és a számolt checksum ellenőrzése
```

```
if(chk==calculated_chk)
```

```
    %ha a mondat helyes, akkor a 10.  
    értéket (ami a magasság)
```

```
    %eltároljuk a GPGGA tömbbe
```

```
    altitude(idx)=str2double(splitted(10));
```

```
    lat(idx)=str2double(splitted(3));
```

```
    lon(idx)=str2double(splitted(5));
```

```
    idx=idx+1;
```

```
end
```

```
end
```

```
end
```



```
%X Y koordináta számolása
% x = R * cos(lat) * cos(lon)
% y = R * cos(lat) * sin(lon)
% z = R * sin(lat)
for(i=1:length(lat))
    lat(i) = floor(lat(i)/100) + (lat(i) -
floor(lat(i)/100)*100)/60;
    lon(i) = floor(lon(i)/100) + (lon(i) -
floor(lon(i)/100)*100)/60;
end
%d = D + m/60

stem3(lat, lon, altitude-min(altitude));
view(45,45);
%plot3(lat, lon, altitude);
grid on
```



Figure 1

File Edit View Insert Tools Desktop Window Help

