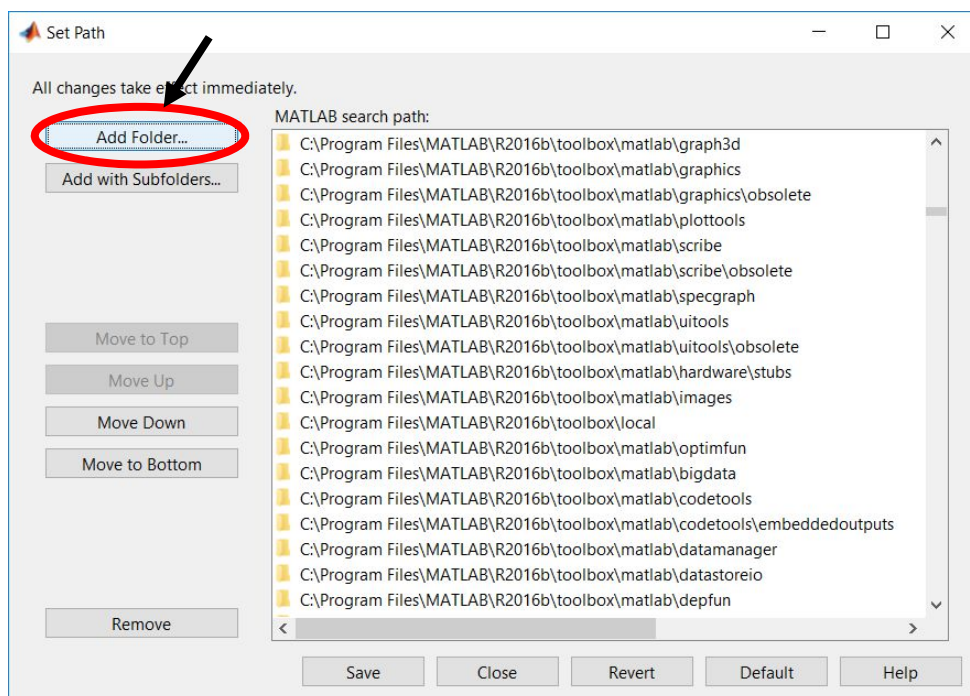
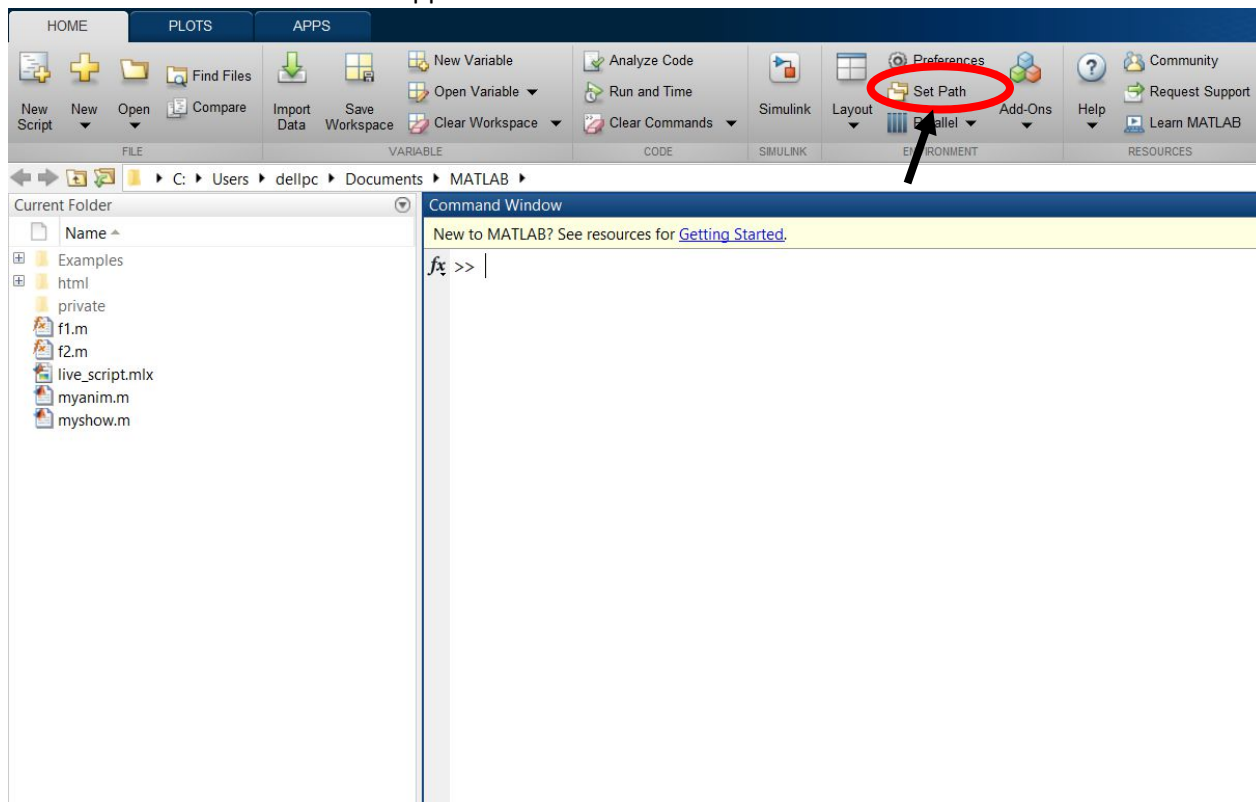
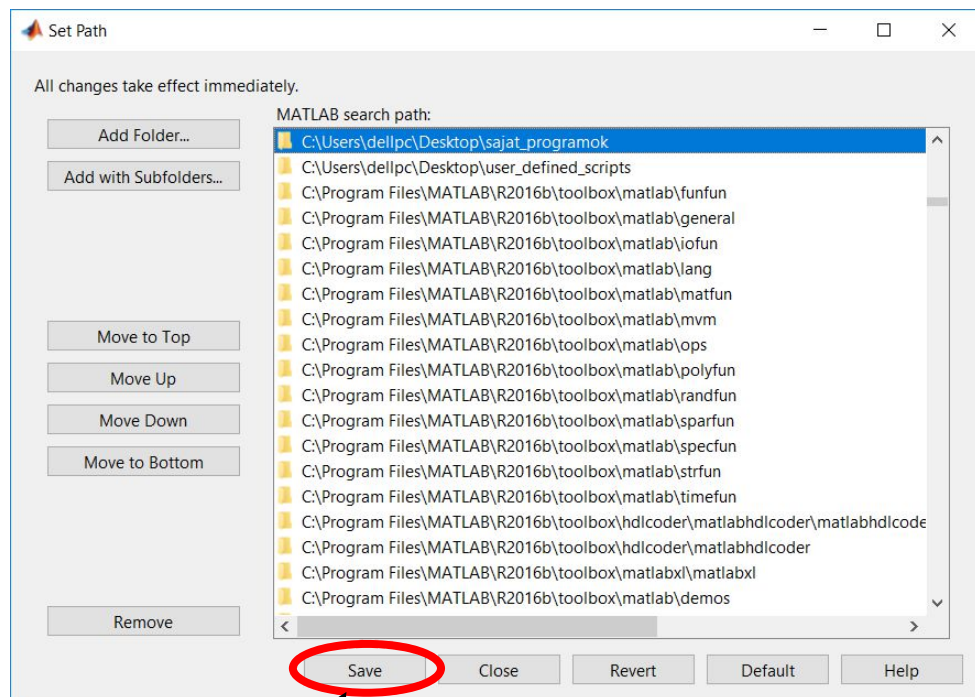
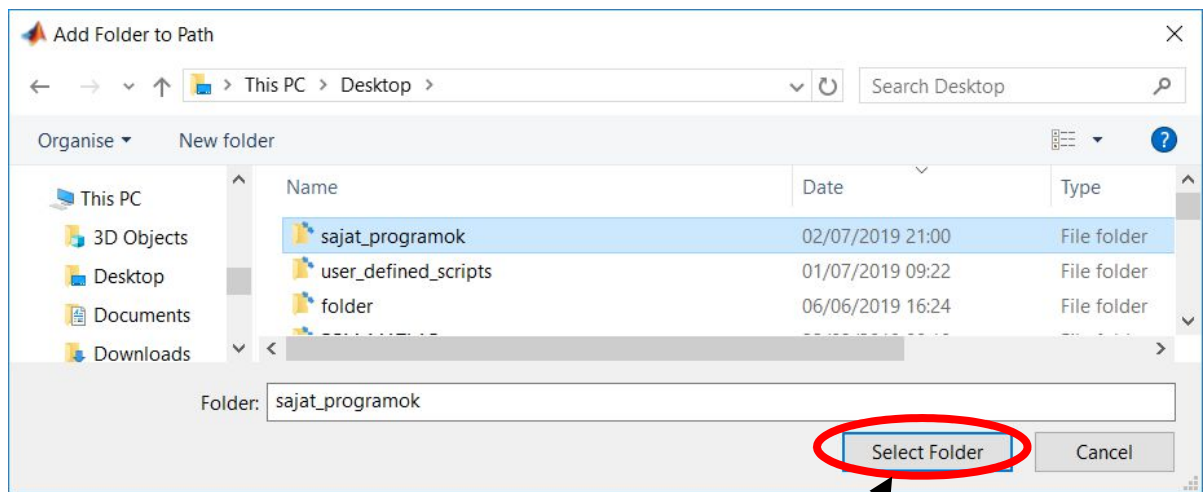


1. Hozzon létre egy tetszőleges nevű mappát. Az elhelyezésekor vegye figyelembe, hogy az elkövetkező feladatokban a Matlab a létrehozott mappát használni fogja. A létrehozott mappát adja hozzá a Matlab keresőútvonalához (Matlab search path). Nyisson meg egy új Matlab script-et. A létrehozott script-et mentse el a létrehozott új mappába. Az elmentett script segítségével rajzoljon ki egy egyenest, amely kezdőpontjának koordinátája (0,0), végpontjának koordinátája (100,100) és a végpontok között 99 pontot tartalmaz.

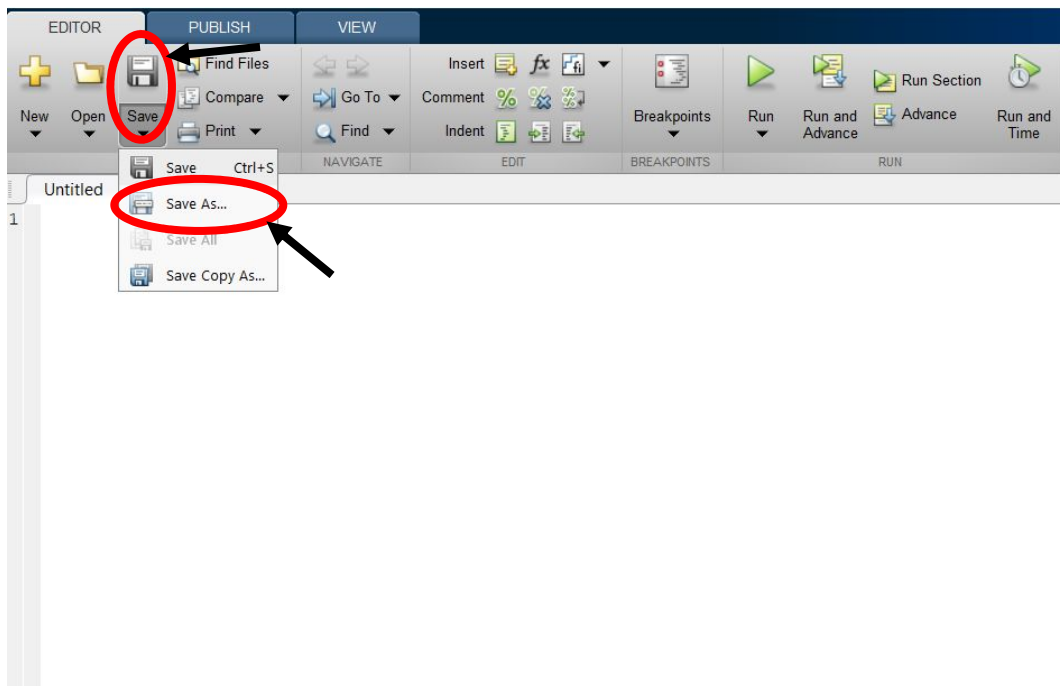
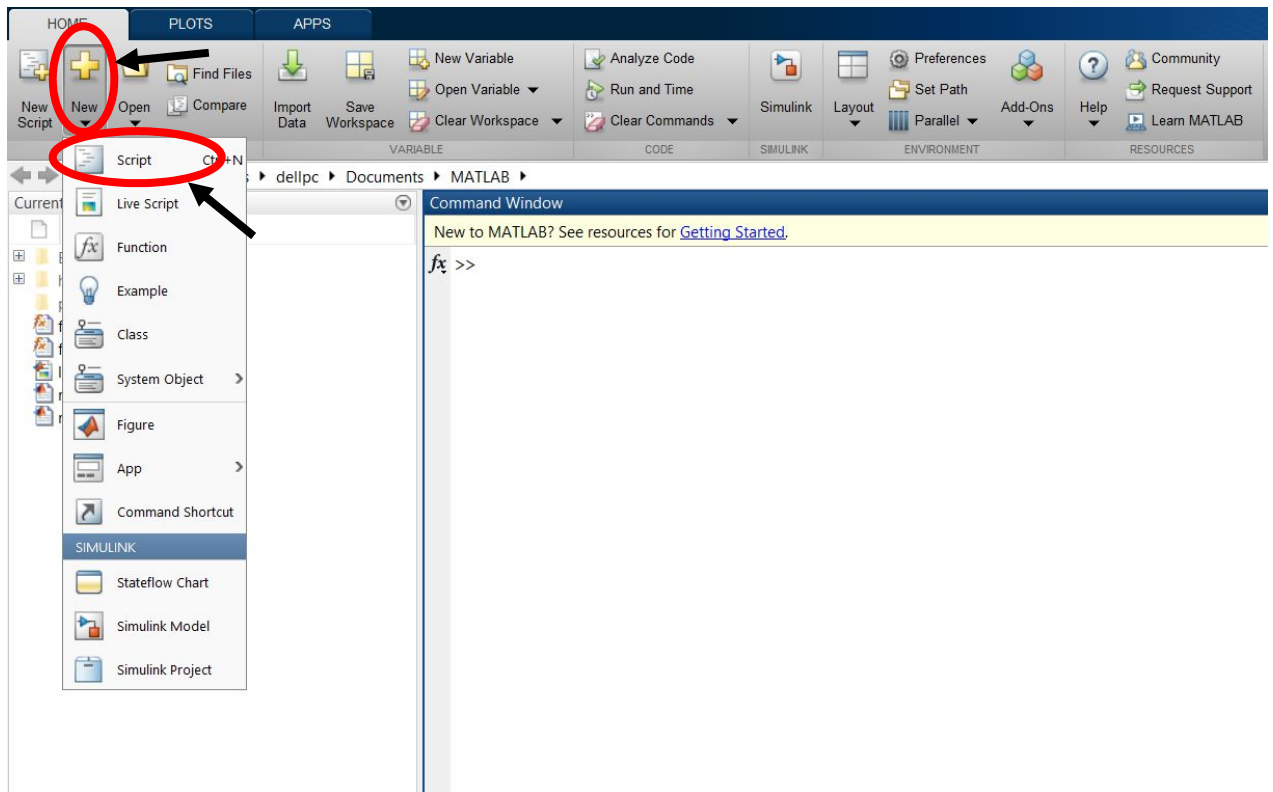
Megoldás:

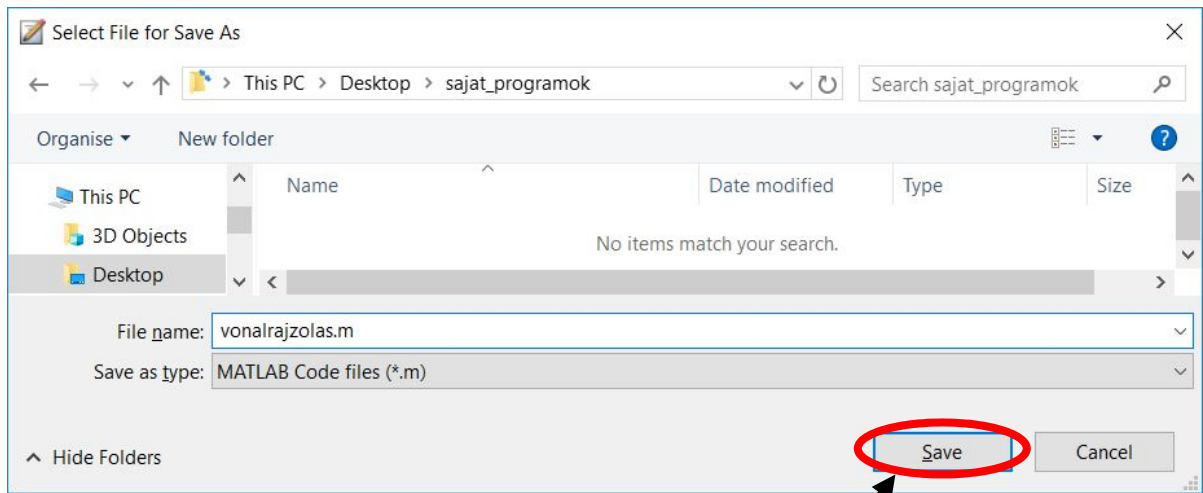
Az új mappa létrehozása után (jelen példában a *sajat_programok* nevű mappát hoztuk létre az asztalon) használjuk a Matlab *Home* címkéje alatt az *Environment* szekcióban található *Set Path* gombot. Erre klikkelve megnyílik egy párbeszédablak, amelynek *Add Folder...* opciója segítségével a fájlrendszerben navigálva kiválasztható a hozzáadni kívánt mappa.





A Matlab *Home* címkéje alatt a *File* szekcióban található a *New* gomb. Erre klikkelve a felugró lehetőségek közül válasszuk a *Script* lehetőséget. A megnyíló, egyelőre üres, névtelen script-et mentsük el az imént létrehozott mappába *vonalarajzolas.m* néven.

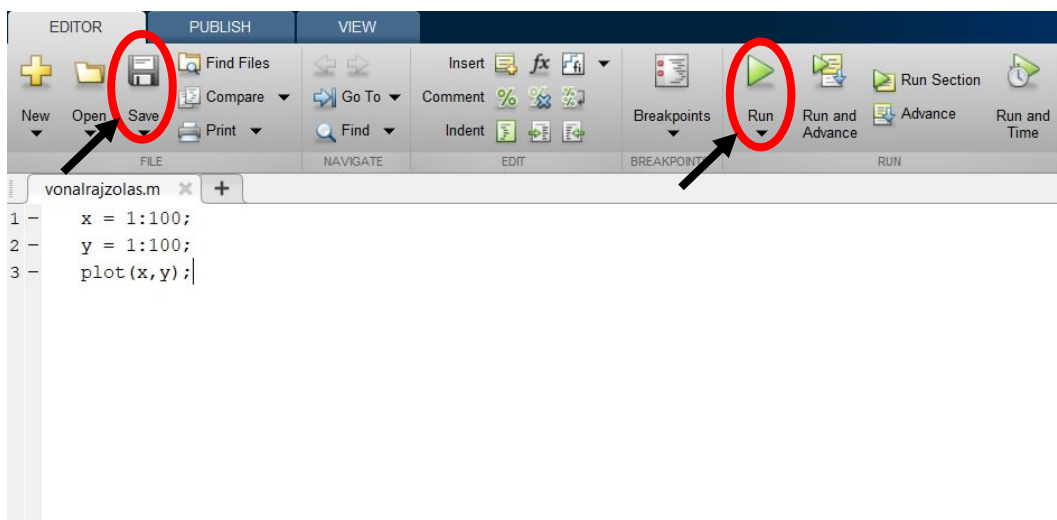




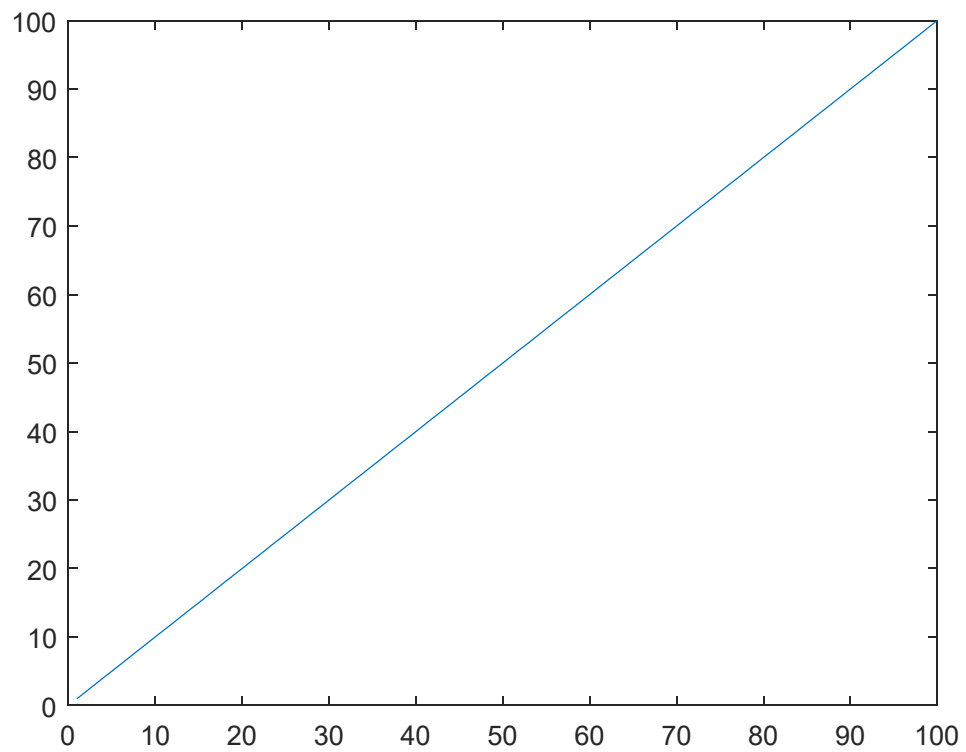
A *vonalarajzas.m* script-be írjuk az alábbi kódot:

```
x = 1:100; % 100 elemet tartalmazó vektor létrehozása  
y = 1:100; % 100 elemet tartalmazó vektor létrehozása  
plot(x,y); % y elemeinek kirajzolása, x elemeinek függvényében
```

Mentsük el, majd futtassuk a script-et.



Eredmény:



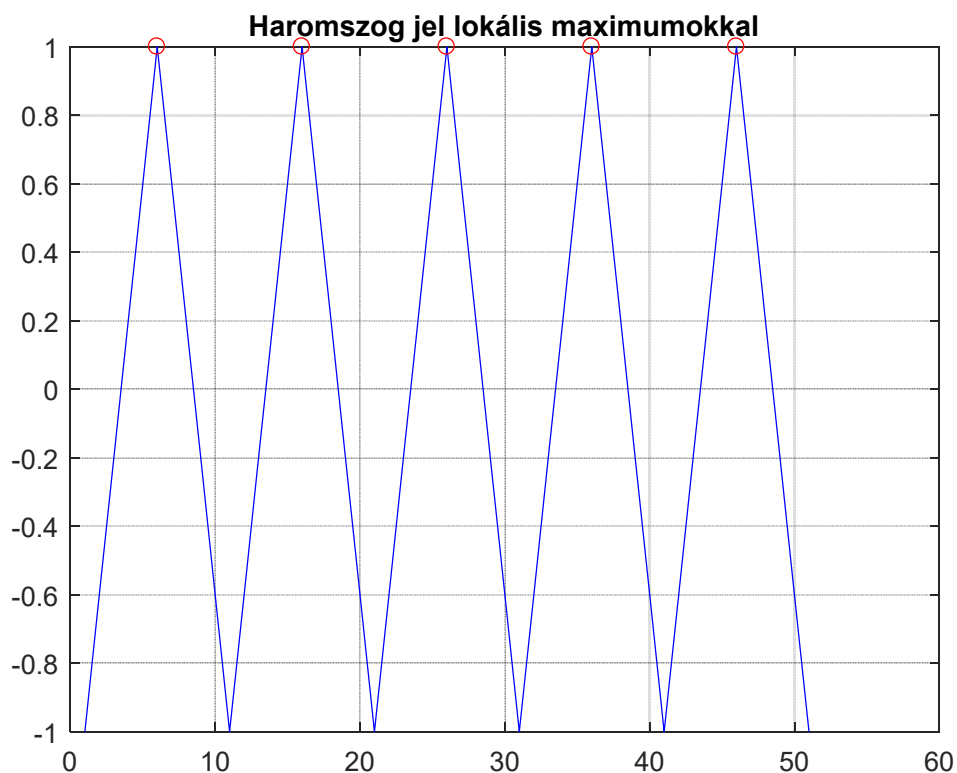
Önálló feladat: Adjon hozzá a Matlab keresőútvonalához két mappát, amelyekben egy-egy azonos nevű, de eltérő kódot tartalmazó Matlab script-et helyez. Próbálja ki, mi történik, ha a parancssorba begépel a script nevét és Enter-t nyom, a kettő közül melyik script fog futni.

2. Készítsen egy Matlab script-et, amely előállítja egy háromszögjel 5 periódusát és a jelben kijelöli a lokális maximumokat. A háromszögjel legyen szimmetrikus, egy periódus tartalmazzon 10 mintát. A program rajzolja ki a háromszögjelet kék színnel és ugyanezen ábrán jelölje meg a lokális maximumokat piros színnel. Segítség: használja a Matlab *sawtooth* és *findpeaks* függvényeit.

Megoldás:

```
T = 10; % változó létrehozása és értékadás
f = 1 / T; % változó létrehozása és értékadás
t = 0:1:5*T; % 0 és 50 közti számok, lépésköz = 1
haromszogjel = sawtooth(2*pi*f*t, 1/2); % szimmetrikus háromszögjel
[val, pos] = findpeaks(haromszogjel, 'MinPeakDistance', 9); % csúcsok keresése a
haromszogjelben úgy, hogy az egyes csúcsok közti távolság legalább 9 minta legyen
figure(1); % új ábra, 1-gyes sorszámmal
plot(haromszogjel, 'b'); % a háromszögjel kirajzolása kék színnel
hold on; % az eddig kirajzolt jelalakok megtartása, amikor új jelet rajzolunk
scatter(pos, val, 'r'); % a „val” változó kirajzolása a „pos” változó
függvényében, piros színnel úgy, hogy az egyes pontok nincsenek összekötve
grid on; % rács rajzolása
title('Haromszog jel lokális maximumokkal'); % az ábra címének beállítása
```

Eredmény:

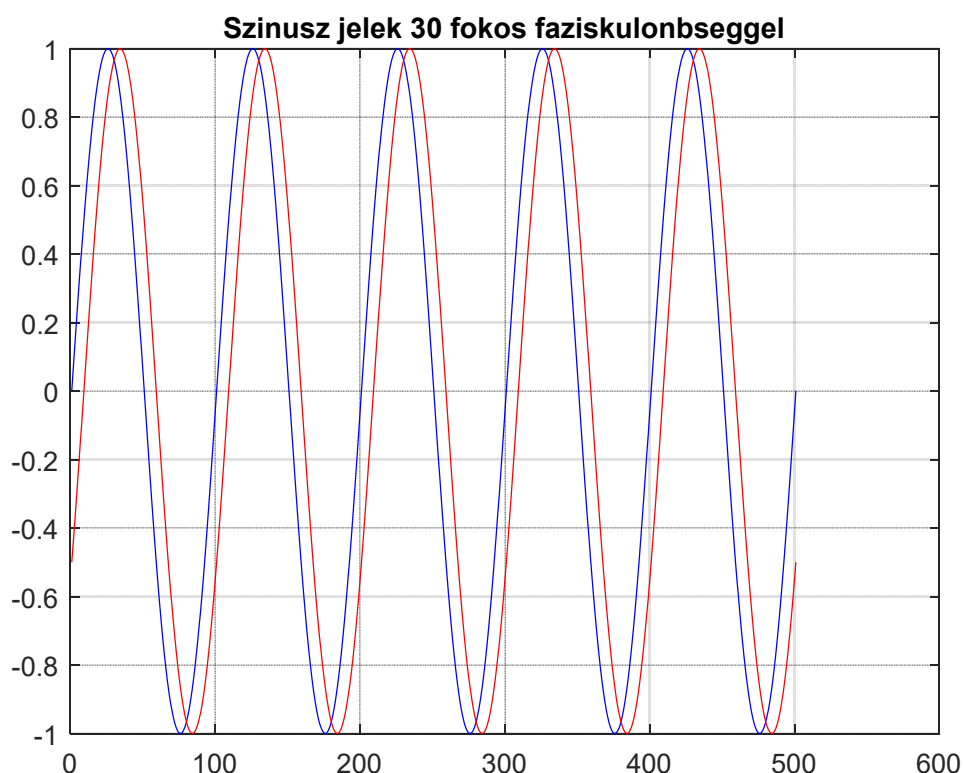


3. Készítsen egy Matlab script-et, amely előállítja két szinusz jel 5 periódusát. A szinusz jelek amplitúdója és frekvenciája legyen megegyező, a fáziskülönbség köztük legyen 30° . A szinusz jelek egy periódusa tartalmazzon 100 mintát. A program rajzolja ki az első jelet kék színnel, az ehhez képest 30° -kal késő jelet piros színnel, egy ábrára.

Megoldás:

```
T = 100; % változó létrehozása és értékadás
f = 1 / T; % változó létrehozása és értékadás
t = 0:1:5*T; % 0 és 500 közti számok, lépésköz = 1
sin_1 = sin(2*pi*f*t); % szinusz jel, frekvencia = 0.01, fázis =  $0^\circ$ 
sin_2 = sin(2*pi*f*t-pi/6); % szinusz jel, frekvencia = 0.01, fázis =  $-30^\circ$ 
figure(1); % új ábra, 1-gyes sorszámmal
plot(sin_1, 'b'); % a sin_1 jel kirajzolása kék színnel
hold on; % az eddig kirajzolt jelalakok megtartása, amikor új jelet rajzolunk
plot(sin_2, 'r'); % a sin_2 jel kirajzolása piros színnel
grid on; % rács rajzolása
title('Szinusz jelek 30 fokos faziskulonbseggel'); % az ábra címének beállítása
```

Eredmény:



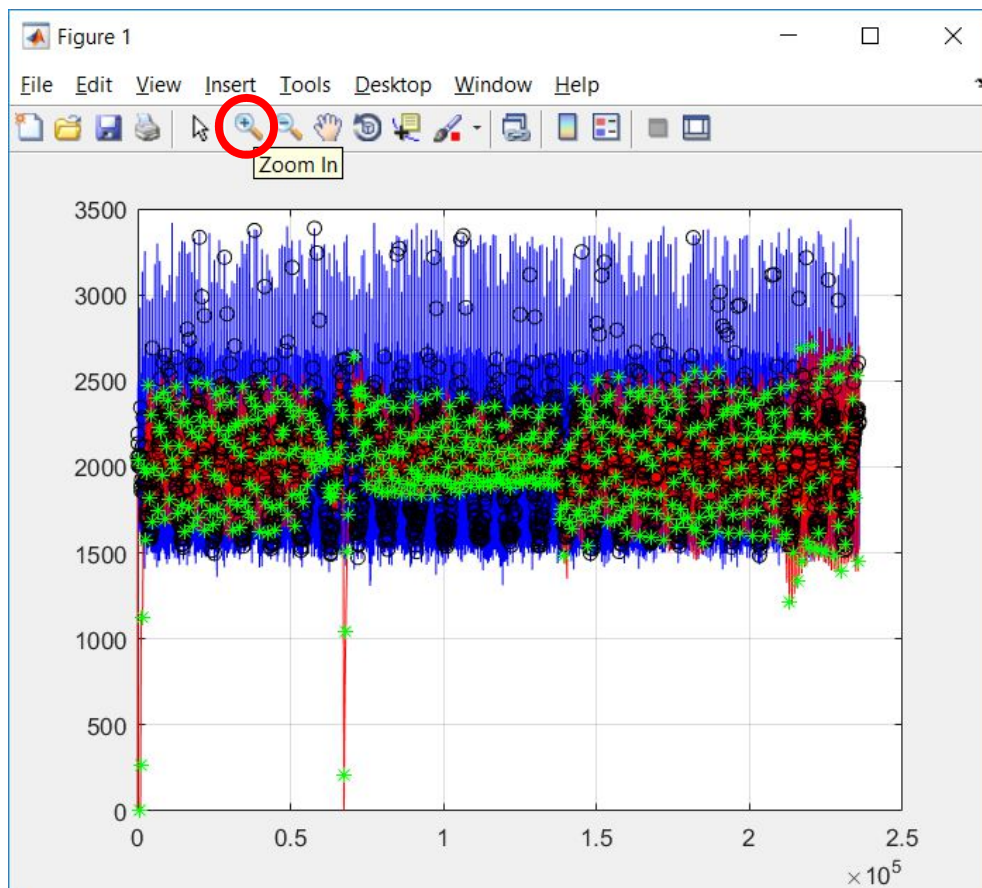
Önálló feladat: Állítson elő és ábrázoljon két szinusz jelet, amelyek frekvenciája nem egyezik meg, az első jel frekvenciája legyen a második jel frekvenciájának 1,5-szerese.

4. Készítsen egy Matlab script-et, amely beolvas egy, a HHMD-vel készült felvételt, majd ugyanazon ábrára kirajzolja az *ecg1* elnevezésű jelet kék színnel és a *ppgl_nir* elnevezésű jelet piros színnel. Válassza ki az *ecg1* elnevezésű jel minden századik mintáját, és ezeket a pontokat jelölje meg fekete színű körrel. Válassza ki a *ppgl_nir* elnevezésű jel minden ötszázadik mintáját és ezeket a pontokat jelölje meg zöld színű csillaggal. Segítség: Másolja be a *hhmbinread.m* nevű Matlab scrip-et abba a mappába, amelybe a saját script-jét helyezi és használja a *hhmbinread* függvényt a jelek beolvasásához.

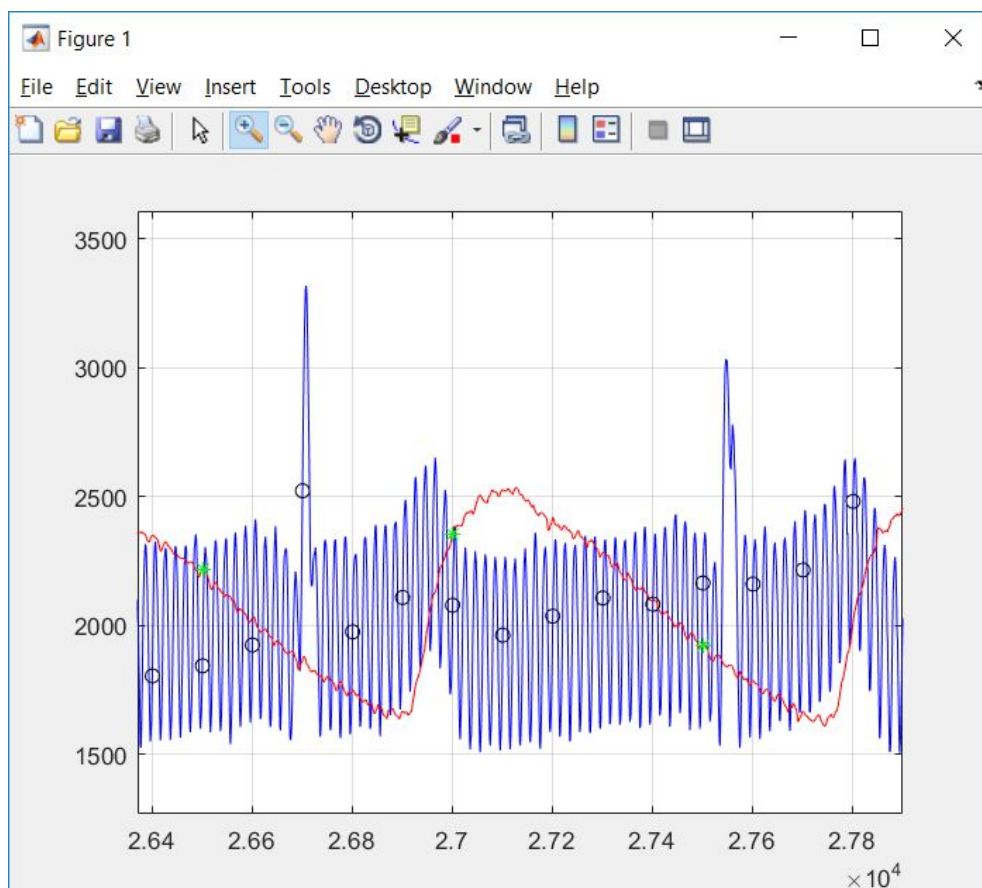
Megoldás:

```
JA_jelei = hhmbinread('D:\hhmdl_recordings\ja_05_12_2017.hhm'); % fájl beolvasása
nyersECG = JA_jelei.ecg1; % az ecg1 nevű jel kiolvasása
nyersPPG = JA_jelei.ppgl_nir; % a ppgl_nir nevű jel kiolvasása
figure(1); % új ábra, 1-gyes sorszámmal
plot(nyersECG, 'b'); % a nyersECG jel kirajzolása kék színnel
hold on; % az eddig kirajzolt jelalakok megtartása, amikor új jelet rajzolunk
plot(nyersPPG, 'r'); % a nyersPPG jel kirajzolása piros színnel
grid on; % rács rajzolása
pontok_szama_ECG = numel(nyersECG); % a nyersECG jelben található elemek számának meghatározása
pontok_szama_PPG = numel(nyersPPG); % a nyersPPG jelben található elemek számának meghatározása
kivalasztott_pontok_ECG = 1:100:pontok_szama_ECG; % a nyersECG jelben található elemek közül minden századik kiválasztása
kivalasztott_pontok_PPG = 1:500:pontok_szama_PPG; % a nyersPPG jelben található elemek közül minden ötszázadik kiválasztása
scatter(kivalasztott_pontok_ECG, nyersECG(kivalasztott_pontok_ECG), 'k', 'o'); % a nyersECG jelből kiválasztott pontok kirajzolása fekete körökkel úgy, hogy a pontok nincsenek összekötve
scatter(kivalasztott_pontok_PPG, nyersPPG(kivalasztott_pontok_PPG), 'g', '*'); % a nyersPPG jelből kiválasztott pontok kirajzolása zöld csillagokkal úgy, hogy a pontok nincsenek összekötve
```

Eredmény:



Jelenlegi formájában az ábra a felvétel globális kiértékelésre használható (például kilógó értékek láthatók a PPG jelen), azonban a felvétel részletes vizuális értékeléséhez ki kell nagyítani az ábrát (*Zoom In*).

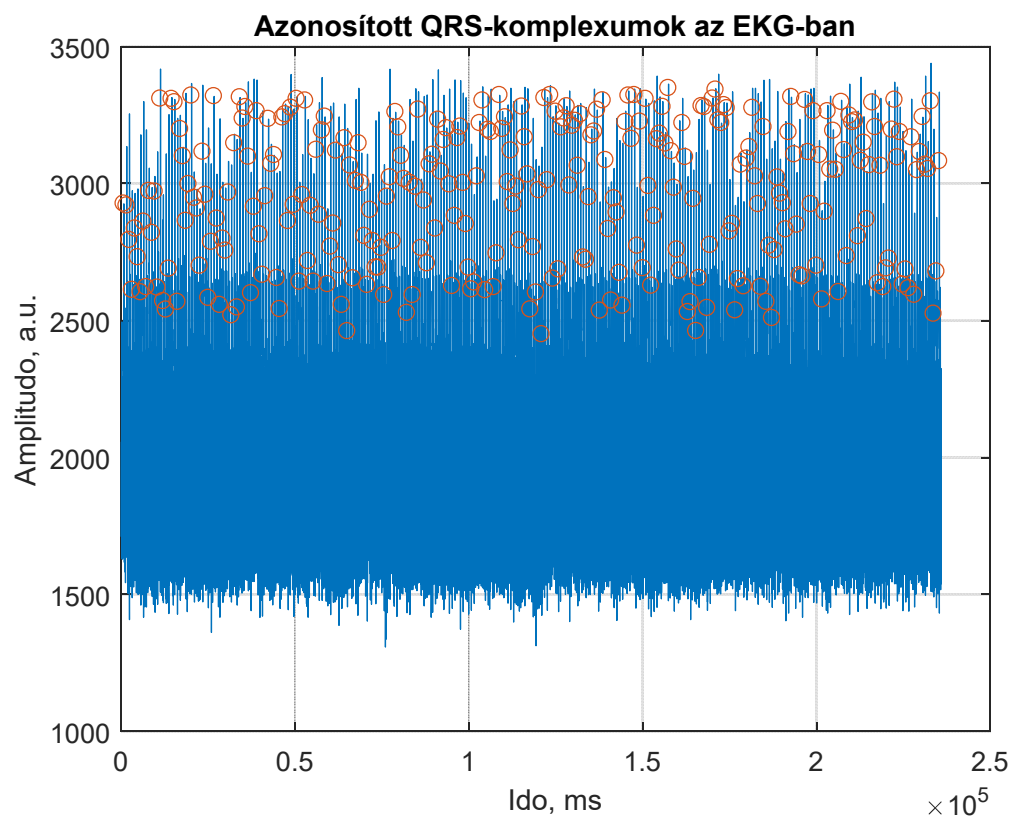


5. Készítsen egy Matlab script-et, amely egy egyszerű QRS detektort valósít meg. A detektált QRS-komplexumokat jelölje meg az eredeti jelen egy ábrán.

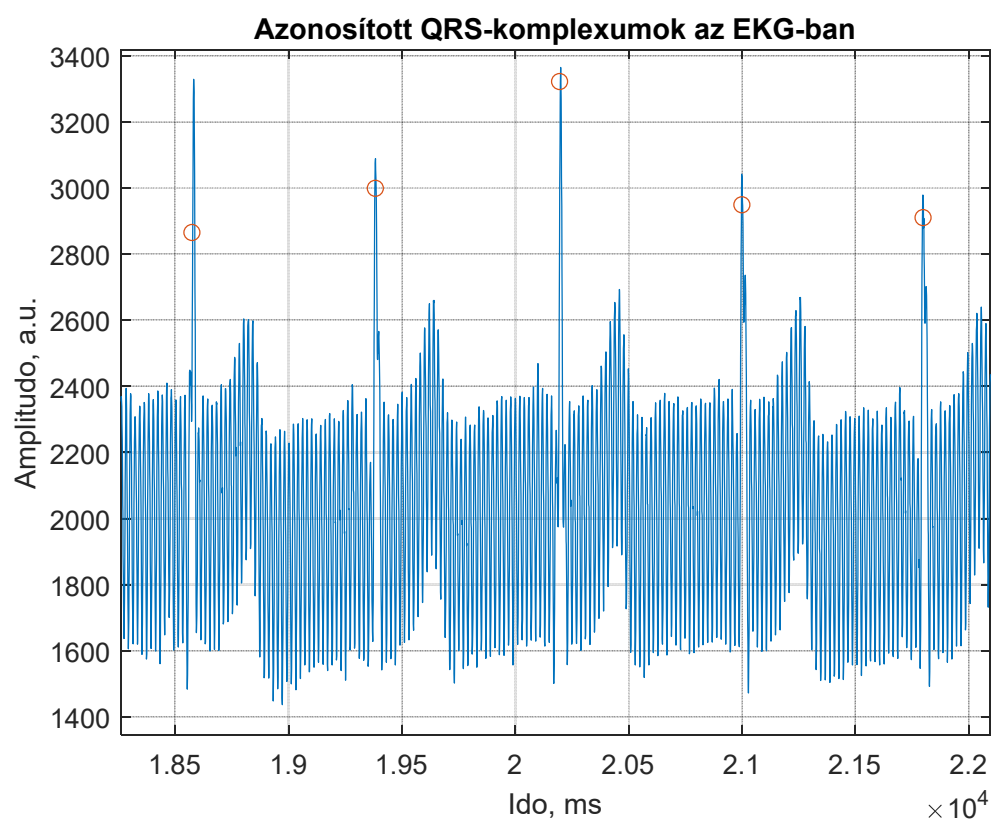
Megoldás:

```
JA_jelei = hhmbinread('D:\hhmdl_recordings\ja_05_12_2017.hhm'); % fájl beolvasása
nyersECG = JA_jelei.ecg1; % az ecg1 nevű jel kiolvasása
N = 3; % a szűrő fokszámának eltárolása az N változóban
W = [5 15]; % a szűrő alsó és felső határfrekvenciájának eltárolása a W
változóban
Fs = 1000; % a mintavételi frekvencia eltárolása az Fs változóban
normalt_frekvencia = W / (Fs / 2); % a szűrő határfrekvenciáját normálni kell (0
és 1 közötti érték kell, hogy legyen), ahol 1 a mintavételi frekvencia felének
felel meg
[b, a] = butter(N, normalt_frekvencia); % a szűrő fokszáma és normált
határfrekvenciája alapján a butter függvény előállítja a butterworth szűrő
számlálójának és nevezőjének együtthatóit
szurteCG = filtfilt(b, a, nyersECG); % a megtervezett szűrő számlálójának és
nevezőjének együtthatói alapján a filtfilt nulla fázistolású szűrést hajt végre
derivaltECG = diff(szurteCG); % a szűrt jel deriválása
negyzetesECG = derivaltECG .^ 2; % a derivált jel négyzetre emelése
integraltECG = []; % üres vektor az integrált jel értékeinek későbbi tárolásához
for i = 1 : numel(negyzetesECG) - 149
    integraltECG(i) = mean(negyzetesECG(i:i+149)); % 150 minta szélességű mozgó
ablakos integrálás végrehajtása
end
[val, pos] = findpeaks(integraltECG, 'MinPeakDistance', 200); % csúcsok keresése
az integrált jelben úgy, hogy az egyes csúcsok közti távolság legalább 200 minta
legyen (200 ms-on belül nem következhet be két QRS)
ECG_terjedelem = max(integraltECG) - min(integraltECG); % az integrált jel
legnagyobb és legkisebb elemének különbségét használjuk annak eldöntésére, hogy a
findpeaks által talált csúcsok QRS-hez tartoznak, vagy nem
kivalasztottCsucok = []; % üres vektor a QRS-ként azonosított csúcsok későbbi
tárolására
for i = 1 : numel(val)
    if val(i) > ECG_terjedelem * 0.6
        kivalasztottCsucok = [kivalasztottCsucok pos(i)]; % az integrált jel
azon csúcsait tekintjük QRS-nek, ahol a csúcs nagyobb, mint a jel legnagyobb és
legkisebb elemének különbségének 0,6-szerese, ezzel kizárva az integrált jel alsó
részén lévő lokális maximumokat
    end
end
azonositottQRSKomplexumok = kivalasztottCsucok + 75; % a mozgó ablakos
integrálás torzításának kompenzálása
figure(1); % új ábra, 1-gyes sorszámmal
plot(nyersECG); % a nyersECG jel kirajzolása
grid on; % rács rajzolása
hold on; % az eddig kirajzolt jelalakok megtartása, amikor új jelet rajzolunk
scatter(azonositottQRSKomplexumok, nyersECG(azonositottQRSKomplexumok)); % a
nyersECG jelből QRS-ként azonosított pontok kirajzolása úgy, hogy a pontok
nincsenek összekötve
title('Azonosított QRS-komplexumok az EKG-ban'); % az ábra címének beállítása
xlabel('Ido, ms'); % az x-tengely címének beállítása
ylabel('Amplitudo, a.u.');
```

Eredmény:



Kinagyítás után:

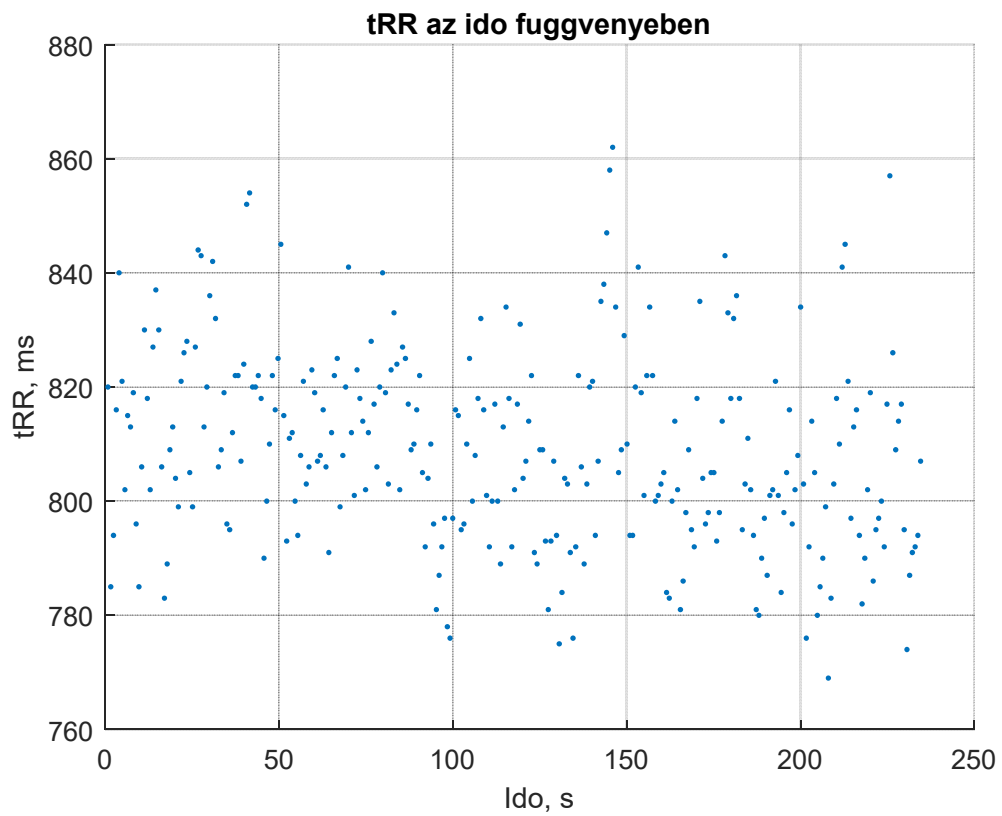


6. Készítsen egy Matlab script-et, amely beolvas egy, a HHMD-vel készült felvételt majd meghatározza az *ecg1* elnevezésű jel alapján a szív ciklus hosszokat (tRR). Ábrázolja a szív ciklus hosszokat az idő függvényében, az időegység legyen másodperc. Megjegyzés: A szív ciklus hosszok egy vektort alkotnak, ezért közvetlenül is ábrázolhatók (*plot(tRR)*), így azonban csak az derül ki az ábrából, hogy hányadik szívütéshez tartozik egy adott szív ciklus hossz, nem derül ki, hogy a szívütés mikor következett be. Átlagos szívfrekvencia (70 ütés / perc) esetén a szív ciklus sorszáma és bekövetkezésének másodpercben mért időpontja hasonló eredményt ad, ezért fontos egyértelműen jelölni, hogy a szívütések sorszáma, vagy a szívütések bekövetkezésének időpontja látható az x-tengelyen.

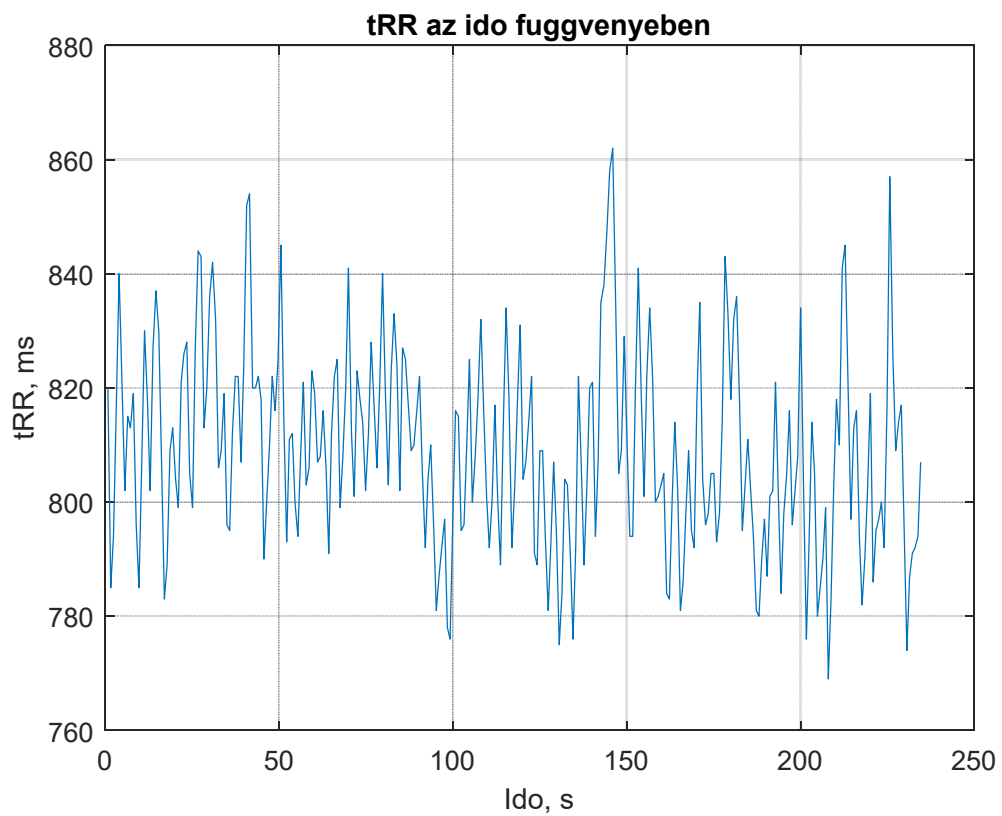
Megoldás:

```
JA_jelei = hhmbinread('D:\hhmdl_recordings\ja_05_12_2017.hhm');
nyersECG = JA_jelei.ecg1;
N = 3;
W = [5 15];
Fs = 1000;
normalt_frekvencia = W / (Fs / 2);
[b, a] = butter(N, normalt_frekvencia);
szurtECG = filtfilt(b, a, nyersECG);
derivaltECG = diff(szurtECG);
negyzetesECG = derivaltECG .^ 2;
integraltECG = [];
for i = 1 : numel(negyzetesECG) - 149
    integraltECG(i) = mean(negyzetesECG(i:i+149));
end
[val, pos] = findpeaks(integraltECG, 'MinPeakDistance', 200);
ECG_terjedelem = max(integraltECG) - min(integraltECG);
kivalasztottCsucsok = [];
for i = 1 : numel(val)
    if val(i) > ECG_terjedelem * 0.6
        kivalasztottCsucsok = [kivalasztottCsucsok pos(i)];
    end
end
azonositottQRSKomplexumok = kivalasztottCsucsok + 75;
tRR = diff(azonositottQRSKomplexumok); % az azonosított QRS-ek közti időkülönbség
(a szív ciklus hosszok, tRR) meghatározása
ido_millisec = azonositottQRSKomplexumok(1:end-1); % a feladat szerint a
szív ciklus hosszokat az idő függvényében kell ábrázolni. Itt ezt úgy kezeljük,
hogy a szív ciklust meghatározó két QRS közül a korábbi időpontjához rendeljük az
adott szív ciklust. Így az első QRS-től az utolsó előtti QRS-ig, minden QRS
időpontjához tartozni fog egy szív ciklus hossz. A mintavételi frekvencia 1 kHz,
tehát az időpontok ms-ban értendők.
ido_sec = ido_millisec / 1000; % az időpontok átszámolása ms-ból s-ba
figure(1); % új ábra, 1-gyes sorszámmal
scatter(ido_sec, tRR, '.'); % a tRR-ek ábrázolása a s-ban értelmezett időpontok
függvényében úgy, hogy a pontok nincsenek egymással összekötve. A pontokat
pöttyökkel ábrázoljuk. Érdemes kipróbálni, milyen hatást kelt az ábra, ha a
pontokat összekötjük: plot(ido_sec, tRR)
grid on; % rács rajzolása
title('tRR az ido függvényében'); % az ábra címének beállítása
xlabel('Ido, s'); % az x-tengely címének beállítása
ylabel('tRR, ms'); % az y-tengely címének beállítása
```

Eredmény:



A pontokat összekötve jól érzékeltethető az időbeliség.



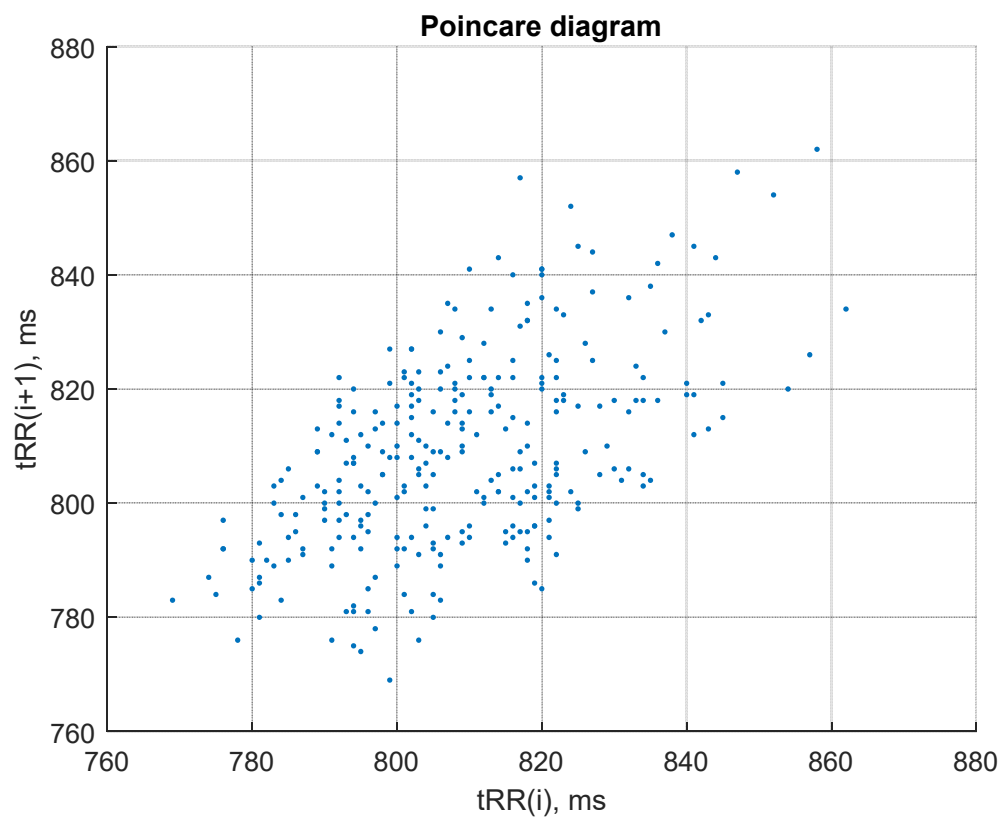
Önálló feladat: Ábrázolja a szomszédos szív ciklushosszok különbségét az idő függvényében.

7. Készítsen egy Matlab script-et, amely beolvas egy, a HHMD-vel készült felvételt majd kirajzolja az *ecg1* elnevezésű jel alapján számított Poincaré-diagramot.

Megoldás:

```
JA_jelei = hhmbinread('D:\hhmdl_recordings\ja_05_12_2017.hhm');
nyersECG = JA_jelei.ecg1;
N = 3;
W = [5 15];
Fs = 1000;
normalt_frekvencia = W / (Fs / 2);
[b, a] = butter(N, normalt_frekvencia);
szurteCG = filtfilt(b, a, nyersECG);
derivalteCG = diff(szurteCG);
negyzetesECG = derivalteCG .^ 2;
integralteCG = [];
for i = 1 : numel(negyzetesECG) - 149
    integralteCG(i) = mean(negyzetesECG(i:i+149));
end
[val, pos] = findpeaks(integralteCG, 'MinPeakDistance', 200);
ECG_terjedelem = max(integralteCG) - min(integralteCG);
kivalasztottCsucsok = [];
for i = 1 : numel(val)
    if val(i) > ECG_terjedelem * 0.6
        kivalasztottCsucsok = [kivalasztottCsucsok pos(i)];
    end
end
azonositottQRSKomplexumok = kivalasztottCsucsok + 75;
tRR = diff(azonositottQRSKomplexumok);
poincare_x = tRR(1:end-1); % a Poincare-diagram x-koordinátáinak (tRR(i))
meghatározása
poincare_y = tRR(2:end); % a Poincare-diagram y-koordinátáinak (tRR(i+1))
meghatározása
figure(1); % új ábra, 1-gyes sorszámmal
scatter(poincare_x, poincare_y, '.'); % az y-koordináták ábrázolása az x-
koordináták függvényében úgy, hogy a pontok nincsenek egymással összekötve. A
pontokat pöttyökkel ábrázoljuk.
grid on; % rács rajzolása
title('Poincare diagram'); % az ábra címének beállítása
xlabel('tRR(i), ms'); % az x-tengely címének beállítása
ylabel('tRR(i+1), ms'); % az y-tengely címének beállítása
```

Eredmény:



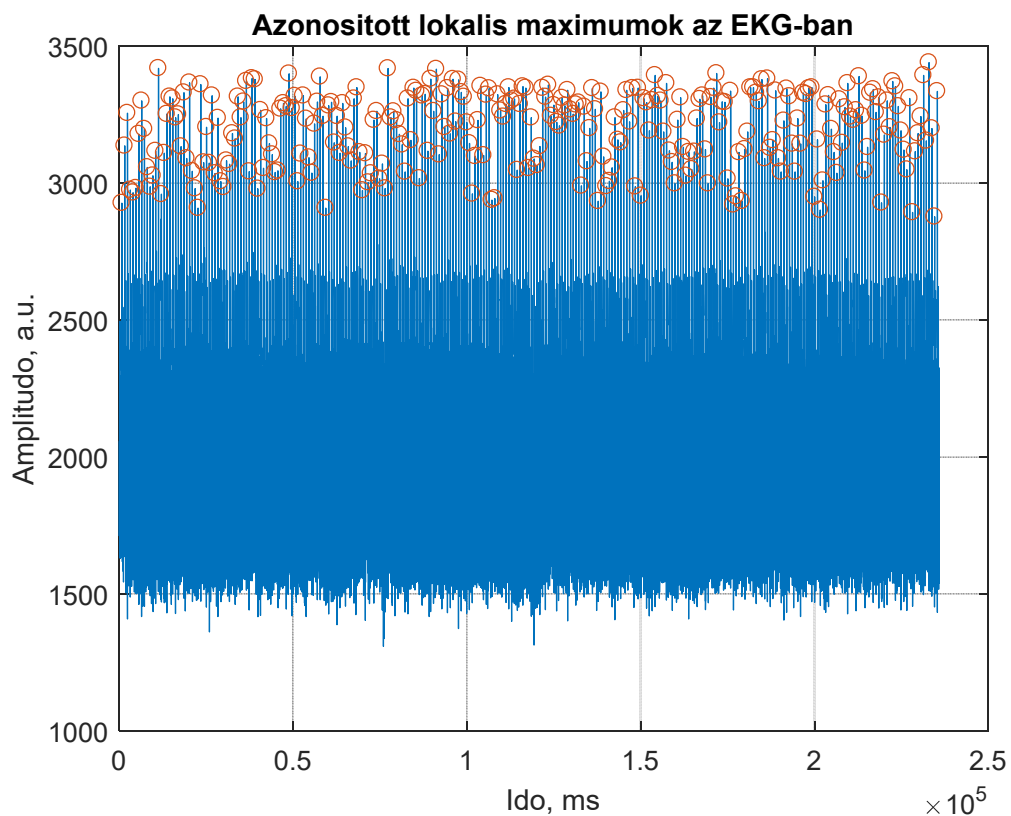
Önálló feladat: A fenti ábrába rajzoljon be egy 45° -os egyenest.

8. Fejlessze tovább az 5. feladatban készített QRS detektort úgy, hogy az azonosított QRS-komplexumok 50 minta szélességű környezetében keresse meg a nyers EKG jel lokális maximumát. A detektált lokális maximumokat jelölje meg az eredeti jelen egy ábrán.

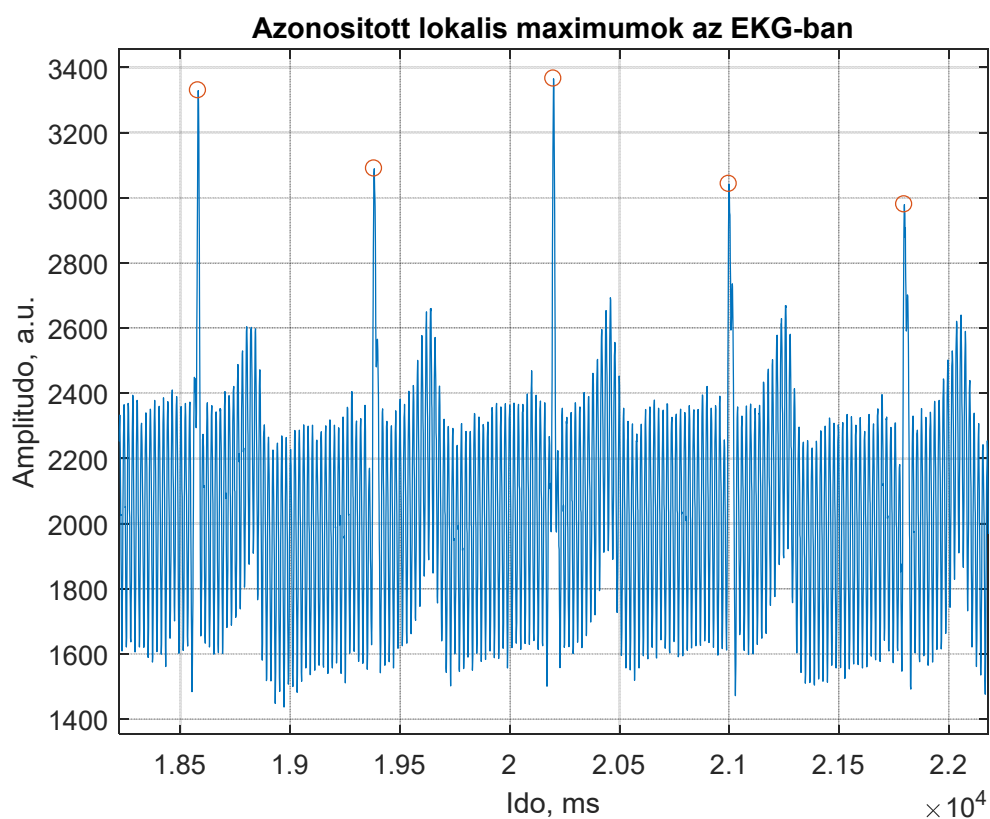
Megoldás:

```
JA_jelei = hhmbinread('D:\hhmdl_recordings\ja_05_12_2017.hhm');
nyersECG = JA_jelei.ecg1;
N = 3;
W = [5 15];
Fs = 1000;
normalt_frekvencia = W / (Fs / 2);
[b, a] = butter(N, normalt_frekvencia);
szurtECG = filtfilt(b, a, nyersECG);
derivaltECG = diff(szurtECG);
negyzetesECG = derivaltECG .^ 2;
integraltECG = [];
for i = 1 : numel(negyzetesECG) - 149
    integraltECG(i) = mean(negyzetesECG(i:i+149));
end
[val, pos] = findpeaks(integraltECG, 'MinPeakDistance', 200);
ECG_terjedelem = max(integraltECG) - min(integraltECG);
kivalasztottCsucsok = [];
for i = 1 : numel(val)
    if val(i) > ECG_terjedelem * 0.6
        kivalasztottCsucsok = [kivalasztottCsucsok pos(i)];
    end
end
azonositottQRSKomplexumok = kivalasztottCsucsok + 75;
lokalisMaximumok = []; % üres vektor a lokális maximumok későbbi tárolására
for i = 1 : numel(azonositottQRSKomplexumok) % végig megyünk az összes QRS-ként
    azonosított ponton
        eddigiMaxPont = azonositottQRSKomplexumok(i); % a lokális maximum keresés
        kezdetén az eddig talált legnagyobb pont az aktuálisan vizsgált QRS-ként
        azonosított pont
        for j = 1 : 50 % az aktuálisan vizsgált QRS-ként azonosított pont 50 minta
            szélességű környezetének pontjain megyünk végig
                if nyersECG(azonositottQRSKomplexumok(i) + j) > nyersECG(eddigiMaxPont)
                    eddigiMaxPont = azonositottQRSKomplexumok(i) + j;
                end
                if nyersECG(azonositottQRSKomplexumok(i) - j) > nyersECG(eddigiMaxPont)
                    eddigiMaxPont = azonositottQRSKomplexumok(i) - j;
                end
            end
            % végig megyünk a környezet pontjain, és ha olyan pontot találunk, akár pozitív,
            akár negatív irányban, amely nagyobb, mint az eddig talált legnagyobb pont, akkor
            mostantól ez a pont lesz az eddig talált legnagyobb pont
            end
            lokalisMaximumok(i) = eddigiMaxPont; % ha végig mentünk a környezet összes
            pontján, akkor az ezek között talált legnagyobb pont lesz az adott QRS-hez
            tartozó lokális maximum
        end
    end
figure(1);
plot(nyersECG);
grid on;
hold on;
scatter(lokalisMaximumok, nyersECG(lokalisMaximumok));
title('Azonosított lokális maximumok az EKG-ban');
xlabel('Ido, ms');
ylabel('Amplitudo, a.u.');
```


Eredmény:



Kinagyítás után:



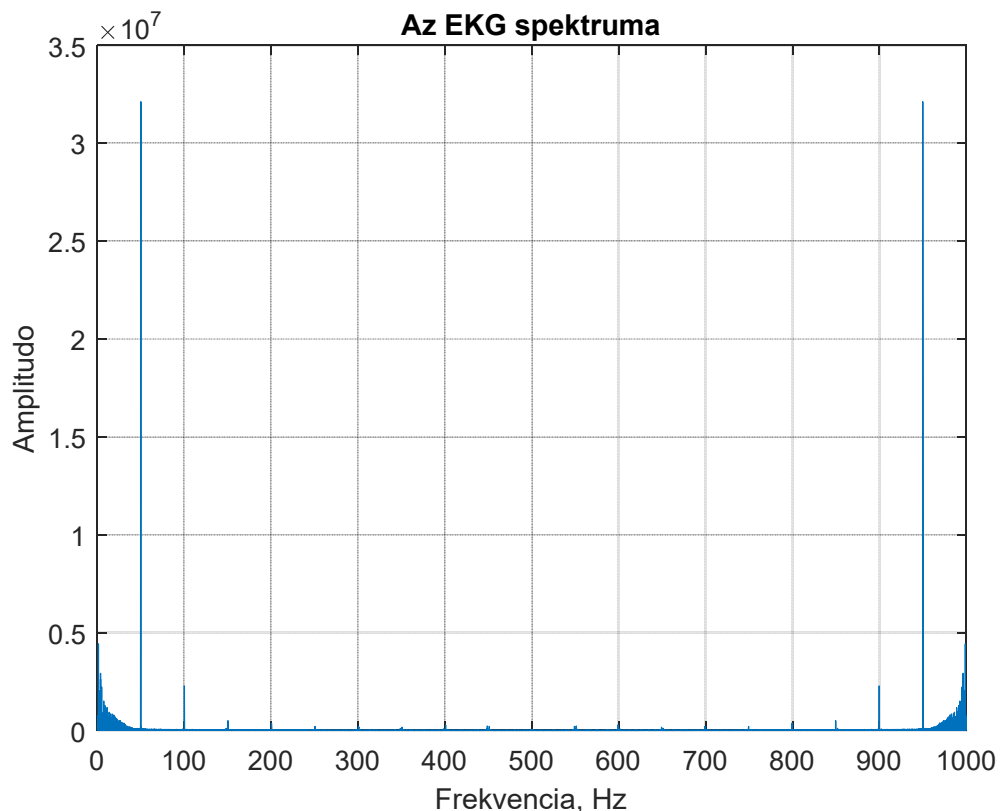
Önálló feladat: A kapott EKG lokális maximumok 300 minta szélességű környezetében határozza meg az ugyanezen felvételhez tartozó *ppgl_nir* jelben található lokális minimumokat. Segítség: Első lépésben célszerű ábrázolni, hová esnek a PPG jelen azok a pontok, amelyek az EKG lokális maximumok időpillanatának felelnek meg.

9. Készítsen egy Matlab script-et, amely beolvas egy, a HHMD-vel készült felvételt majd meghatározza az *ecg1* elnevezésű jel spektrumát. Ügyeljen a vízszintes tengely (frekvencia) megfelelő skálázására, vegye figyelembe, hogy a HHMD-vel készült felvételek esetén a mintavételi frekvencia 1 kHz.

Megoldás:

```
JA_jelei = hhmbinread('D:\hhmdl_recordings\ja_05_12_2017.hhm'); % fájl beolvasása
nyersECG = JA_jelei.ecg1; % az ecg1 nevű jel kiolvasása
ECG_atlag_kivonva = nyersECG - mean(nyersECG); % az jel középértékének kivonása
ECG_spektrum = abs(fft(ECG_atlag_kivonva)); % a jel Fourier-transzformációja után
kapott komplex számok abszolút értékének meghatározása
Fs = 1000; % a mintavételi frekvencia 1 kHz
f = (1:numel(ECG_spektrum)) / numel(ECG_spektrum) * Fs; % a Matlab a spektrumot 0
Hz-től a mintavétel frekvenciáig ábrázolja annyi ponton, ahány pontból az
időfüggvény áll. Jelen esetben a spektrum első pontja 0 Hz-nek, utolsó pontja
1000 Hz-nek felel meg, ezért használjuk a frekvencia tengely átskálázása során a
spektrum elemszámával való osztást és a mintavételi frekvenciával való szorzást.
A kirajzolt spektrum a mintavételi frekvencia felére szimmetrikus, ezért a
spektrum 500 Hz feletti része plusz információt nem hordoz a 0-500 Hz
tartományhoz képest.
figure(1);
plot(f, ECG_spektrum); % a spektrum pontjait az átskálázott frekvencia
függvényében ábrázoljuk
grid on;
title('Az EKG spektruma');
xlabel('Frekvencia, Hz');
ylabel('Amplitudo');
```

Eredmény:



Önálló feladat: Módosítsa a fenti script-et úgy, hogy csak a 0-500 Hz tartomány jelenjen meg az ábrán.

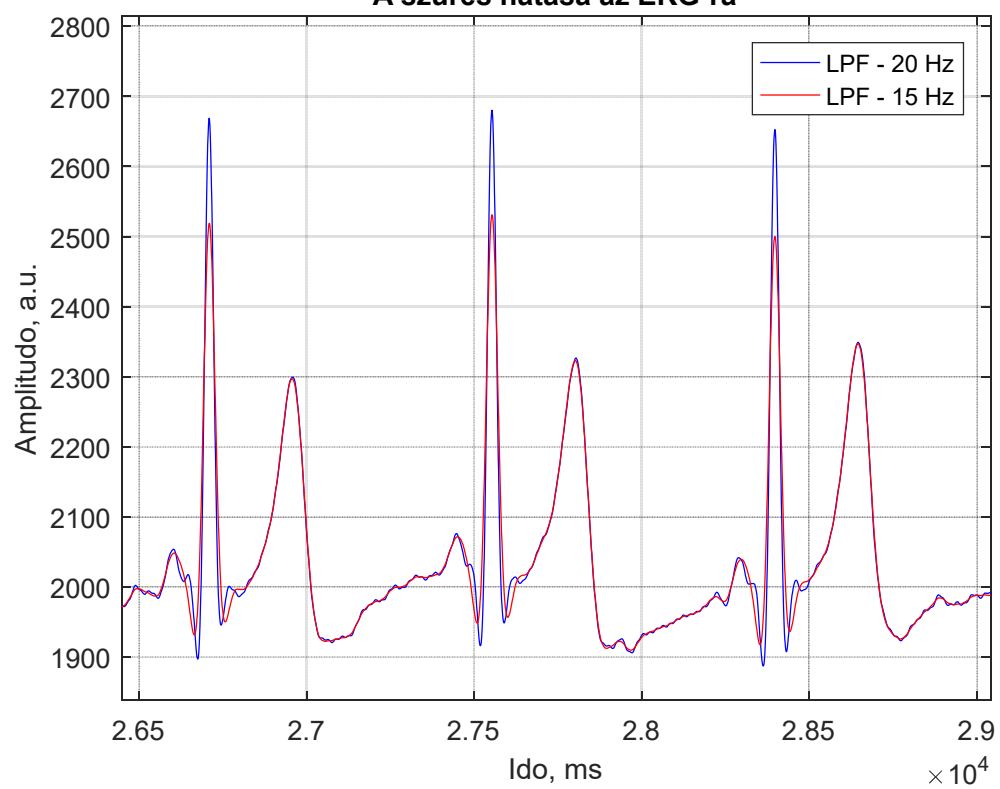
10. Készítsen egy Matlab script-et, amely beolvas egy, a HHMD-vel készült felvételt, majd a felvétel *ecg1* elnevezésű jelén digitális szűrést hajt végre. Hasonlítsa össze két különböző szűrő hatását. A szűrők legyenek 3-ad fokú Butterworth aluláteresztő szűrők. A törésponti frekvencia legyen 20, illetve 15 Hz. (Vegye figyelembe, hogy a HHMD-vel készült felvételek esetén a mintavételi frekvencia 1 kHz.) A kétféle szűrt jelet rajzolja egymásra kék, illetve piros színnel, nagyítson ki három szívciklust, alkalmazzon az ábrán jelmagyarázatot (2-3 jel ábrázolása esetén célszerű a piros, kék, fekete színeket használni, mivel ezek szürkeárnyaltos nyomtatás esetén is megkülönböztethetők).

Megoldás:

```
JA_jelei = hhmbinread('D:\hhmdl_recordings\ja_05_12_2017.hhm'); % fájl beolvasása
nyersECG = JA_jelei.ecg1; % az ecg1 nevű jel kiolvasása
N = 3; % a szűrő fokszámának eltárolása az N változóban
Fs = 1000; % a mintavételi frekvencia 1 kHz
W_20 = 20; % a 20 Hz-es szűrő határfrekvenciájának eltárolása a W_20 változóban
normalt_frekvencia_20 = W_20 / (Fs / 2); % a szűrő határfrekvenciáját normálni
kell (0 és 1 közötti érték kell, hogy legyen), ahol 1 a mintavételi frekvencia
felének felel meg
[b_20, a_20] = butter(N, normalt_frekvencia_20); % a szűrő fokszáma és normált
határfrekvenciája alapján a butter függvény előállítja a butterworth szűrő
számlálójának és nevezőjének együtthatóit
szurtECG_20 = filtfilt(b_20, a_20, nyersECG); % a megtervezett szűrő
számlálójának és nevezőjének együtthatói alapján a filtfilt nulla fázistolású
szűrést hajt végre
W_15 = 15; % a 15 Hz-es szűrő határfrekvenciájának eltárolása a W_15 változóban
normalt_frekvencia_15 = W_15 / (Fs / 2); % a határfrekvencia normálása
[b_15, a_15] = butter(N, normalt_frekvencia_15); % a butterworth szűrő számláló
és nevező együtthatóinak előállítása
szurtECG_15 = filtfilt(b_15, a_15, nyersECG); % a szűrés végrehajtása
figure(1); % új ábra, 1-gyes sorszámmal
plot(szurtECG_20, 'b'); % a 20 Hz-cel szűrt jel kirajzolása kék színnel
hold on; % az eddig kirajzolt jelalakok megtartása, amikor új jelet rajzolunk
plot(szurtECG_15, 'r'); % a 15 Hz-cel szűrt jel kirajzolása piros színnel
grid on; % rács rajzolása
legend('LPF - 20 Hz', 'LPF - 15 Hz'); % jelmagyarázat rajzolása. A vesszővel
elválasztott magyarázatok sorrendje meg kell, hogy egyezzen a jelek
kirajzolásának sorrendjével
title('A szűrés hatása az EKG-ra'); % az ábra címének beállítása
xlabel('Ido, ms'); % az x-tengely címének beállítása
ylabel('Amplitudo, a.u.');
```

Eredmény:

A szűrés hatása az EKG-ra



11. Készítsen egy Matlab script-et, amely beolvas egy, a HHMD-vel készült felvételt, majd a felvétel *ecg1* elnevezésű jelén digitális szűrést hajt végre. Hasonlítsa össze három különböző szűrő hatását. A szűrők legyenek 3-ad fokú Butterworth aluláteresztő szűrők. A törésponti frekvencia legyen 40, 30, illetve 15 Hz. (Vegye figyelembe, hogy a HHMD-vel készült felvételek esetén a mintavételi frekvencia 1 kHz.) A háromféle szűrt jelet rajzolja egymásra olyan módon, hogy fekete-fehér ábrázolás esetén is jól felismerhetők legyenek az egyes jelek. Nagyítson ki egy szívciklust, alkalmazzon az ábrán jelmagyarázatot. Emelje át az ábrát a Microsoft Office Word programba.

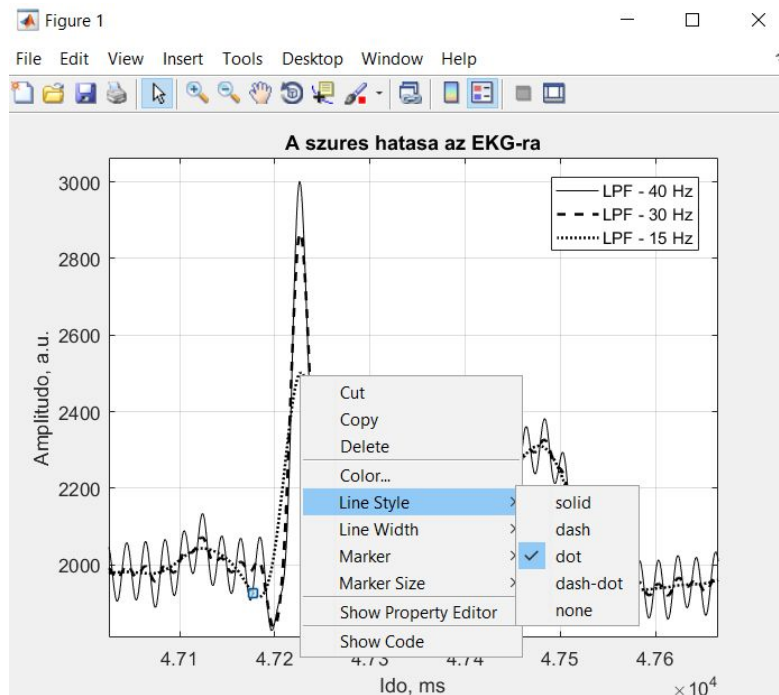
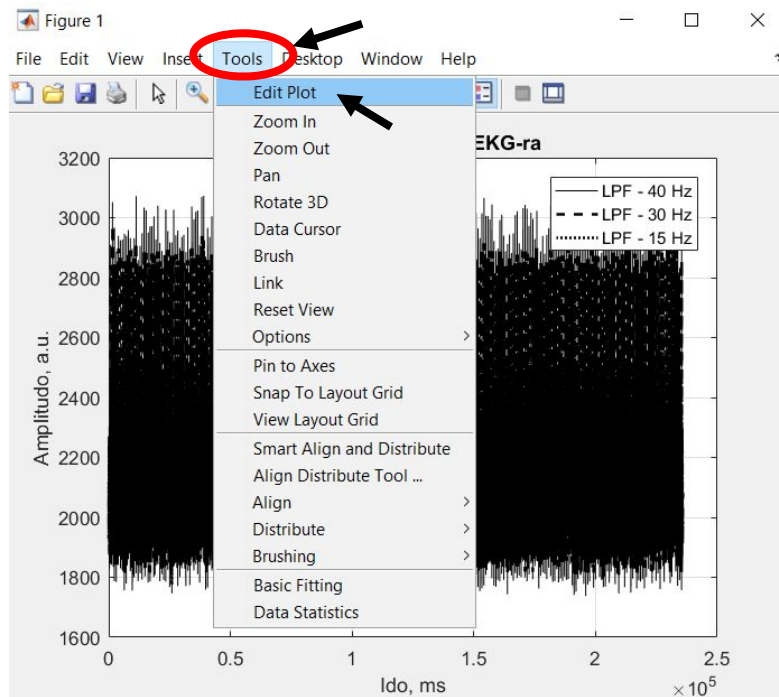
Megoldás:

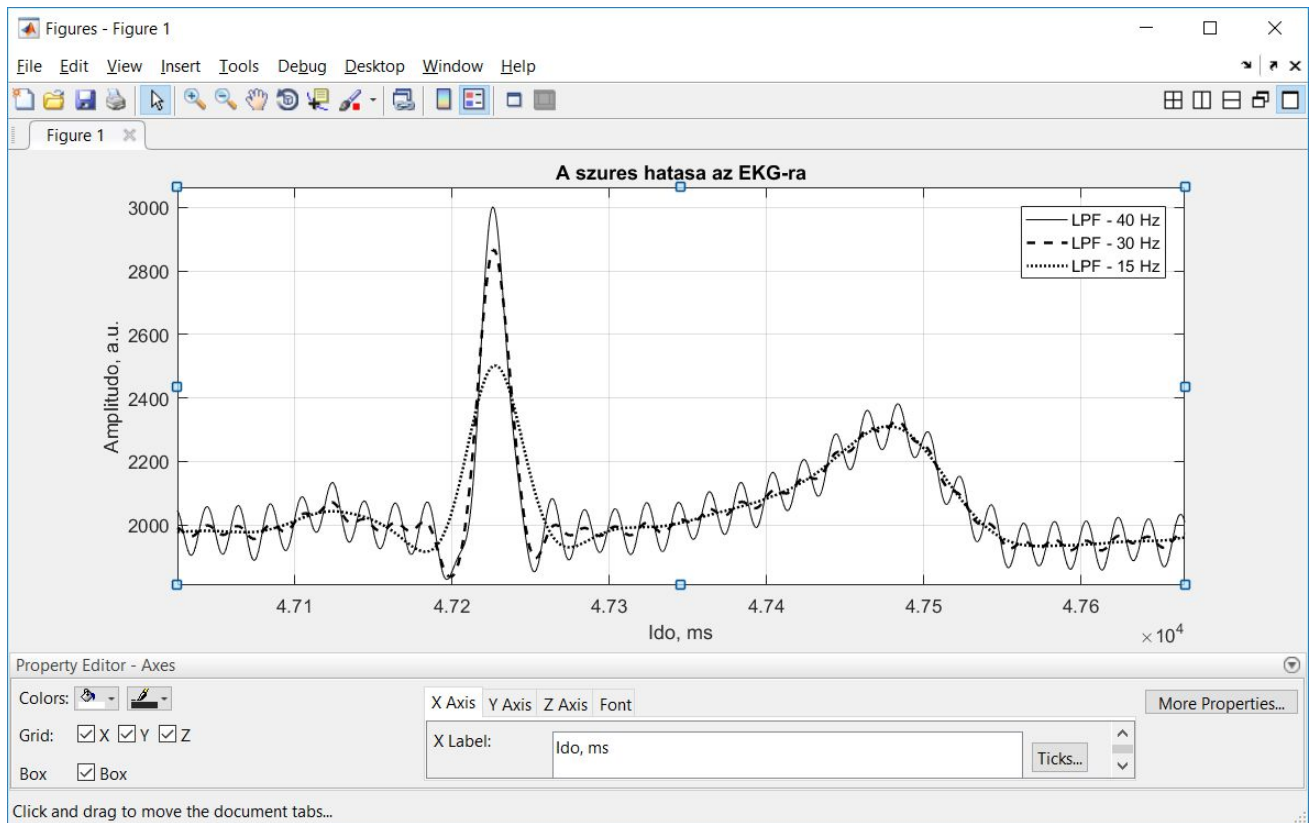
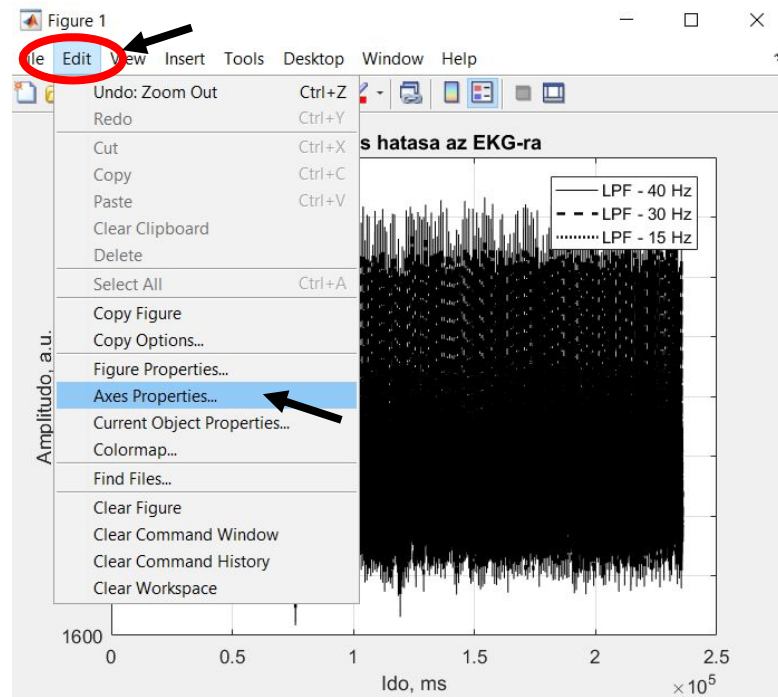
```
JA_jelei = hhmbinread('D:\hhmdl_recordings\ja_05_12_2017.hhm');
nyersECG = JA_jelei.ecg1;
N = 3;
Fs = 1000;
W_40 = 40;
normalt_frekvencia_40 = W_40 / (Fs / 2);
[b_40, a_40] = butter(N, normalt_frekvencia_40);
szurteCG_40 = filtfilt(b_40, a_40, nyersECG);
W_30 = 30;
normalt_frekvencia_30 = W_30 / (Fs / 2);
[b_30, a_30] = butter(N, normalt_frekvencia_30);
szurteCG_30 = filtfilt(b_30, a_30, nyersECG);
W_15 = 15;
normalt_frekvencia_15 = W_15 / (Fs / 2);
[b_15, a_15] = butter(N, normalt_frekvencia_15);
szurteCG_15 = filtfilt(b_15, a_15, nyersECG);
figure(1);
plot(szurteCG_40, 'k-'); % a 40 Hz-cel szűrt jel kirajzolása fekete folytonos
vonallal
hold on;
plot(szurteCG_30, 'k--', 'LineWidth', 1.5); % a 30 Hz-cel szűrt jel kirajzolása
fekete szaggatott vonallal, 1,5-ös vonalvastagsággal
plot(szurteCG_15, 'k:', 'LineWidth', 1.5); % a 15 Hz-cel szűrt jel kirajzolása
fekete pontozott vonallal, 1,5-ös vonalvastagsággal
grid on;
legend('LPF - 40 Hz', 'LPF - 30 Hz', 'LPF - 15 Hz');
title('A szures hatasa az EKG-ra');
xlabel('Ido, ms');
ylabel('Amplitudo, a.u.');
```

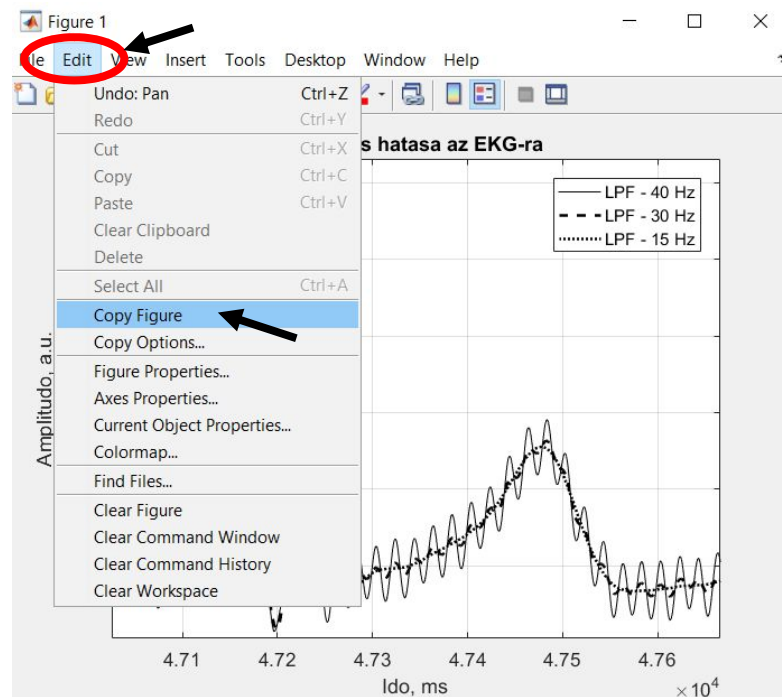
A Matlab lehetőséget ad az ábrák kézzel történő szerkesztésére is. A fenti példában beállított vonalszín, vonalstílus és vonalvastagság például beállítható az ábra eszköztárában (*Tools*) található *Edit Plot* eszközzel. Az *Edit Plot* kiválasztása után jobb-klikkelve a szerkeszteni kívánt jelre, felugrik a szín, stílus, vonalvastagság és további paraméterek beállítását lehetővé tevő menü.

Az ábra tengelyei az ábra menüszalagjának *Edit* elemében található *Axes Properties...* menüponttal szerkeszthetők. Beállítható a tengelyek neve, az ábrázolni kívánt tartomány, az ábrafelirat betűmérete és betűstílusa, a tengelyek osztási pontjai és további paraméterek.

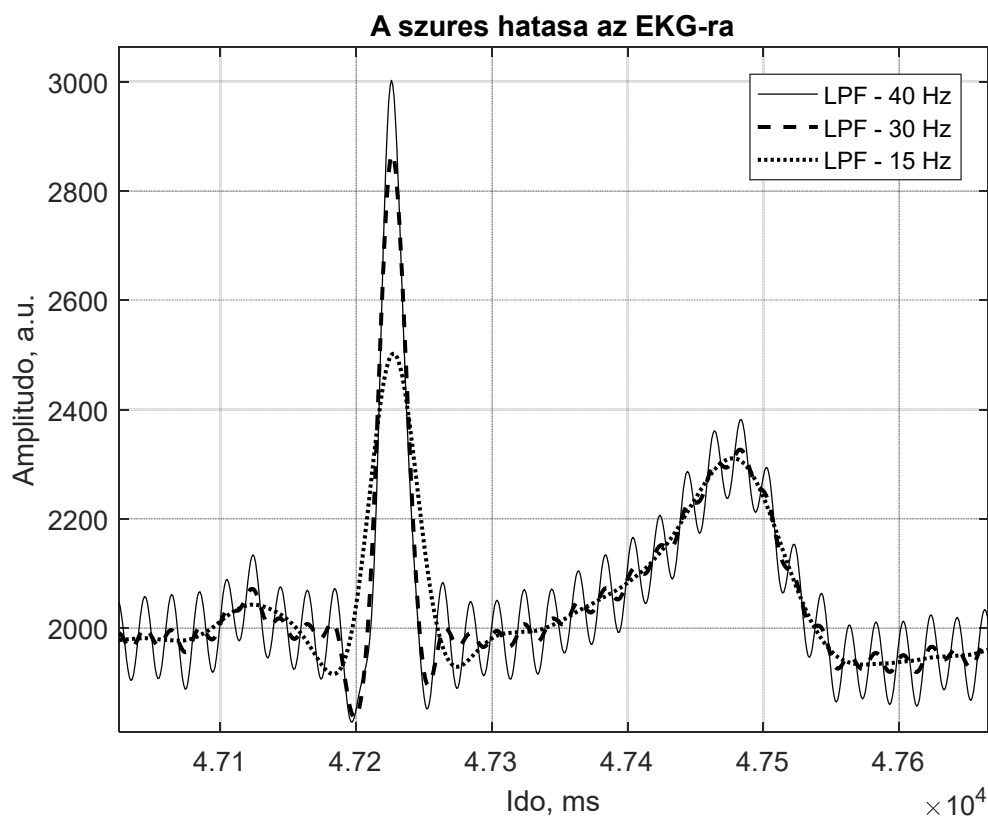
Az ábra átemelése a Microsoft Office Word programba legkönnyebben az ábra menüszalagjának *Edit* elemében található *Copy Figure* menüponttal tehető meg. Ezt követően a Word-ben egyszerű beillesztés parancs alkalmazása után megjelenik a másolt ábra.







Eredmény:



Önálló feladat: Hasonlítsa össze a fenti három szűrés eredményeként kapott jel spektrumát az eredeti nyers jel spektrumával.

Készítette: Nagy Péter, BME MIT

Észrevételeket a nagy@mit.bme.hu címen lehet jelezni.