

Kereszt-korrelációs módszerek alkalmazása gravitációs hullám-kitörések kutatásában

Bartos Imre, Raffai Péter

Témavezetők: Márka Szabolcs¹, Laura Cadonati²

¹California Institute of Technology

²Massachusetts Institute of Technology



A LIGO hanfordi és livingstoni detektorai (forrás: www.ligo.caltech.edu)

2004.

A LIGO (Laser Interferometer Gravitational Wave Observatory) egyik célja rövid, kitörésszerű gravitációs sugárzások észlelése. A Föld különböző pontjain elhelyezett detektorokból származó adatsorok kiértékelése kereszt-korrelációs módszerekkel történik, a korrelálatlan zaj eltávolítása és a korrelált jel felerősítése érdekében. Célunk egyrészt új, érzékenyebb módszerek kidolgozása volt, másrészt kellően fel kellett gyorsítanunk a programot ahhoz, hogy az a detektorokból érkező adatsort valós időben feldolgozhassa. Ennek érdekében az algoritmust két programnyelven párhuzamosan írtuk, így lehetővé vált a sebesség fokozása és a hatékonyabb fejlesztés egyaránt. A programcsomagot továbbá felkészítettük a feldolgozott adatok önálló kiértékelésére, így kimenetként az adott idősorhoz rendelt téves riasztási valószínűség (false alarm rate) jelenik meg, amely lényegesen megkönnyíti a kiértékelést. A programot volt alkalmunk a gyakorlatban is kipróbálni, például a detektorok körüli viharok zavaró hatásának vizsgálatával.

Tartalomjegyzék

1. Bevezetés	4
1.1. A gravitációs hullámokról általában	4
1.2. Gravitációs hullámok detektálása	4
1.3. Detektálás több interferométerrel	6
1.4. A filterezés	8
2. Kereszt-korrelációs módszerek	11
2.1. Egyszerű kereszt-korreláció	12
2.2. Korrelált felesleg (correlated excess)	13
2.3. Korrelációs együttható és konfidencia	15
3. Az algoritmus	16
3.1. A teszt statisztikák	16
3.2. Egész időtartam vizsgálata	17
3.3. Halmazok	18
3.4. Gamma illesztés	21
3.5. Események keresése (Event Scan)	22
3.6. Teszt statisztikák egyesítése	24
4. A program	25
4.1. Input/Output	25
4.2. Sebesség	28
5. Alkalmazások	29
6. Saját eredmények	30
7. Konklúzió	31
8. Kitekintés	31
9. Köszönetnyilvánítás	32

1. Bevezetés

1.1. A gravitációs hullámokról általában

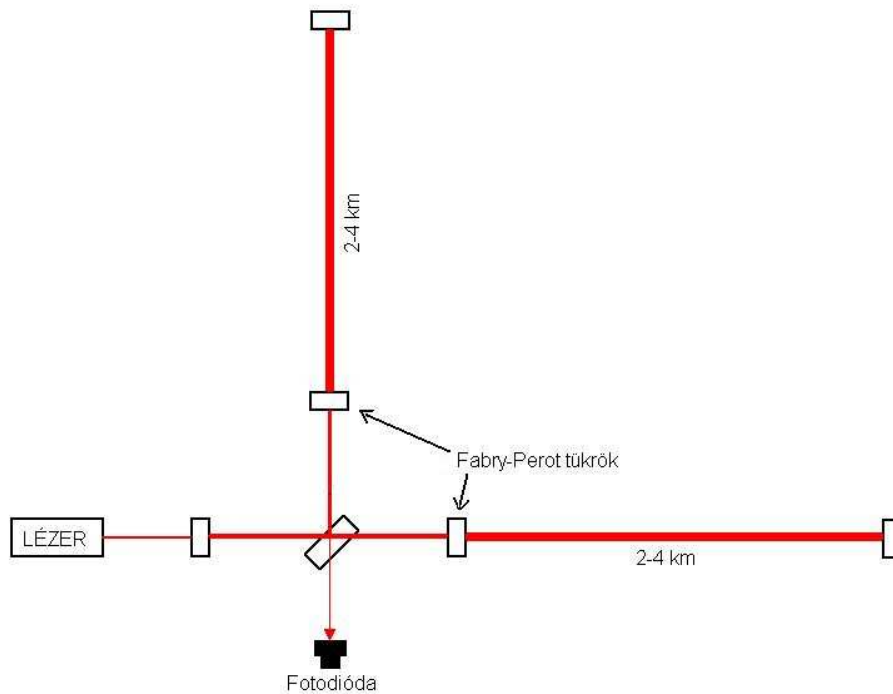
Az asztronómiai megfigyelések döntő többsége napjainkig elektromágneses sugárzás detektálásán alapult. A technikai fejlődés és a fizikai elméletek pontosabbá válása a XX. századra azonban lehetővé tette, hogy más természeti kölcsönhatásokat is vizsgáljunk csillagászati és extragalaktikus méretekben.

Az einsteini általános relativitáselmélet már évtizedekkel napjaink előtt megjósolta egy, az elektromágneses hullámokhoz hasonló, ám gravitációs kölcsönhatásból származtatható sugárzás létezését. Az elmélet szerint a "gravitációs hullámok" a téridő, forrásról leválni képes, fénysebességgel haladó torzulásai, melyeket gyorsuló tömeg-kvadrupól momentumok hoznak létre. Ilyen források lehetnek például szupernóva-robbanások, neutroncsillag és/vagy fekete lyuk ütközések, vagy akár erősen asszimmetrikus neutroncsillagok valamiféle szabálytalan mozgása. Fizikai hatásukat tekintve a gravitációs hullámok képesek a téridő azon pontjainak távolságát növelni ill. csökkenteni, melyeken áthaladnak. E hatás az, melyet (reményeink szerint) a létezésük kimutatására fel tudunk használni. Közvetlen detektálásuk ugyanis a mai napig várat magára, jóllehet létezésükre vonatkozó közvetett bizonyítékok felmutatásáért már Nobel-díj is került kiosztásra. Kimutatásuk azért is bír különleges jelentőséggel, mert várakozásaink szerint mérésükkel egy, az elektromágneses spektrumtól független információhordozó közeg kerülne a birtokunkba, mely nemcsak a gravitációs kölcsönhatás természetére vonatkozó elméletek pontosítását tenné lehetővé, de az űrkutatásban és a korai Univerzum történetének kutatásában is értékes, új adatokkal szolgálna.

1.2. Gravitációs hullámok detektálása

A gravitációs hullámok detektálásában rejlő kutatási lehetőségek kiaknázására a Föld számos pontján épültek detektorok. Ezek lényegében óriási Michelson-interferométerek¹, melyek a hullámok téridő torzító hatásait teszik több lépésben észlelhetővé. E detektorok működési elve mindenhol ugyanaz: két, egymásra merőleges, koherens lézernyalábot interferáltatunk úgy, hogy találkozási pontjukban kioltás valósuljon meg (kioltási interferencia), majd fotodiódákkal a találkozásipont-beli fényintenzitást mérjük meghatározott mintavételi frekvenciával. Gravitációs hullám-jelnek a berendezésen történő áthaladása esetén, az általa okozott téridő torzulás miatt az interferenciakép

¹az interferométer valójában Fabry-Perot tükrökkel is ki van egészítve, hogy a lézernyalábok többszöri visszareflektálásával még hosszabb effektív interferométer-kart érjenek el, s a berendezés érzékenységét ezáltal még jobban megnöveljék.



1. ábra. A *LIGO* interferométer-típusú gravitációshullám-detektorainak szemantikuss vázolata

megváltozik, mely a mintavétel során kapott digitális jelsorozatban is nyomot hagy.

Azon jelek forrásait, melyek időben végesek, gravitációs hullám-kitöréseknek nevezzük. Bár az elnevezés kissé megtévesztő, hisz magába foglal minden véges hosszúságú jelet konkrét időtartamától függetlenül, eredete leginkább technikai okokra vezethető vissza. A legtöbb, jelek háttérből való kibányászását végző algoritmus használhatóságának feltétele ugyanis, hogy a keresett jel időben korlátos legyen (legalábbis az idősoron belül). Másrészt a kifejezéssel szeretnénk elkülöníteni a jól lokalizálható asztronómiai forrásokból eredő jeleket az Ősrobbanásból esetlegesen fennmaradt, állandó jelleggel jelenlévő gravitációs háttérsugárzástól.

A detektálási folyamat technikai megvalósítása korántsem egyszerű. Az einsteini elmélet szerint ugyanis a gravitációs hullámok a négydimenziós téridőben relatív elmozdulásokat képesek okozni, méghozzá - forrástól függően - igen kis mértékben (ez valójában a gravitációs kölcsönhatás igen gyenge jellegében is megnyilvánul - a gravitációs hullámok tehát az anyaggal csupán

gyengén hatnak kölcsön). Adott

$$\epsilon = \Delta l/l \quad (1)$$

hullámamplitúdó esetén, hogy a távolságváltozás abszolút értékét (Δl) mérhető nagyságúvá tegyük, szükséges, hogy az interferométer karjainak hosszát (1) a technikai és gazdaságossági korlátok figyelembevételével a lehető legnagyobbra válasszuk. Az Egyesült Államok-beli LIGO (Laser Interferometer Gravitational Wave Observatory) ezért két 4 kilométer (Hanford, WA; Livingston, LA), valamint egy 2 kilométeres (Hanford, WA) karhosszúságú interferométert használ. Ezek, méretüket tekintve jelenleg nemcsak a világ más tájain hasonló célból épült detektorokat haladják meg, de minden idők legérzékenyebb interferométereinek is minősülnek. Az abszolút távolságváltozás ugyanis, melyet a detektorokkal elméletileg még ki tudunk mérni, mindössze 10^{-18} méter, mely nagyságrendjét tekintve egy proton méretének ezredrésze.

A detektálható jelek kicsiny volta, és a műszer érzékenysége frekvenciatartománytól függően igen kicsiny jel-zaj viszonyt is eredményezhet. Ennek következtében nemcsak a felhasznált berendezések zajszintjét kell különféle fizikai beavatkozásokkal csökkenteni, de a kapott adatsorok elemzésére is minél érzékenyebb feldolgozó eljárásokat kell kidolgozni. Ez a mérnökök és fizikusok számára komoly technikai kihívást jelent.

Munkánk során behatóbban a szoftveres fejlesztés lehetőségeivel foglalkoztunk. Célunk az volt, hogy a zajjal terhelt adatsorok feldolgozásában a már ismert és alkalmazott kereszt-korrelációs módszerek hatékonyságát és érzékenységét növeljük, az adatsorok feldolgozásának pedig minél több mozzanatát automatizáljuk számítógépes program megírása által. A munka során fontos szempont volt, hogy az elkészíteni kívánt program futási idejét oly mértékben lecsökkentsük, hogy a feldolgozás az adatgyűjtéssel egyidőben, adatfeltorlódás nélkül megtörténhessen.

Végző soron tehát egy, a gravitációs hullámok mérés technikai gyakorlatában praktikus alkalmazható eszközt kívántunk létrehozni a matematika, a jelfeldolgozás és a számítástechnika módszereinek segítségével.

1.3. Detektálás több interferométerrel

Gravitációs hullámok detektálására a Föld több pontján is történnek próbálkozások a már tárgyalt módszerrel². Az egyes, térben szeparált interferométerek adatsorainak összevetéséből éppen ezért újabb információkat nyerhetünk a különálló jelsorozatok egyéni vizsgálatához képest. A különböző adatsorokban ugyanis a zaj, forrásának lokalizáltsága miatt - várakozásaink szerint

²GEO 600, Németország; VIRGO, Olaszország; TAMA, Japán

- egymással korrelálatlan lesz, szemben a távoli forrásból érkező gravitációs hullám-jellel³. A jelek korreláltságát nemcsak a közös, távoli forrás biztosítja, de a hullámok gyenge kölcsönhatása is az anyaggal, aminek következtében a jelek kvalitatív jellemzői (jelalak, jelhossz, stb.) a detektorok között megtett távolság alatt nem változnak (kimutathatóan).

Tudjuk azonban, hogy az interferométerek közötti véges távolság megtétele miatt a jelek detektálási időpontja különböző detektorok adatsoraiban eltér. Ez, ismert jelforrás esetén - a gravitációs hullámok terjedési sebességének ismeretében - kompenzálható, ismeretlen forrásból jövő jelek esetén pedig a beérkezési időpontok alapján a forrás éggömbön való elhelyezkedése bizonyos hibán belül meghatározható. A helymeghatározás hibája nyilvánvalóan függ a rendelkezésre álló adatsorok-, azaz végső soron a Föld különböző pontjain található detektorok számától. Minél több detektor áll ugyanis rendelkezésünkre, a forrás lokalizálása annál pontosabban végezhető el.

Mivel a gravitációs hullámok nyilvánvalóan nem terjedhetnek a vákuumbeli fény sebességénél gyorsabban, az egymással korreláló jelek közötti maximális detektálásiidő-különbségre a detektorok távolságának ismeretében felső becslést adhatunk (ez az időérték a LIGO Washington-, valamint Louisiana állambeli detektorára mintegy 10 msec). Két adatsor kereszt-korrelációs módszerekkel történő összehasonlítása tehát ezen időkorlátokon túl értelmet vesztí, hiszen korrelációt ilyen esetben semmiképp nem várhatunk. Miután a világ minden táján az adatok tárolása meghatározott időintervallumokra feldarabolva történik, az előbbi feltétel technikailag azt jelenti, hogy a maximális korrelációs időértéket összehasonlítandó mintafájl-darabszámmá tudjuk átkonvertálni. Ez a későbbi vizsgálatokat nagyban leegyszerűsíti.

Több, különböző detektor alkalmazása a gravitációshullám-kutatásban más előnnyel is jár, melyre a LIGO két, Hanfordban (Washington állam, USA), egy épületen belül található, ám egymástól függetlenül működő interferométere ad konkrét példát. Az interferométerek lézernyalábjai közös vákuumcsőben futnak. Hosszuk azonban különbözik: az egyik interferométer karjai 4-, míg a másikéi 2 kilométer hosszúak. Mivel az interferométerek adatsoraiban megjelenő zaj nagysága (többé-kevésbé) független a karok hosszától, míg a gravitációs hullám-jel amplitúdója azzal egyenesen arányos, jel észlelése esetén könnyen megállapíthatjuk, hogy az valós gravitációs hullám-forrásból ered-e. Ebben az esetben ugyanis a két detektor által rögzített jel alakjának meg kell egyeznie, amplitúdóját tekintve viszont a rövidebb karú műszerének fele akkorának kell lennie, mint a hosszabb karúnak.

³a Big Bang-ből visszamaradt kozmikus gravitációs-sugárzási háttér - ha létezik ilyen - valójában korrelált "zajként" jelenik meg az adatsorokban, ám ennek kiszűrési lehetőségeivel most nem foglalkozunk



2. ábra. A LIGO detektorainak elhelyezkedése a Földgömbön (forrás: www.ligo.caltech.edu/docs/G/G030024-00.pdf)

A fent említetteken túl a több detektoros mérések előnye, hogy az egyes berendezések érzékenysége különböző frekvenciákra eltérhet. A gravitációs hullám-spektrum kiterjedtebb lefedése pedig a lehetséges források számára enged meg nagyobb korlátot, ami ezáltal a jel-észlelés valószínűségét növeli meg. Az idei évben előkészítési szakaszba lépett LISA (Laser Interferometer Space Antenna) project, mely során egy, a már vázolt elven működő gravitációshullám-detektort kívánnak az űrbe - zajtól kevésbé terhelt környezetbe - telepíteni, nagy előnye tehát nemcsak abban áll, hogy hatósugara lesz minden eddiginél nagyobb a hosszabb interferométer karok miatt, de spektrális érzékenysége is különbözni fog a jelenleg használt detektorokétól.

1.4. A filterezés

A kereszt-korrelációs vizsgálatok napjaink legérzékenyebb jelfeldolgozó eljárásainak számítanak. Alkalmazásukkal jelkeresési szempontból rossz jel-zaj arány mellett is nagy hatásfokkal végezhető el a korrelált jelek kiemelése a zajos háttérből. E módszerek hatékonysága azonban az adatsorok előzetes kezelésével tovább fokozható. Jóllehet munkánk során kizárólag már előzőleg "tisztított" adatfájlokat használtunk, mégis szeretnénk rövid, kvalitatív jel-

legű betekintést nyújtani a kereszt-korrelációs analízist megelőző feldolgozási mechanizmusokba is.

Az interferométerekből a fotodiódakon és jeltovábbító apparátuson keresztül bedigitalizált jelek egy, a detektorhoz kapcsolódó számítógépes háttértárban "nyers" adatfájlként kerülnek tárolásra. Ahhoz, hogy a detektálás körülményeit később is rekonstruálni tudjuk, valamint a jelkeresési vizsgálatokat elvégezzük, nemcsak maguk a mért intenzitásértékek, de a detektálás egyéb paraméterei (mintavételi frekvencia, az adatot rögzítő detektor paraméterei, a detektálási idő, ismert forrás esetén annak adatai, stb.) is rögzítésre kerülnek az elektronikus adatbázisban. Ezek egy jelentős része mind nyújt bizonyos többletinformációt ahhoz, hogy a tartományt, amelyre a korrelációs analízist elvégezzük, racionális keretek közé szűkítsük le. Az ilyen eljárásokat összefoglaló néven szűrésnek, filterezésnek nevezzük.

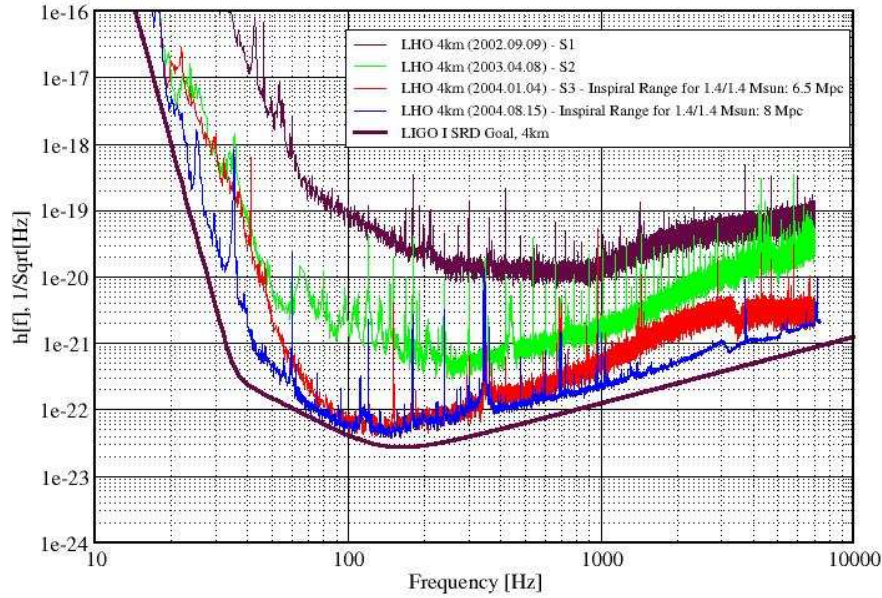
Ilyen adatszűrési eljáráshoz jutunk, ha a nyers adatokat a detektorunk érzékenységi görbéjének megfelelően módosítjuk. A gravitációshullámdetektorok közös jellemzője, hogy a beérkező jelek frekvenciájának függvényében érzékenységük más és más. Ennek leginkább az egyes frekvenciatartományokban megjelenő zajkomponensek eltérő intenzitása az oka. Mivel a mintavétel forrásának számító interferenciakép mechanikus rezgésekre éppúgy megváltozik, mint gravitációs hullámjel beérkezése esetén, a detektor környéki talaj szeizmikus mozgása az alacsony frekvenciás tartományban jelentős mértékben bezavar. Igen alacsony frekvenciákra (kb. 20 Hz alatt) az ilyen jellegű forrásból eredő zaj oly mértékű, hogy az extraterresztialis forrásból jövő - valós - jelek észlelése elhanyagolhatóan kis valószínűséggel következhet csak be. A spektrum alacsonyfrekvenciás tartományának levágásával tehát többet nyerünk, lévén, hogy ezzel az e tartományba eső zajt teljesen le tudjuk választani.

Hasonló jelenséggel állunk szemben közép- (néhány száz Hertz) és nagyfrekvenciás tartományban (ezer Hertz és afölött) is. Az előbbiben a jelészlelést a detektor optikai berendezéseinek hőmozgása nehezíti, míg utóbbiban a fény kvantumosságának következményeként a beérkező fotonok fázisbizonytalansága okoz frekvenciával logaritmikus skálán arányosan növekvő zajtényezőt (shot noise). Az egyes zajkomponensek dominanciája között a frekvenciatérben természetesen folytonos az átmenet.

Az említett effektusok zavaró hatásainak mértéke elméleti úton megbecsülhető. Az így kapott detektor-érzékenységi görbe figyelembevételével aztán az adatsor frekvenciaspektruma igény szerint módosítható (pl. az alacsony frekvenciás komponensek teljes levágása, vagy a spektrum egyes tartományainak megfelelő faktorokkal való súlyozása, stb.), s ezáltal a zajszint jelentősen csökkenthető.

Ismert forrás esetén azonban a vizsgálati tartományra további szűkíté-

Strain Sensitivities for the LIGO Interferometers
 H1 Performance Comparison: S1 through post S3 LIGO-G040439-00-E



3. ábra. A LIGO hanfordi 4 km-es detektorának spektrális érzékenységgörbéje. A különböző színek a detektor különböző adatgyűjtési időszakainak (science run) felelnek meg, míg a legalsó, sima görbe az elmélet szerint elérhető legalsó korlát a detektor érzékenységére. Jól látható, ahogy az egyes működési ciklusok között végzett fejlesztési munkálatok a berendezés valós érzékenységének az elméleti határ felé közeledésében valóban megtérültek. (forrás: www.ligo.caltech.edu/docs/G/G030024-00.pdf)

seket tehetünk. Mivel a gravitációs hullámok forrásai a gyorsuló tömegkvadrupól momentumok, források lehetnek például egymás körül keringő nagy tömegű (neutron)csillagok vagy fekete lyukak. A közös tömegközéppont körüli keringésük során gravitációs hullámok formájában kisugárzott energia révén távolságuk folyamatosan csökken, miközben mozgásuk felgyorsul. A távolság csökkenésével a kettős rendszer kvadrupólmomentuma megnő, ezért az általa keltett gravitációs hullám amplitúdója is folyamatosan nő. A gyorsuló keringés pedig a hullám frekvenciájának növekedését eredményezi. A folyamat addig folytatódik, míg az objektumok végül össze nem ütköznek (ezen ütközési folyamat is generál gravitációs hullámokat, de követhetlensége miatt a jelalak elméleti úton nem meghatározható), és a keletkező új objektum végül valami stacionárius mozgást nem vesz fel. A bespiráló kettős rendszer által keltett gravitációs hullám jelalakja a rendszert alkotó objektumok tömegének és pályájának ismeretében pontosan, ezen információk hiányában pedig becslés-szerűen megadható⁴.

Ha a kereszt-korrelációra alapozott jelkeresési eljárásunk egyik bemenetének egy valós adatsort, másikának pedig az elméleti úton kapott görbét választjuk, az eljárással az elméleti görbéhez hasonló jeleket nagyobb hatáskkal tudjuk az adatsorban megkeresni. Emellett csupán a frekvencia behatárolásával végezhetünk az adatsor frekvenciaspektrumában olyan szűrést, mely a keresett tartományt jobban kiemeli.

Az előbbieken említett szűrési eljárások persze csupán tipikus példái a kereszt-korrelációs analízist megelőző feldolgozási módszereknek, hasonló mechanizmusoknak ugyanis szinte korlátlan fajtája ismert.

Miután a kereszt-korrelációs analízist megelőző eljárásokat áttekintettük, s ezzel a vizsgálatainkat végző matematikai processzus bemenetét is tisztáztuk, rátérhetünk az általunk használt korrelációs algoritmus tárgyalására.

2. Kereszt-korrelációs módszerek

Mint ahogy azt a bevezetésben láttuk, a gravitációs hullámok érzékelésében tehát az érzékenység jelentős növekedését érhetjük el, ha nem egy, hanem több detektort használunk, és az adatsorok keresztkorreláltatásával felerősítjük a korrelált jelet a korrelálatlan zajjal szemben. Az ilyen módszerek lényege, hogy különböző detektorok adatsorából két, azonos hosszúságú darab produktumával jellemezzük a korreláció mértékét.

⁴az ilyen típusú jelek hangfrekvenciás tartományban történő lejátszása emelkedő hangmagasságú, erősödő, "füttyszerű" jelet eredményez, melyet emiatt szakmai körökben "chirp"-nek is neveznek

A megfelelő darabok kiválasztásának három fő paramétere van. Minde-
nekelőtt meg kell adnunk, hogy milyen időpontban keressük a korrelációt,
azaz milyen időpontot feltételezünk a jel megérkezésének. Másrészt a beve-
ető részben tárgyaltaknak megfelelően a jel - amennyiben nem a két detektor
síkjára merőlegesen érkezik be - nem egyszerre fogja elérni a detektorokat, így
meg kell adnunk valamilyen időeltolást azok adatsorai között. A harmadik
paraméter az érkező jel hosszára jellemző, azaz azt mondhatjuk meg, hogy
milyen hosszú adatdarabra integrálunk. Nyilván fontos, hogy a korrelált jel
minél nagyobb hányada belekerüljön ebbe az intervallumba, hogy a korreláció
miatt felerősödjön a hatása a zajhoz képest, viszont nem jó, ha az integrá-
lási hossz túlnyúlik a jelen, mivel ekkor a nagyobb mennyiségű zaj könnyen
elmoshatja a jel hatását.

A felsorolt szempontok figyelembevételével munkánk során három külön-
böző eljárást használtunk a korreláció megállapítására a felsorolt paraméte-
rekkel. Ezek a *teszt statisztikák*, amelyek a következőkben kerülnek ismerte-
tésre.

2.1. Egyszerű kereszt-korreláció

A legegyszerűbb felhasználása a keresztkorrelációval nyert adatsornak nyilván
az, ha a két adatsor megfelelő produktumát használjuk a korreláció mérté-
kének jellemzésére. Vesszünk tehát két adott hosszúságú szakaszt a két adat-
sorból, és azokat egyszerűen pontonként összeszorozzuk.

A keresztkorreláció az eddigiek alapján egy háromdimenziós tér egy pont-
ján történik. Vizsgálataink azt mutatták, hogy nem érdemes három dimen-
zióban maradni, sokkal kifizetődőbb az időeltolás paraméterre integrálni. Ez
azért lehet hatékonyabb, mert a véges kiterjedés miatt a jelek több kisebb
időeltolás esetén még részben *bent maradnak* az integrálandó tartományban,
ott nagyobb mértékű korrelációt eredményezve, ami tovább erősítheti a kor-
relálatlan zaj kiszűrését. Ugyancsak fontos szerepe lehet a jel felerősítésében,
ha a jel hosszabb, mint az integrálási tartomány. Ekkor az eltolások közül
több is teljes egészében rajta maradhat a jelen, erősítve az integrálás utáni
korrelációt az integrálás nélkülihez képest. Vizsgálataink szerint a használt
próbajelekre ez az integrálás bizonyos esetekben néhányszorosára is növelheti
az érzékenységet.

A három paraméterből tehát az időeltolást (\pm max időeltolásig tartó integ-
rálással) kiküszöböltük. Mostantól tehát már csak a maradék két dimenziós
teret kell végigpásztáznunk.

Fontos megjegyezni, hogy az integráláskor el kell tárolnunk az integrált
elemek közötti maximális korrelációhoz tartozó elemhez tartozó időeltolást,
a későbbiekben ez nyújthat ugyanis információt a jel beérkezési irányáról.

Az integrálással együtt tehát az első keresztkorrelációs módszer a következőképpen néz ki. Legyen a két adatsorunk A és B . A -n kijelölünk egy k hosszúságú szakaszt, ez lesz az eljárás magja (core). B -ből veszünk egy n hosszúságú szakaszt, amelyet integrálni fogunk (n =integrációs hossz). Ez után ezen szakasz minden pontjának képezzük a prodiktumát a mag minden egyes pontjával majd az eredményeket összeadjuk. mindeközben megjegyezzük, hogy hol volt a legnagyobb mértékű a korreláció, így közvetve információt tárolva a beérkezett gravitációs hullám forrásának irányáról. Az eljárás formalizálva tehát a következő:

```

> eltolás = 0
> összkorreláció = 0
> for i = 1 to mag mérete
>   eltolás ++
>   korreláció = 0
>   for j = 1 to integrációs hossz
>     korreláció += A(hely + j) · B(hely + j + eltolás)
>   end
>   összkorreláció += korreláció
>   if korreláció > max
>     save itt volt a maximum
>   endif
> end

```

Itt a *hely* azt az időpontot jelöli, aminél a mag kezdődik.

A fenti algoritmus csak vázlat, ehhez képest a programban a két adatsorra szimmetrikusan kiszámoljuk mindkettő eltolásával a korrelációkat, és ezek négyzetes összegével számolunk.

Megjegyzendő, hogy ez az első módszer egyszerű, és nem annyira érzékeny mint az egyéb, bonyolultabb korrelációs módszerek, mindazonáltal hasznosnak bizonyulhat az egyéb módszerekkel vett kombinációban.

2.2. Korrelált felesleg (correlated excess)

Az előző módszer esetén egy mérésből nem nyerünk információt arról, hogy a mért korreláció milyen mértékben származik a zajból és milyen mértékben az esetleges jelből. Az is elképzelhető továbbá, hogy a háttér karakterisztikája az idővel valamilyen módon, ha nem is túl gyorsan, de változik, így célszerűnek tűnik egy adott méréshez mintavételezni külön a háttérrel is. Ezt végzi el ez a módszer.

Első lépésben az előzővel megegyező módon elvégezzük a keresztkorrelá-

ciót egy központi tartományra, a feltételezett jel környékén. Ezután a két adatsort a már korábban tárgyalt korrelációs időkorláton túl (jelen esetben $10msec$) nyúló időeltolással is keresztkorreláltatjuk egy egy tartományon, pozitív és negatív eltolásra is, ezek lesznek az oldalsó tartományok (lobe), ezekkel mintavételezzük a vizsgált jel környezetét. A korreláció számítása itt sem egy, hanem több pontra történik, tehát különböző eltolásokkal vett keresztkorrelációkat összegzünk. Az ilyen nagy eltolás melletti keresztkorrelációval aztán normáljuk a középső részt, így mintegy figyelembe véve a háttérét. A normálás az átlag levonását, majd a szórással való osztást jelenti, ennek megfelelően eredményül a háttérhez képesti felesleget kapjuk, ami az esetleges jelnek felel meg. Az algoritmus tehát a következőképpen néz ki:

```

> for  $i \in core$ 
>   eltolás - annak megfelelően, hogy épp hol tart a ciklus
>   korreláció = 0
>   for  $j = 1$  to integrációs hossz
>     korreláció +=  $A(hely + j) \cdot B(hely + j + eltolás)$ 
>   end
>   corekorreláció( $i$ ) = korreláció
>   if korreláció > max
>     save itt volt a maximum
>   endif
> end
>
> for  $i \in bal\ lobe; jobb\ lobe$ 
>   eltolás - annak megfelelően, hogy épp hol tart a ciklus
>   korreláció = 0
>   for  $j = 1$  to integrációs hossz
>     korreláció +=  $A(hely + j) \cdot B(hely + j + eltolás)$ 
>   end
>   lobekorreláció( $i$ ) = korreláció
> end
>
> Átlag = 0
> STD = 0
> for  $i = 1$  to hossz(lobekorreláció)
>   Átlag += lobekorreláció( $i$ ) / hossz(lobekorreláció)
> end
> for  $i = 1$  to hossz(lobekorreláció)
>   STD += (lobekorreláció( $i$ ) - Átlag)2

```

```

> end
>  $STD = \sqrt{STD} / (2 * \text{hossz}(\text{lobekorrelláció}) - 1) >$ 
> for i = 1 to hossz(corekorrelláció)
>   corekorrelláció(i) = (corekorrelláció(i) - Átlag) / STD
> CC += corekorrelláció(i)
> end

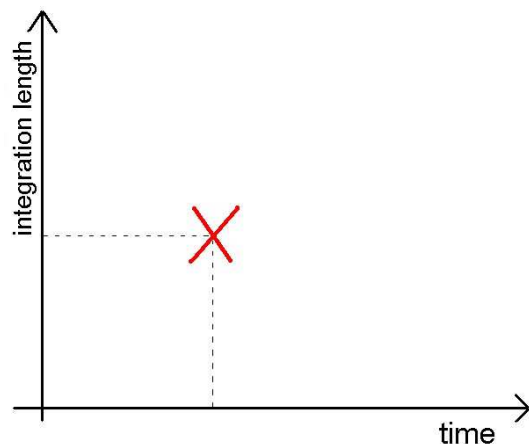
```

A kimenet, a korrelláció jellemzése tehát a CC változó lesz. A for $i \in$ bal lobe; jobb lobe azt jelenti, hogy a ciklusban i a bal, ill. jobb oldalsó tartomány elemeinek indexein fut végig. Az előzőhöz hasonlóan ez a leírás is vázlatos. A programban a két adatsort szimmetrikusan toljuk el, és az eredményeket négyzetesen összeadjuk (tehát a fenti algoritmust lefuttatjuk így, és A,B felcserélésével, és a végső eredmény a két részeredmény négyzetösszegének gyöke). Különbség még, hogy a programban több lobe rész is megadható különböző távolságokkal a középponttól és méretekkel. Egyéb különbségek is vannak, ezek azonban nem befolyásolják lényegében az eljárást.

2.3. Korrellációs együttható és konfidencia

A harmadik teszt statisztika két részből tevődik össze. A korrellációs együttható (Correlation Coefficient) egyszerre a két adatsor megfelelő részének korrellációját mondja meg, azaz egy -1 és 1 közötti számot ad vissza. A konfidencia egy statisztikus teszt, ami azt hivatott eldönteni, hogy a mintánk befolyásolva van-e valamilyen jel által. Ennek eldöntésére a Kolmogorov-Smirnoff tesztet használjuk. Ez a mért pontonkénti keresztkorrelláció eloszlását hasonlítja össze a normális eloszlással, kumulatív eloszlásokat használva. A normális eloszlástól vett eltérés jellemzésére a kettejük között levő legnagyobb különbséget használja. Ezt a különbséget megmérve kiszámítja annak valószínűségét, hogy a két eloszlás azonos. Ezen módszer használatához fel kell tennünk, hogy a háttér pontonkénti keresztkorrellációjának eloszlása közel esik a normális eloszláshoz, valamint, hogy elegendő pontot tudunk felhasználni egy-egy statisztika elkészítéséhez. Ezek a feltételek a mérések szerint a teszt megkövetelt pontosságával, vagyis robusztusságával összehasonlítva teljesülnek.

A korrellációs együtthatót és a konfidenciát egymással összeszorozva használhatjuk, azaz azokat a korrellációs értékeket használjuk csak, amelyek helyén a korrelláció átment a konfidencia teszten. A programban ez az összeszorozás még nincs teljesen megoldva, egyelőre külön számolunk a két statisztikával.



4. ábra. *A teszt-statisztikákkal egyszerre egy pontot mérhetünk meg az időpont-integrációs hossz síkon.*

3. Az algoritmus

Rendelkezésünkre áll tehát 3 különböző eljárás, amelyekkel az adatsorokból megkaphatjuk, hogy egy adott időpontban, adott időeltolásokkal (azaz adott irányhalmazból) és adott integrálási hosszal milyen mértékű a korreláció. Egy ilyen pontot mutat a 10. ábra.

Vizsgálatunk azonban legtöbbször nem korlátozódhat egy adott időpontra és irányra, mivel legtöbbször nem egy konkrét, ismert forrás gravitációs hullámaint akarjuk megtalálni, hanem egy nagyobb időintervallumban kutatunk jelek után (fontos megjegyezni, hogy bizonyos esetekben ilyen, célzott vizsgálat is történhet, néhány gammakitörés esetén például feltételezzük, hogy a gammakibocsátással párhuzamosan gravitációs hős keletkeztek. Ekkor nyilván mind az irányt, mind az időpontot ismerjük), az detektorok futtatásának (Science Run) egész időtartamát végig szeretnénk nézni. Az egész időintervallum végigpáztázása abból a szempontból is kulcsfontosságú lehet, hogy ezáltal információhoz juthatunk a keresztkorrelációval kapott adatsor eloszlásáról, azaz felmérhetjük a háttérrel, ami létfontosságú a jelek kiszűrése szempontjából. Nézzük meg tehát, hogy hogyan épül fel a program a felsorolt feladatoknak megfelelően.

3.1. A teszt statisztikák

Azt mondhatjuk, hogy a teszt statisztikák jól működnek az adott paraméterter (időpont, időeltolás, integrációs hossz) egy egy pontjához, így a nagy

időintervallumon történő vizsgálatot a legcélszerűbb ilyen, kis pontokon elvégzett mérésekből összerakni.

3.2. Egész időtartam vizsgálata

Induljunk ki tehát abból, hogy van egy, keresztkorrelációs módszerekkel működő eljárásunk, amely képes megmondani a jelek korreláltságának mértékét adott időponthoz, adott integrációs idő mellett, többféle módszernek megfelelően. Egy ilyen, (időpont-integrációs idő) pont lesz az alapegység (pixel), ilyenekből akarjuk összerakni az egész síkot. A síkban az idő dimenzióban nyilván az adatsor időtartamát kell lefednünk (illetve amekkora tartományt éppen vizsgálni akarunk), az integrációs hossz síkban pedig valamennyi, általunk választott hosszúságot kell megnéznünk. Ezen értékek kiválasztása nehéz feladat, mivel adott jelalak és jelhosszúság mellett ki lehet ugyan számolni (vagy megmérni) az ideális integrációs hosszat, a méregravitációs hullámok frekvenciájáról és alakjáról azonban csak nagyon halvány elképzeléseink vannak (ugyanilyen nehézségek voltak a LIGO frekvenciaérzékenységeinek megtervezésekor). Ezért is van szükség több, különböző integrációs hossz mérésére. A rendelkezésünkre álló információra építve 10 és 20 közé eső számú integrációs hosszra végeztük el a számolásokat, ezek az $1msec$ és a $0.5sec$ tartományba estek (a tartomány nagy bekapcsolódásunkkor nagy részben már meg volt határozva, ezt csak egészen kissé módosítottuk, elsősorban a nagyobb intervallumok bevonása irányában, mivel ezek gyakorlatilag elhanyagolható gépidőt jelentenek, bizonyos esetekben azonban ezeknek is információértékük lehet).

Fontos kérdés még az idő-integrációs hossz sík felosztása, azaz hogy a pixelek milyen távolságonként követik egymást. Erre az általunk kapott két programcsomagban két különböző megoldás volt. Az egyikben a lépés legkisebb egységeként, azaz $1/4096sec$ -onként történt a LIGO-ból érkező feldolgozott adat frekvenciájának megfelelően, de tervbe volt véve ennek 90%-os átfedésre történő módosítása (azaz az egymás melletti integrációs tartományok ekkora mértékben fednek át egymással), a másik program viszont 50%-os átfedést használt. Mi ezt beállítható átfedésre változtattuk, a kísérleti mérések alapján azonban azt mondhatjuk, hogy az 50%-os átfedés legtöbb esetben elégséges (ennél viszont már nem jó lejjebb menni), így a jelentős gépidőmegetakarítás miatt általában érdemes ezt használni.

Meghatároztuk tehát a síkok pixeleinek helyét és azok paramétereit. A program ilyen módon elvégzi minden pixelre a mérést a három teszt statisztikával, a rendelkezésre álló adatokat pedig elmenti.

Egy ilyen mérés egy interferométerpárra végezhető el. Feltételezhetjük azonban, hogy ennél több interferométer áll rendelkezésünkre (LIGO-H2,

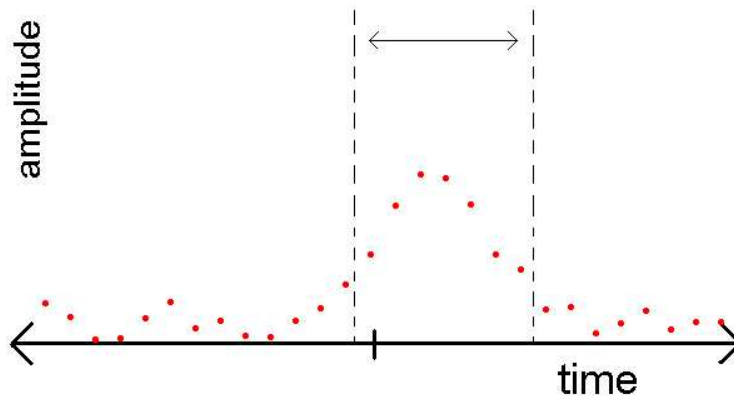
VIRGO, Tama, stb). A több interferométerből nyert adat pedig nem csak az irány jobb meghatározása végett lehet hasznos, hanem a nagyobb adatmennyiség miatt a korreláltság erősítését és a korrelálatlanság gyengítését is tovább fokozhatja. Ennek megfelelően a program az összes adott interferométerpárra (a kívánt interferométerpár méréseket az inputfile-ban külön meg lehet adni) elvégzi a sík felmérését, végül pedig mindhárom tesztstatistikára átlagolja a síkok értékeit (az általunk kapott programok közül az egyik 2 detektort, a másik hármat használt, és ezekre átlagolt, nálunk mind a detektorok száma, mind a vizsgálandó detektorpárok pontos listája megadható).

Ezen programrész kimenete tehát minden egyes teszt statisztikához egy egy, az interferométerpárokra átlagolt ábra az időpont-integrációs hossz síkon. Ezek kerülnek aztán további feldolgozásra.

3.3. Halmazok

A meglévő, időpont-integrációs hossz síkhoz rendelt, a három teszt statisztikával kapott, interferométerpárokra átlagolt értékek arról tartalmaznak információt, hogy egy-egy pontban milyen mértékben korreláltak a detektált adatsorok, tehát valamilyen módon az adott helyen egy lehetséges jel valószínűségéről kapunk információt. Nagy korreláltság azonban egy-egy pontban könnyen megjelenhet pusztán a zaj fluktuációi miatt is. Másfelől ha valahol egy külső jel miatt mérünk korrelációt, akkor ez a jel nagy valószínűséggel néhány, szomszédos pixelre is hatással van, részben, mert az időlépés kisebb lehet, mint a jel hossza, így a szomszédos pontokra integrálva is adódik még valamilyen maradék hatás, másfelől különböző integrációs hosszak egyaránt tartalmazzák a jelet, ill. egy részét. Azt mondhatjuk tehát, hogy egy adott jel hatása nem csak egy pixelben, hanem egy kitejedt tartományban jelenik meg (ahogy azt vázlatosan az 5. ábra mutatja), míg a zaj lokális fluktuációja semmiképpen sem terjed ki ennyire, így a jelek azonosításához mindenképpen érdemes nagyobb, összefüggő területeket felhasználni, amelyből lényegesen több információt nyerhetünk ki (azaz tovább javul a jel-zaj arány).

Felmerül a kérdés, hogy milyen alakú lesz egy adott jel esetén a hozzá tartozó ponthalmaz(cluster), amelyből a legtöbb információt ki tudjuk nyerni. Erre több, kevésbé hatékony módszer kiválasztása után a következő eljárást találtuk. Azt lehet tudni, hogy egy adott *effektív* hosszúságú jel mely pixel értékeit fogja befolyásolni. Ez egy meghatározott alak lesz az időpont-integrációs hossz síkon, amely függ az effektív hosszától. Jellemezzük az adott jelet a következő módon. Vesszünk egy legnagyobb értéket a síkon (a különböző integrációs hosszhoz tartozó pontok nem hasonlíthatók össze egyértelműen, így azok valamilyen normalizációjára van szükség. Ezt most előlegezzük meg, a részleteket a következő pontban tárgyaljuk). Ezt megkaphatjuk a

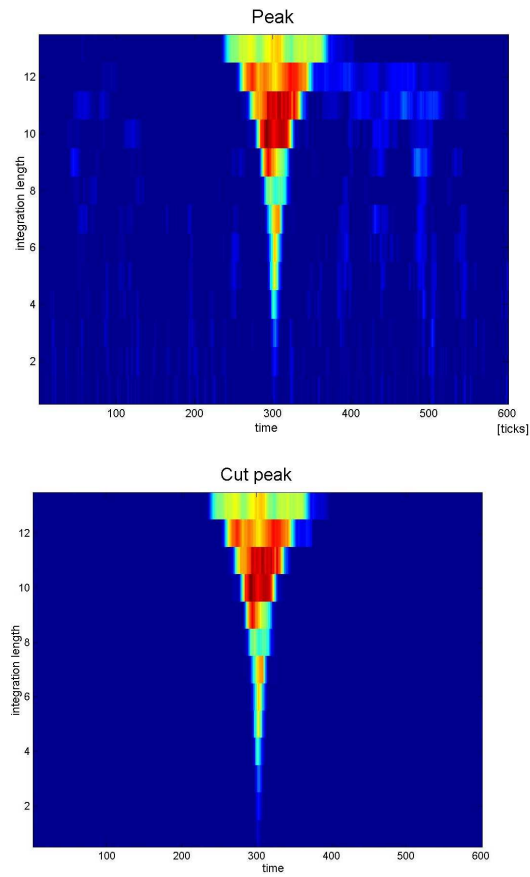


5. ábra. Az egyes jelek az adatsor egy nagyobb tartományát befolyásolhatják.

síkbeli értékek valamiféle simításával is, például, hogy a szomszédos elemek átlagainak keressük a maximumát, így kevésbé valószínű módon kapjunk a zajból származó értékeket. Ez a plusz eljárás azonban a kísérletek során nem bizonyult döntőnek, így megmaradtunk az egyszerű maximumkeresésnél. Az így megtalált maximum valamilyen integrációs hosszhoz fog tartozni. Azt mondjuk, hogy ez legyen a potenciális jel effektív hossza (így visszafelé definiálhatjuk az effektív hosszat, miszerint ez az a hossz, aminél ha az integrációs hossz ekkora, akkor ideális esetben, tehát zaj hiányában maximális korreláltságot kapunk (a normálásnak itt is fontos szerepe van)). Adott effektív hossz esetén viszont meg tudjuk mondani, hogy pontosan milyen alakú területre lehetett hatással az adott jel. Ezt a területet használjuk tehát föl.

Egy erős beérkező (szimulált) impulzus hatását mutatja a 6. ábra. Ezen jól látható a megjelenő jellegzetes alak, ami a befolyásolási tartományt jellemzi. Az ábra második pontja a háttérből kivágott jelet mutatja. Ilyen erős jelnél ez ránézésre is könnyen ellenőrizhető.

Nem akarjuk azonban, hogy a sok, leginkább a zaj által befolyásolt pont elmossa az értékesebb, nagyobb korreláltságot mutató pontokat, így nem az összes, csak meghatározott számú pontot használunk fel ebből a halmazból. Ezt mi a programban annyinak választottuk, ahány integrációs hosszat használunk. Ezen kiválasztott pontok értékeinek az átlagát vesszük, és az adott potenciális jelhez az adott teszt statisztikában ezt a számot rendeljük. A tényleges jelnagyság megállapításához a módszert különböző erősségű szimulált jelekkel futtattuk, így felmérve, hogy adott érkező jelmérethez mekkora értéket ad az eljárás.



6. ábra. Szimulált gravitációs hullám hatása a keresztkorrelálás utáni amplitúdóértékekre. A második ábra a program által a háttérből kivágott alakot mutatja.

3.4. Gamma illesztés

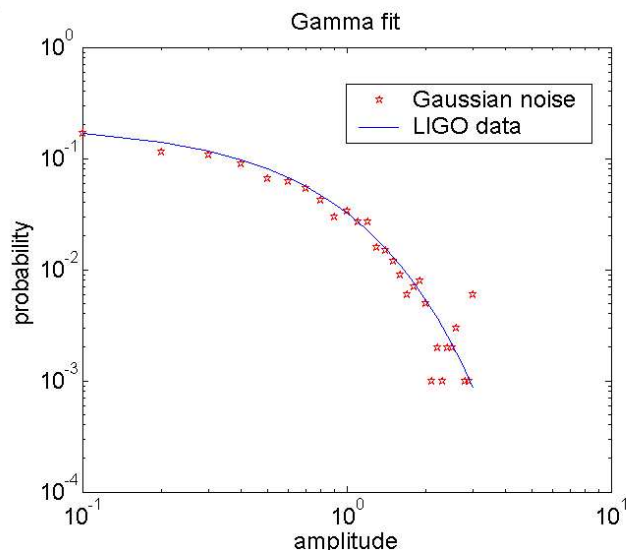
A fenti eljárásnál összehasonlításra kerültek a különböző integrációs hosszaknál kapott pontok. Ez azonban nem végezhető el egyszerűen a kapott értékekkel, mivel a különböző integrációs hosszak esetén más más értékek tartoznak adott jelhez, az amplitúdók nem összehasonlíthatók. Ezen probléma kiküszöbölésére normálnunk kell az egyes sorokat. Ennek érdekében statisztikát kell készítenünk a sorokban található értékekből. Lényeges azonban, hogy az egymás melletti pontok az átfedés miatt korreláltak, így ha minden pontot felhasználnánk, a statisztika nem lenne megfelelő. Ennek érdekében ki kell választanunk egymástól távolabbi, nem korrelált pontokat. Nem jó azonban, ha ezek nagyon távol helyezkednek el egymástól, mivel ez a felhasználható pontok számának csökkenése miatt jelentősen ronthatja a statisztika pontosságát. Ennek érdekében a következő eljárást használjuk.

1. Kiválasztjuk a legnagyobb elemet az adatsorból, ezt fel fogjuk használni a statisztikában.
2. Ezen pontot és környezetét, amely az átfedések miatt vele korrelált lehet, kivágjuk az adatsorból (minden elemét nullával tesszük egyenlővé).
3. Az eljárást előlről kezdjük (goto 1).

Ezzel már korrelátlan adatsorhoz jutunk, amelyen már készíthetünk statisztikát. Fontos hangsúlyozni, hogy a pontokat nem véletlenszerűen választjuk ki első lépésben (bár ezt is tehetnénk), hanem mindig a legnagyobb elemet választjuk ki, ezzel befolyásolva, hogy milyen pontok kerülnek be. Ez némileg más statisztikát eredményez a véletlenszerű esethez képest. Ezt az eljárást azért csináljuk (csinálhatjuk), mert számunkra a nagy értékek, a csúcsok statisztikája az érdekes, ilyeneket keresünk. Az a fontos tehát számunkra, hogy ezek azonosan legyenek súlyozva az egyes sorokban. Minde mellett azt tapasztaltuk, hogy nincs lényeges eltérés a két lehetőség (a másik a véletlenszerű válogatás) között, mindkettő hasonló módon használható.

Elkészítve a statisztikát azt tapasztaltuk, hogy mind fehér (Gauss) zaj, mind a detektor által mért háttér esetén a statisztikára jól illeszthető Gamma függvény. Egy ilyen minden sorra illesztve, annak paramétereinek megfelelő módon normálva elérhetjük, hogy minden integrációs hosszhoz tartozó sor azonos eloszlást kövessen.

Ilyen módon már közvetlenül összehasonlíthatóvá válnak a különböző sorokban található pontok. Ezt az eljárást a program teljes egészében elvégzi, és az így normált, összehasonlítható pontokon végzi el a további számításokat.

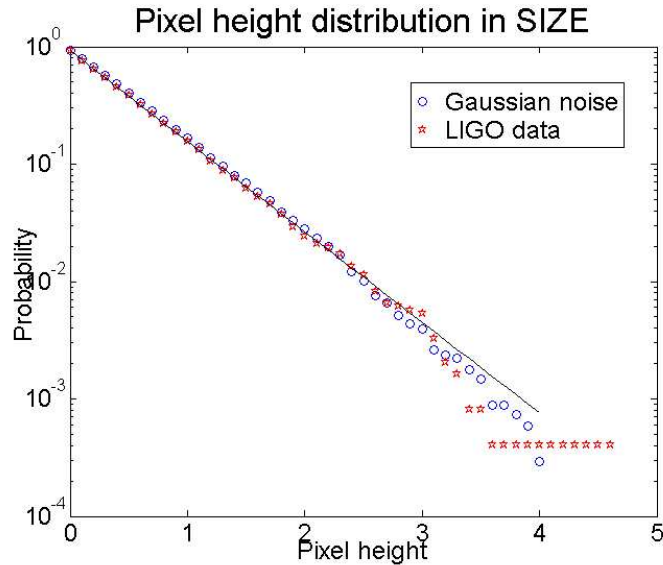


7. ábra. *Gamma illesztés*

3.5. Események keresése (Event Scan)

Az előzőekben tárgyalt eljárással tehát egy adott időpontban kapunk egy, megfelelő módon kiszámított korrelációs értéket a hozzá tartozó effektív hosszal, illetve az ebből számolt befolyásolási tartománnyal együtt. Meg tudjuk azt is határozni, hogy ha ez a mért valami egy beérkezett jel, akkor annak mekkora az amplitúdója. Döntő jelentőségű azonban az, hogy az adott pontban milyen valószínűséggel található tényleges jel, illetve milyen valószínűséggel van csak zajról szó. Ezen érték, a téves riasztási valószínűség (false alarm rate) meghatározásához statisztikát kell készítenünk a zajból adódó amplitúdóértékekről. Ehhez felhasználhatjuk, hogy a vizsgált adatsorban elhanyagolható a tényleges jelek száma. A következő algoritmust használjuk amplitúdóstatistika készítésére:

1. Kiválasztjuk a legnagyobb elemet az időpont-integrációs hossz sík-ról
2. A pont helyéből kapott effektív időtartammal meghatározzuk azon pontok halmazát, amelyekben a potenciális jel hatása megjelenhet
3. Ebből a halmazból kivesszük a legnagyobb n elemet (n =integrációs hosszak száma), ezeknek vesszük az átlagát, és ezt az átlagot feleltetjük meg a potenciális jel magasságának



8. ábra. Szimulált Gauss zaj, illetve a detektoroknál mérhető háttér esetén a program által elkülönített halmazok amplitúdójának statisztikája, ill. a rájuk illesztett exponenciális függvény.

4. Azt a területet, amelyre az előzőekben mért potenciális jel hatással lehetett, kivágjuk az időpont - integrációs hossz síkból (minden pontjának értékét 0-ra állítjuk), majd előlről kezdjük az eljárást (goto 1).

Az eljárással nyert potenciális jel-amplitúdókat eltároljuk, és ezen értékekből készítünk statisztikát. Azt tapasztaltuk, hogy függetlenül attól, hogy fehér (Gauss) zajjal, vagy a detektorok háttérével dolgozunk, Erre a statisztikára jól illeszthető exponenciális függvény. A mért görbét, illetve az illesztést a 8. ábrán láthatjuk. Egy exponenciális függvényt illesztve már könnyen meghatározható egy adott amplitúdóértékhez, hogy az milyen valószínűséggel tényleges jel, illetve milyen valószínűséggel csak a zaj hatása, így tehát meghatároztuk a téves riasztási valószínűséget (false alarm rate).

Ezt a számítást a program elvégzi, elmentve minden ponthoz a hozzá tartozó téves riasztási valószínűséget, így a felhasználónak elég ezen számok ismeretében vizsgálni az egyes időpontokban előforduló történéseket. Erre külön mindenképpen szükség van, mivel sok ilyen érzékelt, valószínűnek tűnő jelről állapítható meg, hogy milyen, földi forrásból származik. A LIGO-nál több száz különböző érzékelő található a detektorok környezetében, amelyek

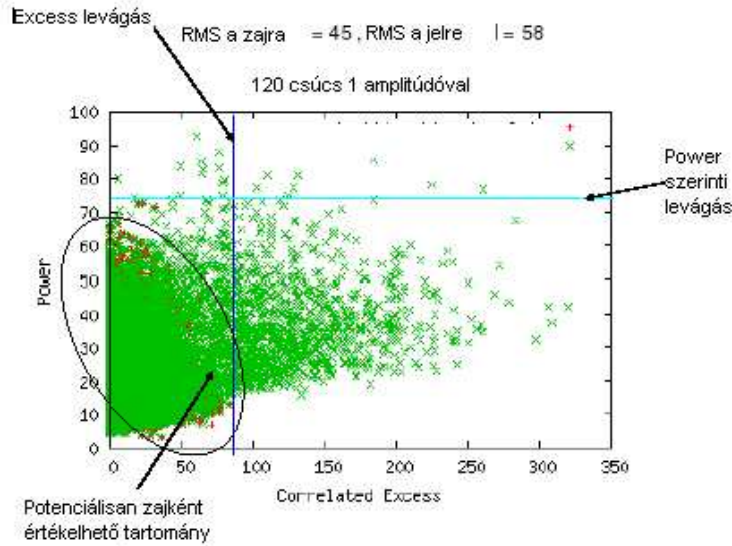
segítenek azonosítani az egyes földi forrásokat, így kézzel kiszűrhetők egyes nagy amplitúdójú jelek.

3.6. Teszt statisztikák egyesítése

A fenti eljárást mindhárom teszt statisztikával elvégezve három különböző listát kapunk a lehetséges pontokkal. Mivel a statisztikák más más módon dolgozzák fel az adatot, érzékenyséjük is más lehet, más-más módon lehetnek érzékenyek egyes jelekre, így együttes használatukkal azt várhatjuk, hogy így még több információhoz jutunk, összességében tovább növelve ezáltal az érzékenységet. Ez a részfolyamat még fejlesztés alatt áll, lényege az, hogy a kísérletek szerint a három teszt-statisztika választerében a zajból származó jelek ellipszoidszerű, de legalábbis legömbölyített alakú részben szóródnak szét. Ha csak az egyes tesztstatisztikákra követelünk meg külön külön korlátot, miszerint azokat tekintjük jelnek, amelyek kiemelkednek ezen korlátok fölé, ez egy téglatest kijelölését jelenti a térben. Ehhez képest az ellipszoidszerű térből kiemelkedés megkövetelése kisebb területet jelent, így értékes pontokat vonunk ve a jel kategóriába, amelyek nem jutottak volna át a szűrésen pusztán a külön külön alkalmazott korlátok esetén. A teszt statisztikák együttes használata fokozza tehát az érzékenységet. Ezt a gyakorlatban olyan módon vihetjük be az algoritmusba, hogy a kiértékelésnél a három teszt statisztika esetén kapott időpont-integrációs hossz ábráknak valamilyen kombinációját használjuk, amelyet az előzőekben megemlített ellipszoid alakja határoz meg, és ezzel végezzük el a későbbi számításokat. Ez a rész még sajnos nincs beépítve a programba, jöllehet egyelőre külön program formájában ezen analízis is elvégezhető. Az algoritmus e részének pontosítására még részletesebben tanulmányoznunk kell az ellipszoidot, illetve annak megfelelő definiálási módját.

Néhány, programunk e különálló részével készített ábrát azonban áttekintő jelleggel bemutatnánk (lásd a 9. és 10. ábrát).

A fentiekben vázolt algoritmussal tehát összességében a detektorokból származó, előre feldolgozott (preprocesszált) adatsorból adott időpontokhoz rendelt potenciális jelamplitúdókat kapunk, a hozzájuk tartozó effektív időterttel, téves riasztási valószínűséggel, a jel valószínű irányával (egy tartományon belül) együtt, így a felhasználó számára.



9. ábra. Mesterségesen generált jelek és zaj együttesére végzett analízis eredményei a korrelált felesleg-egyszerű korrelációs kimenet síkon. A zajtól terhelt tartomány az ábrán jól elkülöníthető, s ezáltal a valós események kiszűrése elvégezhető.

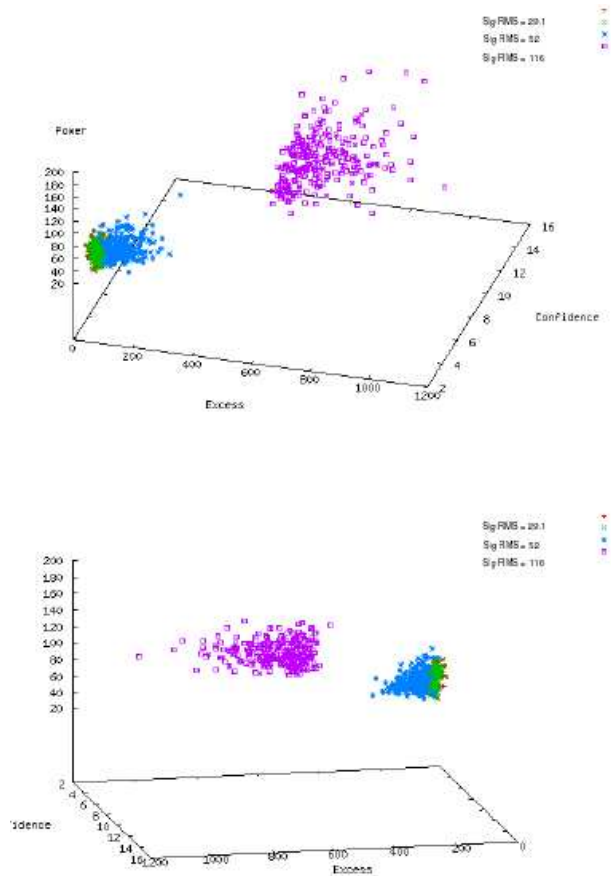
4. A program

Az előző fejezetben ismertettük az algoritmust, amelyet a program használ az adatsorok feldolgozására. Mindemellett azonban vannak olyan programrészek illetve tulajdonságok, amelyek nem kapcsolódnak közvetlenül az adatfeldolgozáshoz, hanem egyéb szempontok miatt van rájuk szükség. Ezek kerülnek ismertetésre ebben a fejezetben.

4.1. Input/Output

A program írása során fontos szempont volt, hogy a felhasználó könnyen megváltoztathassa a futás egyes paramétereit, illetve egyszerűen hozzáférhessen az eredményekhez. Ezeknek azért van különösen nagy jelentősége, mivel rengeteg olyan paraméter van, amely változtatható a programban, illetve kimenetként is óriási mennyiségű adat jelenik meg. Ezek egyszerű kezelése elengedhetetlen.

A program számára a bemeneteket egy külön adatfileban kell megadni, így az összes paraméter egy helyen megtalálható, nem kell azokat minden



10. ábra. *Mesterségesen generált jelek és zaj együtteseire végzett analízisek eredményei mindhárom teszt-statisztika kimenetének háromdimenziós terében. A valós jelek zajtól terhelt tartományoktól való elkülönülése itt még szembevetűőbb.*

futtatásnál újra írni, illetve egyszerűen megtalálható, valamint megváltoztatható egy egy közülök. A következő paramétereket lehet megadni:

- ifonames - a felhasznált detektorok nevei
- ifopairnames - amilyen interferométerpárokra el akarjuk végezni a kereszt-korrelációs számításokat
- sample_rate - ilyen a mintavételezési frekvenciája a kapott adatsoroknak (a LIGO előre feldolgozott (preprocessed) adatsorai 4096 Hz -esek)
- Length - az adatsor hossza. Ennyit olvas be a program

- `start_time` - az adatsor adatából ennyiedik másodpercnél kezdjük meg a feldolgozást
- `duration` - ennyi másodpercnyi adatot dolgozunk fel az adatsorból
- `size_core` - a teszt statisztikáknál a Core paraméter mérete (*msec*-ban)
- `size_lobe` - a teszt statisztikáknál a Lobe paraméter mérete (*msec*-ban)
- `size_gap` - a teszt statisztikáknál a Gap paraméter mérete (*msec*-ban)
- `intlen` - a mért integrációs hosszak
- `delta` - az átfedés mértéke (pl 0.5 - 50%)
- `datafile_name` - ebből a fileból olvassa ki a program az adatsorokat
- `InputMode` - ezzel dönthetjük el, hogy a program az adatfileből beolvasott adatot dolgozza fel, vagy a beépített Gauss zaj-generátorral és berakott jelekkel dolgozzon tesztelési célból
- `Plot` - beállítható hogy a kirajzolásra kerüljenek e bizonyos ábrák, illetve hogy legyen e egyáltalán ilyen kimenete a programnak.
- `RMS` - zajgenerálásnál mennyi legyen a zajamplitúdók értéke az egyes virtuális detektoroknál.
- `ALPHA` - ez a konfidencia adja meg, hogy mekkora valószínűség fölött tekintse 0-nak, illetve az alatt 1-nek a korrelációt.
- A többi beállítható paraméternek elsősorban a tesztelésnél van jelentősége, illetve ha a programnak csak bizonyos részeit akarjuk használni, itt most nincs igazán jelentőségük.

Az egyes nevek után közvetlenül be kell írni az adatot. Új sorba is lehet írni bármit, a lényeg, hogy a program minden *space*-el elválasztott egységet egy egy paraméternek tekint az adott változóhoz, egészen egy másik változónévig. Az egyes változókat tetszőleges sorrendben meg lehet adni. Egy sor megjegyzésnek minősül, azaz nem kerül felhasználásra, ha bárhol található benne egy % jel.

Az outputnál a kiírás különböző file-okba történik. Külön fileba kerülnek a legnagyobb események, a hozzájuk tartozó valószínűségekkel illetve az eltolásokkal, amikből az irány meghatározható. Lényegesebb azonban, hogy az összes lényeges információt a program (Matlab verzió) egy Matlab adatfileba

tárolja el, ahonnan megtekintésre, illetve további feldolgozásra is behívhatók az eredmények.

Megfelelő beállítás ($Plot \geq 1$) esetén a programnak (Matlab verzió) grafikus kimenete is van, ábrázolásra kerülnek az egyes teszt statisztikákhoz tartozó időpont - integrációs hossz síkok a megfelelő amplitúdókkal. Ez lényeges eleme a kiértékelésnek, a *látványból* sokminden elmondható egy adott részletről, sokminden ilyen módon is megállapítható. Ellenőrzés és tesztelés esetén is döntő szerepe lehet az ábráknak.

4.2. Sebesség

Ha az előre feldolgozott adatot tekintjük, egy LIGO detektor 300 másodpercenként nagyjából 20Mbyte-nyi adatot készít, $4MHz$ -s mintavételezéssel. Ezen adatmennyiség feldolgozása Rendkívül lassú, épp ezért mindent el kell követnünk annak érdekében, hogy a folyamatot felgyorsítsuk.

A programot alapvetően Matlabban írtuk. Ennek az az oka, hogy itt aránylag könnyebb a fejlesztés, mint az alacsonyabb nyelvű programnyelveken, és gyakorlatilag minden szükséges jelfeldolgozási művelet rendelkezésre áll. Az előnyök mellett azonban ez a nyelv sajnos aránylag lassú, különösen az egyes ciklusok terén. Fontos volt, hogy a programnyelven belül megtaláljuk az egyes feladatokra a lehető leggyorsabb algoritmusokat. Ehhez - bizonyos változtatásokban - kihasználtuk, hogy a for ciklusok helyett mátrixműveletek használata jelentősen megnöveli a sebességet, valamint néhány Matlab parancsot, mint például a szórás számolását magunknak írtuk meg, ezzel kiküszöbölve a Matlab parancs hibaszámolási módja miatti lassulást. Ezekkel a változtatásokkal az alapprogramban körülbelül 300%-os gyorsulást értünk el. A célunk a valós idejű adatfeldolgozás lenne, ettől azonban még az így felgyorsított Matlab programmal távol voltunk körülbelül egy nagyságrenddel. Ennek érdekében az egész programot elkezdtuk írni egy másik programnyelven, DMT-ben is. Ez egy, a Caltechen fejlesztett $C++$ -alapú nyelv, amely elsősorban jelfeldolgozási feladatokra van kiatlálva. Ez a C nyelv gyorsasága miatt lényegesen gyorsabb, mint a Matlab. A fejlesztés a két nyelvben párhuzamosan fojt. Ez ugyancsak jó alkalom volt az eredmények ellenőrzésére, mivel ez a független megírás miatt kölcsönösen lehetővé tette az eltérésekkel járó hibák megtalálását. A DMT-ben íródott változat még nincs teljesen befejezve, de úgy tűnik, hogy képes lesz elérni a valós idő analízishez szükséges sebességet.

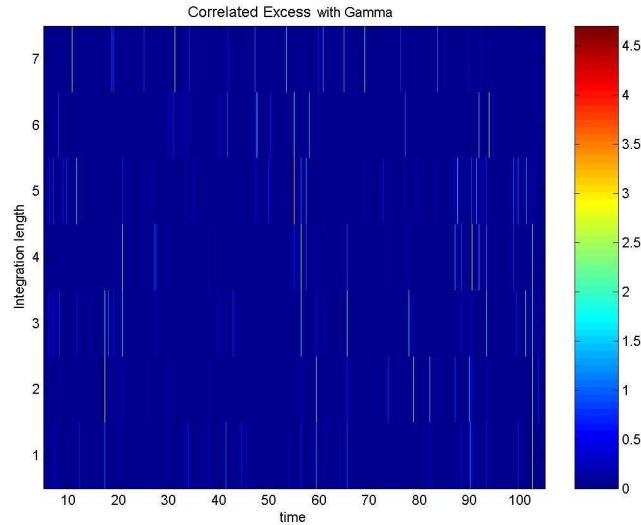
5. Alkalmazások

A programot megírás után volt alkalmunk a gyakorlatban is kipróbálni.

Az egyik legfontosabb feladat a LIGO detektoroknál a zaj csökkentése, illetve a befolyásoló földi hatások kiszűrése. Erősen zavaró tényezők lehetnek a detektorok közelében lecsapó villámok, amelyek mágneses hatása ellen az interferométerek betonburkolata nem jelent feltétlenül védelmet. A villámok komoly problémát okozhatnak, mivel már előfordult, hogy egy ilyen következtében le kellett állítani a mérést, ami viszont lényegesebb, hogy ha a villámok megjelenhetnek a mért adatsorban, mint valamilyen jelek, akkor nem lehetünk biztosak abban, hogy mikor találunk valamilyen jelet, akkor az nem egy valamilyen távolságra lecsapó villám következménye e, így folyamatosan figyelniük kell az ilyen tényezőkre, ami komolyan megnehezítheti a vizsgálatot. Korábban már egyéb módon valaki felfedezni vélt valamilyen erős hatást, amellyel a közeli villámok befolyásolnák az adatokat, így mi egy ilyen esetleges hatás felderítésére vállalkoztunk. Néhány olyan időpont közelében vizsgáltuk meg az adatsorokat, ahol az egyéb műszerek illetve megfigyelések nagy és a hanfordi detektorhoz közeli villámot mutattak. A hanfordi 1-es (*4km*-es) detektorból származó adatot kereszt-korreláltuk a program segítségével külön esetekben a másik hanfordi, ill. a livingstone-i detektorok által mért adatokkal. Ha valamilyen hatása lenne a villámnak, akkor azt a két hanfordi detektor korrelációjával bizonyosan ki lehetne mutatni az adott érzékenyséig. Ehhez képest mi azt találtuk, hogy az általunk elért érzékenységgel ilyen hatás nem mutatható ki. A vihar során mért nagy villám időpontjában, ill. akkörül mért amplitúdóértékek a 11. ábrán láthatók.

Úgy tűnik tehát, hogy a mérésekből nem kell kiszívni azokat az időpontokat, amikor valamelyik detektor környékén vihar zajlik.

Bár a Matlab kód gyakorlatilag teljesen készen állt a használatra, a teljeskörű dokumentáció, az elfogadtatás, illetve ellenőrzés mindig egy hosszabb folyamat, így együttműködésekben zajló mérésekre még nem volt használható. Volt viszont egy olyan változatunk, amely az egyik témavezetőnk programjának felgyorsított változata volt ($\sim +300\%$), egyéb módosítások nélkül. Erre a változatra gyorsan volt szükségünk, így ezt külön dokumentáltuk és ellenőriztük, hogy biztosan azonos eredményeket adjon az eredeti programmal. A felhasználás egy, a Tama detektorból származó adatsoron történt volna. Ezen adatsorba azonban a nemzetközi megállapodás révén mi nem tekinthettünk be, így a programot már a témavezetőnk futtatta az adaton.



11. ábra. Az egyéb mérőberendezésekkel érzékelt villám és környezetének hatása a keresztkorrelált adatban. A villám ideje 60sec körül volt.

6. Saját eredmények

A dolgozat során néhány helyen említést tettünk arra vonatkozóan, hogy mely részeket csináltuk mi a munka során, illetve melyek álltak rendelkezésünkre mások munkája által, a határ azonban eléggé összemosódik, így fontosnak látjuk ezzel külön foglalkozni.

Megérkezésünkön két külön program létezett kereszt-korrelációs mérésekre, amelyet a két témavezetőnk külön külön írt. Mindkettő egy egy teszt statisztikát használt a háromból, az egyik 2 detektorra működött, feltételezve, hogy a jel iránya ismert, míg a másik nagyobb időtartamú méréseket végzett. A mi feladatunk a két program egyesítése, felgyorsítása volt, valamint a munka során írtunk meg olyan részeket, amelyek a kezdeti keresztkorreláció után normálják, majd kiértékelik az adatokat. Gyakorlatilag minden részt továbbfejlesztettünk ill. újraírtuk, az eredeti Matlab programnyelv mellett pedig DMT-ben is elkészítettük a megfelelő báltozatokat. Fontos gyorsítás volt, hogy ez egyes keresztkorrelációs eljárásoknál a keresztkorrelációs számítás, ami a gépidő egy jelentős részét jelenti, sokban hasonló az egyes teszt statisztikáknál, így az összevonásnál el tudtuk érni, hogy ne jelenjen meg a teszt statisztikák számával arányos sebességcsökkenés. Az egyszerű input/output részt mi fejlesztettük ki az integrációs hossz-függő átfedések miatti újfajta eljárást igénylő kirajzolásig. Mindazonáltal ez egy csapatmunka volt, a munka

során rendszeresen konzultációkat tartottunk a témavezetőkkel, akik segítettek meghatározni a továbbhaladás irányát, saját igényeiket is megfogalmazva. Velünk dolgozott továbbá Pinkes Patel (Drexel University, Philadelphia), aki egyenrangú társként vette ki a részét a munkából.

7. Konklúzió

Összefoglalva a nyári gyakorlat során megírtunk egy programot, amely az előfeldolgozott (preprocessed) adatból megtalálja a lehetséges eseményeket a hozzájuk tartozó téves riasztási valószínűséggel és a jeé érkezési irányával. A programhoz különálló adatfile tartozik ahol egyszerűen megadhatók a bemeneti adatok, valamint a kimenet is megkönnyíti a további feldolgozást, így a program egyszerűen kezelhető. Jelentősen meggyorsítottuk a Matlab kódot, mindemellett a valós idő analízishez a DMT kódra lesz szükség. A meglévő programot alkalmazáson keresztül ki is próbáltuk. Eredményül az adódott, hogy az általunk elért érzékenységgel mérve a detektorok közelében lecsapó villámok nincsenek hatással a mért adatra.

8. Kitekintés

A jövőben teljesen be szeretnénk fejezni a DMT kódot, hogy az készen álljon a valós idő analízisre. Ugyancsak be szeretnénk fejezni a teszt statisztikák egyesítése kapcsán megemlített módszert. Ezen változtatások után a program készen fog állni a következő LIGO mérések analízisére.

9. Köszönetnyilvánítás

Mindenekelőtt szeretnénk köszönetet mondani Márka Szabolcsnak, aki megadta nekünk a lehetőséget, hogy résztvehessünk ezen az egyedülálló gyakorlaton, valamint az egész időszak alatt segítette a munkánkat és mind szakmailag, mind emberileg sokat tett a fejlődésünkért. Szeretnénk köszönetet nyilvánítani Laura Cadonatinak, aki, mint második témavezető segítette a munkánkat, az együtt töltött időszak alatt együtt dolgozhattunk vele az általa végzett munkában. Szeretnénk továbbá köszönetet mondani John Zweizignek és Patrick J. Suttonnak, akik segítettek betekintést nyerni a DMT programnyelvbe (amely nagy részben John Zweizig fejlesztett ki). Szeretnénk köszönetet mondani Alan Weinsteinnek, az általa tartott ismertető előadások nélkülözhetetlenek voltak a LIGO felépítésének és működésének felépítésében, valamint amiért előben volt alkalmunk megtekinteni a Caltech-en felállított $40m$ -es interferométert. Szeretnénk köszönetet mondani Kenneth G. Libbrechtnek a a hanfordi LIGO detektorok bemutatásáért, valamint az általa tartott, rendkívül hasznos pénteki szemináriumokért. Végül szeretnénk köszönetet mondani Pinkesh Patelnek, akivel végig együtt dolgozhattunk és sokat segített az ötleteivel, a szorgalmával és a látásmódjával.