

EÖTVÖS LORÁND TUDOMÁNYEGYETEM  
INFORMATIKAI KAR

# EKG jelek feldolgozása Hermite-függvények segítségével

BSc Szakdolgozat

Készítette: Dózsa Tamás  
ELTE IK  
Programtervező informatikus  
BSc

Témavezető: Dr. Kovács Péter  
Adjunktus  
ELTE IK  
Numerikus Analízis Tanszék



./Abrak/Egyeb/elte\_logo.jpg

Budapest, 2016.11.16.

# Tartalomjegyzék

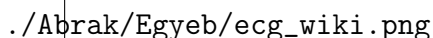
<b>1. Bevezetés</b>	<b>2</b>
1.1. A modell ismertetése . . . . .	2
1.2. Matematikai háttér . . . . .	3
1.2.1. Jelek approximációja . . . . .	3
1.3. Az approximáció optimalizálása . . . . .	4
<b>2. Felhasználói dokumentáció</b>	<b>5</b>
<b>3. Fejlesztői dokumentáció</b>	<b>6</b>

# 1. fejezet

## Bevezetés

### 1.1. A modell ismertetése

A modern orvostudományban nagy jelentőséggel bírnak a valamely élő szervezet által kibocsátott úgynevezett **biológiai jelek**. Ezek közé sorolható az **Elektro Kardio Gram**, vagy **EKG**, amely a szív állapotáról képes információt adni. Bár ennek a dologzatnak nem célja az EKG jelek pontos elemzése, fontos néhány sorban ismertetni egy átlagos EKG jel meghatározó hullámait. Egyetlen szívütés EKG reprezentációja három fő részre bontható: a szívütés elején megjelenő **P** hullámra, az ezt követő **QRS** komplexumra, és az ütés végén található **T** hullámra. Ezek rendre a pitvari összehúzódást, a kamrák depolarizációját és elektromos újratöltődését reprezentálják. Diagnosztikai szempontból a **QRS** komplexus a legfontosabb, ezért ezt nagy pontossággal kell tárolni. Általánosságban elmondható, hogy ezeknek a hullámoknak kezdő és végpontjai, valamint maximum és minimum értékei vesznek részt az orvosi diagnosztikában. Az említett paraméterek az 1.1 ábrán láthatóak.



./Abak/Egyeb/ecg\_wiki.png

1.1. ábra. Az EKG jel egy szívütése, illetve annak főbb diagnosztikai jellemzői.

Az irodalomban ismert tömörítő algoritmusokat [?] alapján három kategóriába sorolhatjuk: 1) egyszerű paraméteres becslések (pl.: interpoláció, különbségi kódolás, stb.), 2) direkt módszerek (pl.: csúcsok, meredekségek, stb. tárolása), 3) transzformációs eljárások. Az utóbbi osztály tartalmazza azokat az algoritmusokat, melyek a jelet egy előre adott függvényrendszer szerinti sorfejtéssel approximálják. Így az eredeti adatsorozat helyett csak az együtthatókat és a rendszer paramétereit kell tárolnunk. Ezen kategóriába sorolandó a dolgozatban bemutatott algoritmus is. Nevezetesen, az eredeti adatsorozatot speciális, Hermite-polinomok segítségével előállított függvényrendszerrel fogjuk közelíteni. A módszer alapját képező eljárás [?], jól ismert az irodalomban, mely nem csak a jelek tömörítéséhez, de azok modellezéséhez [?], illetve

osztályozásához [?, ?] is alkalmazható. A dolgozatban az EKG jelekkel való hasonlóságuk miatt Hermite-függvényeket használunk az adatok reprezentálásához. Ezeket egy argumentum transzformáción keresztül szabad paraméterekkel egészítjük ki. Ennek köszönhetően az eredeti jelet egy adaptív bázisban írhatjuk fel. Az említett paraméterek megválasztásához a Nelder-Mead optimalizációs eljárást alkalmaztuk. Mivel az EKG jelek diszkrét adatsorozatok, ezért a módszert [?] alapján implementáltuk diszkrét ortogonális Hermite-polinomokra is. A dolgozatban különböző tesztekkel demonstráljuk az algoritmus hatékonyságát. Ehhez, több órányi, zajjal terhelt, valódi EKG felvételt használtunk. Ezen keresztül a bemutatott módszert összehasonlítottuk több másik, az irodalomban jól ismert tömörítő algoritmussal is [?].

A tömörítő eljárást egy c++ nyelven megírt, objektum elvű alkalmazás implementálja, melyet egy webes felületen keresztül érhetünk el. A felület lehetőséget biztosít a dolgozatban jelölt tesztek újrafuttatására, valamint a teszteléskor felhasznált adatbázis további jeleinek a tömörítésére. Szintén a webes felületen keresztül nyílik alkamunk a már tömörített EKG jelek helyreállítására.

Az alkalmazás megejtvezésekor külön hangsúlyt kapott a kód újra felhasználhatósága. Ennek érdekében a felhasznált algoritmusok, illetve matematikai modellek a lehető legáltalánosabb formában lettek implementálva. Fontos szempontot jelentett továbbá a c++11-es nyelvszabvány által nyújtotta lehetőségek minél hatékonyabb kihasználása. Jó példa erre a lambda függvények alkalmazása az optimalizációs algoritmusok implementációja során. A hatékony működés mellett azonban a program igyekszik megfelelni a modern felhasználók igényeinek. Ennek érdekében a felhasználói felület weboldalként lett implementálva. A rendszerfüggetlen, és installáció mentes elérés lehetővé teszi a gyors és egyszerű használatot, valamint az eredmények megosztását.

## 1.2. Matematikai háttér

### 1.2.1. Jelek approximációja

EKG jelek feldolgozásakor sok esetben szembesülünk gyakorlati kihívásokkal. Két sűrűn előforduló példa a hosszú mérések tárolása, valamint a zajjal terhelt mérések ábrázolása. Ezekre a nehézségekre egyszerre ad kielégítő megoldást, ha a jeleket valamely  $\mathcal{H}$  Hilbert-tér sima függvényeiből álló  $(\Phi_n, n \in \mathbb{N})$  ortogonális bázisában reprezentáljuk és a jelet véges sok  $\Phi_0, \Phi_1, \dots, \Phi_n$  bázisbeli elem lineáris kombinációjával közelítjük. Az  $f \in \mathcal{H}$  jel legjobb közelítését a tér  $\|\cdot\|$  normájában az

$$S_n f := \sum_{k=0}^n \langle f, \Phi_k \rangle \Phi_k$$

leképezés nyújtja, ahol  $\langle \cdot, \cdot \rangle$  az  $\mathcal{H}$  tér skaláris szorzatát jelöli. A jel és a közelítés eltérésének négyzete a

$$\|f - S_n f\|^2 = \|f\|^2 - \sum_{k=0}^n |\langle f, \Phi_k \rangle|^2$$

képplettel adható meg. Adott hibán belüli közelítést véve a jel helyett elég az  $S_n f$  approximációt reprezentáló  $\langle f, \Phi_k \rangle$  ( $k = 0, 1, \dots, n$ ) Fourier-együtthatókat tárolni. Zajos jel esetén az ilyen típusú approximáció szűrőként is szolgál. A közelítés megvalósításához a klasszikus ortogonális rendszerek közül EKG görbék közelítésére az Hermite-féle függvények bizonyultak használhatónak. Ezt támasztják alá a [...] dolgozatok. Az Hermite függvények alkalmazása azzal is indikolható, hogy grafikonjuk hasonlít az EKG görbékre (lásd a Függelék).

### 1.3. Az approximáció optimalizálása

## 2. fejezet

### Felhasználói dokumentáció

## 3. fejezet

### Fejlesztői dokumentáció