Kozmikus részecskeadatsorok online monitorozása

BSC SZAKDOLGOZAT

Kálmán Dávid* III. fizika BSc hallgató, informatikus-fizikus szakirány ELTE TTK



Témavezetők:

dr. Barnaföldi Gergely Gábor MTA Wigner FK RMI Elméleti Fizikai Főosztály

dr. Varga Dezső ELTE TTK Komplex Rendszerek Fizikája Tanszék

Budapest, 2013.

Tartalomjegyzék

1.	Bevezetés	1
2 .	A kozmikus sugárzás kutatása és alkalmazásai	2
	2.1. Ultranagy energiás kozmikus részecskék vizsgálata	3
	2.2. A Pierre Auger Obszervatórium	3
	2.3. Az IceCube kísérlet	5
	2.4. Légköri müondiagnosztika	6
	2.5. Kozmikus részecskék alkalmazott kutatásokban	6
	2.6. Kozmikus részecskék az oktatásban	7
3.	A REGARD müontomográf	8
	3.1. A müontomográf általános felépítése	8
	3.2. A közelkatódos kamrák felépítése és működése	9
	3.3. Az adatgvűjtő és -feldolgozó rendszer (DAQ)	11
	3.4. A müontomográf kezelőfelülete (HMI)	12
4.	Mérések a DÖME berendezésnél	12
	4.1. Analóg adatok a DÖMÉ-nél	14
	4.2. Az elnvelődés vizsgálata	16
	4.3. A DÖME leképezése kozmikus részecskékkel	17
5.	Adatfeldolgozás	18
	5.1. Az adatok szerkezete	19
6.	Az online feldolgozó és megjelenítő keretrendszer	20
	6.1. A szerver oldali feldolgozás	20
	6.2. Kliens oldali feldolgozás	24
	6.3. Online monitorozás	25
	6.4. Archivált adatok vizsgálata	26
	6.5. További tervek, kitekintés	28
7.	Összefoglalás	29

Ábrák jegyzéke

1.	Kozmikus részecskezápor vázlatos képe az Argentínában található Pierre		
	Auger Obszervatóriumban [6]		
2.	A Pierre Auger Obszervatórium térképe [7]		
3.	Az IceCube kísérlet detektorrendszerének sematikus rajza [11]		
4.	A HiSPARC hálózat detektorai egy középiskola tetején. [19] 8		
5.	A REGARD müontomgráf felépítése [4]		
6.	Egy közel katódos kamra keresztmetszete, valamint a kamrában kialakuló		
	elektromos térerősség vázlatos rajza [2, 3, 22]		
7.	A müontomgráf adatgyűjtó rendszerének blokkvázlata [23]		
8.	Referenciamérés elrendezése a DÖME berendezésnél.		
9.	Analóg adatok a Run 320-as mérésnél.		
10.	0. A triggerelt események és az illesztések gyakorisága		
11.	A mért fluxus a sűrűséghossz függvényében összehasonlítva az elméleti szi-		
	mulációból kapott görbével [24]		
12.	A merőleges müonhozam csökkenése a DÖME oldalfalánál mérve [4] 17		
13.	DÖME berendezés leképezése [4]		
14.	Az adatszerkezet		
15.	Az adatfeldolgozás folyamata		
16.	Az adatkezelő program sémája		
17.	Az alkalmazás online felülete, analóg adatok		
18.	Az alkalmazás offline felülete, analóg adatok		
19.	Az alkalmazás offline felülete, fluxus hisztogramok		

1. Bevezetés

A természetet vizsgáló emberiséget mindig is érdekelte a világot felépítő részek, alapelemek kutatása. Ennek nyomait már az ókori görögöknél is megtalálhatjuk. Ugyanakkor a modern fizikát megalapozó nagyenergiás részecskekutatás csak alig évszázados múltra tekint vissza. Ezalatt az idő alatt egyre nagyobb energiájú részecskegyorsítók készültek, melyekkel az anyag elemei építőkövei azonosíthatóak. E kutatásokkal párhuzamosan a természetben előforduló nagyenergiás részecskék vizsgálata is megindult, melyek sok esetben nagyobb energiájúak, mint mesterségesen laboratóriumi körülmények között gyorsítottak. Így olyan fizikai folyamatokat is tesztelhetünk, amelyekre csak a "jövő gyorsítóiban" nyílhat lehetőség.

Szakdolgozatom elkészítése során bekapcsolódtam az MTA Wigner FK és az Eötvös Loránd Tudományegyetem közös, gáztöltésű detektorok kutatásával és fejlesztésével foglalkozó csoportjának munkájába. A REGARD (RMKI ELTE Gaseous Detector Research and Development) csoport korábban kifejlesztett egy olyan gáztöltésű, proporcionális, közelkatódos (CCC: Close Cathode Chamber) kamrákból [1] álló detektort, amellyel kozmikus müonok fluxusának szögeloszlását lehet mérni. A müontomográfot a csoport a kozmikus részecskék szögeloszlásában mutatkozó különbségek mérése alapján főleg rejtett, földalatti kőzetinhomogenitások (pl. rejtett barlangi járatok) keresére használja [2, 3], de emellett sok más szerkezeti vizsgálatot is el tud végezni a detektor [4]. Igény mutatkozott azonban a mérések során arra, hogy bizonyos esetekben a felhasználó online is figyelni tudja az adatokat, például hogy az eszköz helyes működéséről meggyőződhessen, illetve korábban rögzítésre kerül adatokat kielemezhessen. Távlati célként pedig több ilyen detektor hálózatba kötése szerepel, ahol lehetőség nyílhat például kozmikus részecskezáporok monitorozására, magaslégköri események elemzésére.

Diplomamunkám során létrehoztam egy olyan online keretrendszert, azaz egy szerver oldali adatfogadó és -feldolgozó, valamint egy kliens oldali vizualizációs alkalmazást, ami rendelkezik e kívánt funkciókkal. Feladatom ennek megtervezése és a szükséges analizáló, adatgyűjtő és adattömörítő (DAQ: Data Aquisition System) szoftver elkészítése volt.

2. A kozmikus sugárzás kutatása és alkalmazásai

A kozmikus részecskezáporok vizsgálata már több mint 100 éves múltra tekint vissza. A Viktor Hess híres kísérletei [5] óta eltelt idő alatt egyre jobban megismerhettük a kozmikus eredetű ionizáló sugárzást. A később kísérletek segítségével megállapították, hogy a világűrben terjedő primer kozmikus sugárzás többkomponensű: elsősorban protonokból (89%), vasig bezárólag stabil atommagokból (10%) és elektronokból (1%) áll. További vizsgálatok azt is igazolták, hogy ezek a nagyenergiás primer részecskék jellemzően körülbelül 30 km-es magasságban, a ritka felsőlégkör O és N atommagjaival ütközve másodlagos részecskezáporokat keltenek. A részecskezáporokban – az erős kölcsönhatás törvényei szerint – elsősorban pionok (π^0 , π^{\pm}) keletkeznek, amelyek β -bomlással müonokká (μ^{\pm}) alakulnak.



1. ábra. Kozmikus részecskezápor vázlatos képe az Argentínában található Pierre Auger Obszervatóriumban [6].

2.1. Ultranagy energiás kozmikus részecskék vizsgálata

A nagy energiás részecskék vizsgálata napjaink részecske és magfizika kutatásainak középpontjában áll, gondoljunk csak a RHIC¹, a Tevatron vagy az LHC² részecskegyorsítókra és a hozzájuk tartozó kísérleti berendezésekre! Az ezekben az óriási gyorsító-komplexumokban elérhető energia még mindig nagyságrenddekkel alacsonyabb, mint egyes kozmikus részecskéké. Továbbá, az egzotikus részecske- és magfizikai folyamatok előfordulási valószínűsége az ultranagy energiákon lesz számottevő. (Definíció szerint ultranagy energiásnak a 10¹⁸ eVnál nagyobb energiájú részecskéket hívjuk.) Így, érthető módon, szükség van a kozmikus záporok szerkezetének vizsgálatát lehetővé tévő detektorok építésére. Az alábbiakban bemutatok ezen kozmikusrészecske-detektorok közül néhány jelentősebb kísérleti együttűködést.

2.2. A Pierre Auger Obszervatórium

1992-ben Jim Cronin és Alan Watson vetette fel a lehetőségét egy – a fent említett céllal épített – detektrorendszernek, amelyet Pierre Auger Obszervatóriumnnak neveztek el. Mivel az ultranagy energiás kozmikus részecskék igen ritkák, ezért nagy kiterjedésű "érzékeny felületre" volt szükség. Az Argentínában megépült komplexum munkálatai 2008ban fejeződtek be teljesen, azonban már 2004 óta gyűjt adatokat a mintegy 3000 km²-en elterülő detektorhálózat.



2. ábra. A Pierre Auger Obszervatórium térképe [7]

 $^{^1 \}rm Relativistic$ Heavy Ion Collider: Relativisztikus Nehézion Ütköztető, Brookhaven, NY, USA $^2 \rm Large$ Hadron Collider: Nagy Hadronütköztető, Genf, Svájc

A Pierre Auger Obszervatóriumban két módszerrel vizsgálják a kozmikus záporokat [8].

- 1. Az egyik módszer az ún. Cserenkov-sugárzás mérésén alapul. Ha egy töltött részecske gyorsabban halad egy közegben, mint a közegbeli fénysebesség, akkor az anyag molekuláit polarizálja. Az alapállapotba visszakerülő elektronok fotonokat bocsátanak ki, amelyek konstruktívan interferálva egymással jellegzetes képet adnak. A Pierre Auger Obszervatórium területén több mint 1600 Cserenkov detektor van egymástól 1.5 km távolságra elhelyezve (2. ábra, *kék pontok*). Az egyes detektoroknak legfőbb eleme egy zárt víztartály amelyen a részecskék áthaladnak. Ebben fotonokat keltenek, amit egy fotoelektron-sokszorozó gyűjt be. Utóbbira azért van szükség, mert a Cserenkov sugárzás során csak néhány Cserenkov foton keletkezik, amelyek által keltett fotoelektronok még nehezen detektálhatóak. A fotoelektronokat megsokszorozva azonban már mérhető jelet kaphatunk. Ha időben bizonyos toleranciával ugyanakkor szólnak detektorok, akkor egy adott primer részecskétől származó zápor elemeinek tekintik őket. Az egyes részecskék által keltett fotonok számából következtetnek a keltő primer részecske energiájára. A jelek idő és térbeli differenciájából pedig a primer részecske pályája adható meg.
- 2. Ezt kiegészítendő az érzékeny felület körül ún. légköri fluoreszcencia detektorokat is elhelyeztek (2. ábra, *lila vonalak*). A kozmikus részecskezápor töltött részecskéi gerjesztik a légkör nitrogén atomjait, amik újra alapállapotba kerülve UV fotont bocsátanak ki. Az optikai detektorok ezeket a felvillanásokat észlelik. Ennek a módszernek hátránya, hogy csak éjszaka, tiszta égbolt mellett kivitelezhető, ami jelentősen lecsökkenti a statisztikát és a napi mérési időt.

A Pierre Auger Obszervatóriumnál a két módszert együtt használják. A két független mérés lehetőséget biztosít a részecskezápor rekonstrukciójára és a szisztematikus hibák kiküszöbölésére. A rendszert több irányba is tervezik továbbfejleszteni. Ezek közül a legfontosabbak [9]:

- a föld alá épített müondetektorok (AMIGA: Auger Muons and Infill for the Ground Array, Auger Müonok és Betét a földi hálózathoz),
- a rádióteleszkópok, amik a másodlagos részecskék által kibocsátott koherens rádiótartományú sugárzást mérnék,
- új fluoreszcencia teleszkópok, amik magasabb légköri tartományokba is ellátnak (HEAT: High-Elevation fluorescence Telescopes, nagy magasságú fluoreszcencia teleszkópok).

2.3. Az IceCube kísérlet

Az IceCube kísérlet a nehezen detektálható neutrínókat hivatott mérni. Közel nulla tömegük és kis kölcsöható képességük miatt a neutrínókat nehéz detektálni. Ezért általában a neutrinó detektorok nagy méretűek, hogy minél nagyobb valószínűséggel következzék be valamilyen kölcsönhatás a detektor térfogatában. Egyike a legnagyobb ilyen detektorrendszereknek az Antarktiszon található IceCube [10].

Az IceCube lényegében egy közel 1 km³ méretű érzékeny térfogatú detektorrendszer, amely főleg neutrinók detektálására szolgál. Az Antarktisz jegébe elhelyeztek úgynevezett Digital Optical Module-okat (digitális optikai modul, DOM), amelyek hasonlóan a korábban említett Pierre Auger Obszervatóriumhoz, szintén a felvillanó Cserenkov fotonokat figyelik.

A több mint 5000 fényérzékeny egységet 2010 májusa óta kiegészíti a DeepCore, ami az alacsonyabb energiás (~ 10GeV) neutrinók mérésére lett kifejlesztve. A DeepCore az eredeti érzékeny térfogat alatt helyezkedik el 2100 m mélységben.



3. ábra. Az IceCube kísérlet detektorrendszerének sematikus rajza [11].

2.4. Légköri müondiagnosztika

A kozmikus müonok föld felszíni intenzitása függ bizonyos atmoszférikus folyamatoktól. A müondiagnosztika módszere éppen ezt használja ki. A Moszkva melletti Dubnai Kutatóközpontban található müonhodoszkóp egy 45 m² felületű, szcintillátorokból összeállított, 2 dimenzióban érzékeny detektor. A helyfelbontása 1 cm, a szögfelbontás pedig 1°. Ezzel a rendszerrel lehetőség van a helio-, magneto és atmoszférában lezajló dinamikus folyamatok vizsgálatára.

A fenti vizsgálatokkal néhány esetben olyan változásokat észleltek a fluxusban, amik később időjárási események előjelének bizonyultak. Így például nagy viharok és tornádók előtt is jelentős változást lehetett tapasztalni a müonhozamban. A teljesítmény spektrumban az általuk hullámfolyamatnak (angolul waveprocess) nevezett jelenség látszódott. A viharok által okozott lökéshullámok jelentek meg a felszínen tapasztalható fluxusban. 2009-ben egy alkalommatl ez a hullámfolyamat 3 órával előzött meg egy olyan nagy havazást, amelyet a meteorológusok nem jeleztek előre. 2007-ben és 2010-ben szintén 3 illetve 1 órával korábban jelentkeztek a müonhozamban az anomáliák két, nagy károkat okozó vihar előtt³.

Az építők távlati céljai közt szerepel, hogy olyan stratégiai fontosságú helyekre is telepítsenek ilyen detektorokat, mint például a repülőterek. A detektorok rövidtávú, de megbízható előrejelzésével elkerülhetők komolyabb katasztrófák, hiszen például időben le lehet zárni a repteret egy nagyobb vihar vagy havazás előtt [12, 13].

2.5. Kozmikus részecskék alkalmazott kutatásokban

A müontomográfia módszere, azaz a kozmikus müonok szögeloszlásánk és fluxusának mérése, egyre több alkalmazott kutatásban is szerepet kap. Ezek közül hármat emelnék ki.

Elsőként a régészeti kutatásokban való alkalmazást mutatom be. Az 1960-as években még folyt az egyiptomi piramisok belsejének feltérképezése. A régészeket főleg az addig rejtett kamrák keresése érdekelte. A korabeli módszerekkel azonban ez leginkább ásással és a piramis szerekezetének megbontásával járt. Luis W. Alvarez vetette fel először azt az ötletet, hogy a kozmikus részecskék hozamának változásából következtessen a detektor feletti anyagsűrűségre és ezzel rejtett kamrákat keressen. Munkatársaival meg is építettek egy detektort, amit a Kephren piramis alatt, a Belzoni kamrában helyeztek el. A kamra felett elhelyezkedő Királyok Kamrája mellett még más kamrák létét is feltételezték, ezeket próbálták megkeresni. Méréseik alapján kimutatták, hogy nincs további 1 – 2 méternél nagyobb üreg a piramisban [14].

³Akkor még nem használták a detektort előrejelzésre.

- Egy másik fontos alkalmazási terület a vulkán-tomográfia, amivel főként japán tudósok foglalkoznak. A vulkánok szerkezetének vizsgálata fontos a működésük megértése szempontjából. Azonban az aktív vulkánok környezetében a körülmények (magas hőmérséklet, mérgező gázok és gőzök) általában megnehezítik a szokásos geológiai méréseket. A müontomgráfia segítségével viszont lehetőség nyílik a vulkán belsejébe tekinteni. Ehhez szcintillátorokból és fotoelektron sokszorozókból épített müonteleszkópot használtak. Az első ilyen jellegű mérést K. Nagamine és munkatársai végezték a Tsukuba-vulkánnál, a 90-es évek közepén. Mérésük során azt vizsgálták, hogy előrejelezhetőek-e a vulkán kitörések a müontomgráfia módszerével [15]. Később H. Tanaka és munkatársai szintén a müonok eloszlásának segítségével próbálta meg az Asama vulkánt feltérképezni. A japán kutatók méréseikkel kimutatták a vulkán belső szerkezetét, a kürtő helyét és becslést tudtak adni a magma sűrűségére is [16].
- Az Egyesült Államokban a terrorizmus elleni megelőző intézkedések kiemelt szereppel bírnak. Ennek egyik eleme a bomba készítésére alkalmas nukleáris anyagok nyomonkövetése és ellenőrzése. A nagy sűrűségű anyagok kimutatására általában röntgen átvilágítást alkalmaznak. A Los Alamosi Nemzeti Laboratóriumban azonban L.J Schultz és K.N. Borozdin vezetésével egy kozmikus müonokon alapuló rendszert próbálnak kifejleszteni [17]. A müonok anyagban történő szóródását felhasználva lehetséges a nagy rendszámú elemek érzékelése, még akkor is ha csak kis mennyiségben van jelen, vagy árnyékolás mögött van az adott elem.

A rendszer felépítése egyszerű: a mérendő objektum (például egy kamion) fölött és alatt egyaránt elhelyeznek két-két detektort. A talált anyag összetétele és vastagsága megmondható, ha az alsó detektorok közé fémlapokat helyezve lényegében egy kalorimétert építünk, amivel a müonok impulzusát lehet megmérni. Két detektor és két fémlap esetén ez 50%-os pontosságú impulzus mérést jelentett.

A rendszer egyik fő előnye, hogy nem használ mesterséges sugárzást, így nem teszi ki felesleges sugárterhelésnek sem a vezetőt, sem az esetleg raktérben tartózkodókat. A módszer emellett elegendően gyors, annak eldöntése, hogy van-e a raktérben nagy rendszámú anyag (például 92-es rendszámú urán), néhány percet vesz igénybe.

2.6. Kozmikus részecskék az oktatásban

A dolgozat szempontjából szintén fontos "alkalmazás" a HiSPARC projekt [18]. A Hollandiából elindult kezdeményézés célja főként oktatási jellegű. A projekt keretében középiskolák és egyetemek tetején helyeznek el nagyenergiás részecskedetektorokat, amelyeket egy közös hálózatba kötnek. Egy-egy ilyen detektor két egymás felett elhelyezett szcintillátorból áll, amelyeknek a végén egy-egy fotoelektronsokszorozót helyeztek el. A méréshez használt elektronika lényegében egy koincidencia egység. Az összeállításhoz hozzátartozik még egy GPS egység is, ami az eseményeket idő és hely koordináta "bélyeggel" látja el. Általában egy autós tetőcsomagtartóba rakják a detektort, hogy védjék az időjárástól. A detektorok adatait feltöltik a világhálóra, így azok közvetlenül is elérhetővé válnak az



4. ábra. A HiSPARC hálózat detektorai egy középiskola tetején. A szcintillátorok és a fotoelektronsokszorozók a tetőcsomagtartókban vannak elhelyezve.

érdeklődők számára. A kíváncsibb középiskolások eközben megismerkedhetnek a nagyenergiás fizikával előadások és a detektorok karbantartása és felügyelete közben, továbbá lehetőségük nyílik tanáraik segítségével akár az adatok elemzésére is. A program jelenleg is tart, és folyamatosan csatlakoznak iskolák és egyetemek, melyek száma mostanra már elérte 100 állomást.

3. A REGARD müontomográf

A csoportunk célja egy olyan detektor kifejlesztése volt, mely főként alkalmazott kutatásokban használható, például geológiai mérésekre alkalmas (barlang-tomográfia), hasonlóan a 2.5 fejezetben bemutatott példákhoz. Egy detektor, amely laboratóriumon kívül is használható, a szokásosnál ellenállóbb felépítést kíván. Ez mind méret mind pedig működésbeli korlátokat támasztott a detektorral szemben, melyek nagy részére már sikerült optimális megoldásokat találni. A továbbiakban részletesen is bemutatom a RE-GARD müontomográfot, amellyel már több sikeres mérést végeztünk barlangokban, kutatóintézetekben és ipari létesítményekben [2, 4, 20].

3.1. A müontomográf általános felépítése

Az elkészített detektor jelenleg 15 kg tömegű, a detektorház pedig 51 cm × 46 cm × 32 cm térfogatú. A detektor felépítése az 5. ábrán látható. A müonteleszkóp főbb részei a 4 darab közelkatódos kamra⁴ és a középső kamrák között elhelyezkedő adatgyűjtő egység, amelybe integrálták a nagy- és kisfeszültségű modulokat, a trigger egységet, ill. a kezelőfelületet.

 $^{^4\}mathrm{CCC}:$ Close Cathode Chamber, közelkatódos kamra



5. ábra. A REGARD müontomgráf felépítése [4]

3.2. A közelkatódos kamrák felépítése és működése

A müontomográf érzékelő részei az úgynevezett közelkatódos kamrák. A kamrák REGARD csoport által kifejlesztett CCC technológiával készültek. Ez a sokszálas proporcionális kamra, más néven $MWPC^5$ (1968, G. Charpak [21]) egy továbbfejlesztése.

A sokszálas kamra működése azon alapul, hogy a kamrán áthaladó részecske a benne lévő gázt ionizálja. Az ionizáció során keletkező szabad elektronok a kamrában lévő szálakon összegyűlnek, és ezt detektálhatjuk. Egy részecske azonban centiméterenként csak mintegy 100 elektront kelt, ami még nem detektálható. Ezért megfelelő elektromos teret al-kalmazunk, aminek hatására a szabad elektronok mozgási energiára tesznek szert. Az így nyert mozgási energiával már ezek az elektronok is képesek ionizációra, ezzel újabb elektronokat keltve. Ez a jelenség az elektronlavina. Egy ilyen lavinában $10^6 - 10^7$ db elektron keletkezik, ami megfelelő erősítés után már alkalmas lehet a detektálásra.

A CCC konstrukció annyiban más az MWPC-hez képest, hogy itt két féle szálat alkalmazunk, amik ráadásul kb. 1 – 3 mm távolságban aszimmetrikusan és közelebb vannak a katódhoz, mint az MWPC esetén. A két különböző vastagságú szálnak eltérő a funkciója. A térformáló szál (field-wire 100 μ m) negatív potenciálon van, míg az anódszál (sense-wire, 25 μ m) pozitívan. A 6. ábra jobb oldalán látható a szálak elhelyezkedése, a 6. ábra bal oldalán pedig az elektromos térerősség vonalak alakulása.

 $^{^5 \}mathrm{Multi-Wire}$ Proportional Chamber, sokszálas proporcionális kamra

A töltések érzékelése kétdimenzióban történik: egyfelől az anódszálakon, másfelől ezzel egyidejűleg a katódon, amely a szálakra merőlegesen van "parkettázva" a kamra alsó lapján. A padeken a térformáló szálak közelében összegyűlt pozitív ionok tükörtöltése jelenik meg. Az anódszálakat, és a parkettákat (padeket) egy időben kiolvasva megadható a beütés két dimenziós, síkbeli koordinátái.

Az eddigi tapasztalatok alapján a kamrák töltésére az Ar (80%) $CO_2(20\%)$ gázkeverék vált be legjobban. Az argon mivel nemesgáz, kémiai reakciókban nem vesz részt. Nem alkot molekulákat sem, így nincsenek forgási és rezgési módusai, csak rugalmasan ütközik más atomokkal vagy a keltett elektronokkal. A lavina keltése során maguk az Ar atomok is gerjesztődnek. Miközben visszakerülnek alapállapotba egy 11,6 eV-os UV fotont bocsátanak ki. Ezek a nagyenergiás UV fotonok beleütközve a rézből készült katód lemezbe, a fotoeffektus által, további elektronokat lökhetnek ki, így az elektronsokszorozódási folyamat önfenttartóvá válna és a kamra mérésre alkalmatlan lenne. A CO_2 adalék feladata éppen az, hogy a feleslegesen keletkező UV fotonokat elnyelje. További előnye ennek a gázkeveréknek, hogy olcsó és nem éghető⁶.



6. ábra. Egy közel katódos kamra keresztmetszete, valamint a kamrában kialakuló elektromos térerősség vázlatos rajza [2, 3, 22]

A detektor építése során fontos szempontok voltak a mechanikai ellenállóság, valamint a szélsőséges körülmények (magas páratartalom, alacsony hőmérséklet) tűrése. A ház masszív, átlátszó plexi lemezekből került kialakításra, amelyek megfelelő stabilitást biztosítanak, ugyanakkor nem teszik túlzottan nehézzé sem a berendezést. A CCC technológiával készült kamrák szintén jól tűrik a mechanikai terhelést, ez is egy előnyük az MWPC kamrákhoz képest. A páralecsapódás, valamint a kamrákban kialakuló esetleges kisülések elkerülésének érdekében a kamrákon szekvenciálisan átfolyó gázt visszavezetjük a detektorházba is, ezzel csökkentve, kiszorítva a házból a páradús levegőt. Így mintegy 40-60%-os relatív páratartalom csökkenés érhető el, ami hasznos lehet 100%-os páratartalmú helyeken történő mérések esetében.

 $^{^6\}mathrm{Ezt}$ az olcsó gázkeveréket az iparban is nagy mennyiségben használják védőgáz néven.

A szakdolgozat készítése során volt alkalmam többször is mérést végezni a müontomográffal, amelyek egyikéről a Nukleon folyóiratban egy cikk is megjelent [4]. Később egy jóval nagyobb érzékeny felületű detektor kamráinak építési munkálataiban is aktívan részt vettem.

3.3. Az adatgyűjtő és -feldolgozó rendszer (DAQ)

A DAQ vagyis az adatgyűjtő és adattovábbító rendszer feladata, hogy a padeken és az érzékeny szálakona FEE elektronikák segítségével összegyűjtse, a DAQ felé továbbítsa majd ott eltárolja. Az alábbiakban ezt a folyamatot mutatom be részletesen.

A térformáló szálakról illetve a padekről kiolvasott töltésmennyiséget front end elektronika erősíti és digitalizálja, ezt küldi az adatfeldolgozó egységnek. A detektor emellett rögzít analóg adatokat is: páratartalom 3 helyen (a detektoron kívül, az alaplapon és a házban), hőmérséklet (ugyanazokon a pontokon), nyomás, tápfeszültség értékeket. A müontomográfban történt fejlesztések miatt a detektor most már rögzíti azt is, hogy milyen trigger mintázat szerint került eltárolásra az esemény. A triggerelési feltétel is kevésbé szigorú, elegendő hogy legalább két kamra "megszólaljon", a korábbi három helyett [2]. Az adatgyűjtő rendszert Melegh Hunor, a BME hallgatója készítette el, szakdolgozata keretében [23]. A DAQ blokkvázlata a 7. ábrán látható.



7. ábra. A müontomgráf adatgyűjtó rendszerének blokkvázlata [23]

A front end elektronikák, CMOS technológiával, a REGARD csoport saját fejlesztésében készültek el. Egy-egy elektronikán 16 csatorna van. Egy kamrában 80 pad és azokra merőlegesen 80 szál van, ezért összesen 10 db kártyára van szükség, 5-re a szálak csatornáinak, 5-re pedig a padek csatornáinak kiolvasásához. A kártyák felerősítik és digitalizálják az egyes kamrák jeleit. Amikor a DAQ felől érkezik egy trigger jel, a csatornákon lévő biteket eltárolják a shift regiszterükbe. A DAQ ezután 2MHz-es órajellel kiolvassa a tárolók tartalmát.

Egy triggerelt esemény 465 bit memóriát foglal el. A PIC32 alapú DAQ 32kB belső memóriával bír, és 100 eseményenként kiírja az addig összegyűjtött adatot egy FAT16 fájlrendszerű SD kártyára, amelynek maximális tároló kapacitása 2GB, ami nagyjából 10 millió esemény. Ez 10 méter földekvivalens mélységen körülbelül 2 hónapnyi mérést jelent.

Egy-egy esemény rögzítéséhez tehát hozzátartozik a külső környezeti változók analóg rögzítése, az érzékeny szálakról az analóg (összegyűlt töltéssel arányos) jel, a padekről és a térformálószálakról 16 csatornánként egy-egy D.IN 4.16 típusú kártyával pedig digitális jelek kiolvasása. Később lehetőség lesz rá, hogy szabványos Ethernet porton keresztül, megadott IP cím felé küldje az egyes esmények adatait. Ennek az online kidolgozása volt a szakdolgozatom elkészítése során az egyik feladatom.

3.4. A müontomográf kezelőfelülete (HMI)

A detektor kezelése a detektor külső plexidobozának elején található LCD kijelző és a hozzá tartozó nyomógombok segítségével történik. Itt indíthatjuk el és állíthatjuk le a mérést. A kezelőfelület felhasználóbarát és egyszerű, mindössze két darab gombbal rendelkezik. A különböző menükön keresztül elérhetők az analóg szenzorok adatai és a rögzített események száma is.

4. Mérések a DÖME berendezésnél

2012 nyarán lehetőségem nyílt résztvenni a REGARD csoport egy mérésében az MTA Energiatudományi Kutatóintézetében. Célunk az volt, hogy a müonok anyagban történő elnyelődéséről illetve energialeadásáról szerezzünk pontosabb információkat. Ezek az ismeretek elengedhetetlenek lesznek később a müontomográfiás módszer pontosításához. További célunk volt azt is demonstrálni, hogy lehetséges a tomográfiás eljárással nagy sűrűségű anyagok (esetünkben egy néhány tonnás vas-ólom objektum) gyors leképezése. A mérés tárgya az MTA Energiatudományi Kutatóintézet kísérleti reaktora mellett elhe-

A meres targya az MTA Energiatudomanyi Kutatomtezet kiseneti reaktora menett emelyezkedő $DOME^7$ nevű berendezés volt. A DOME egy 80 cm × 80 cm × 80 cm belső térfogatú, 15,6 cm falvastagságú, öntöttvas rétegekből felépülő eszköz. Súlya körülbelül 8,5 tonna.

⁷DÖME: DÖgnehéz MérőEszköz

A kutatóintézetben a DÖMÉ-t alacsony hátterű mérésekre használják. A DÖME különlegessége, hogy csak kis mértékben járul hozzá a háttérsugárzáshoz, ugyanis anyaga a II. világháború előttről, egészen konkrétan a németek által felrobbantott Erzsébet-hídból származó öntöttvas. Így az általa adott "háttérsugárzása" kicsi, hiszen még jóval az atomkorszak előtt készült, ezért még nem kerültek bele a reaktorokból, atomrobbantásokból származó izotópok. Ezen értékes tulajdonsága alkalmassá teszi kis intenzitású sugárzásokkal járó folyamatok pontos vizsgálatára. Emellett a nagy anyagmennyiség a külső sugárzást is jól árnyékolja, tovább csökkentve a háttér sugárzások által okozott zajt. Esetünkben azonban csak a DÖME, mérete és masszív szerkezete miatt vált érdekessé. A mi méréseinket a háttérsugárzás nem kozmikus eredetű komponense nem befolyásolja, a pályaillesztési feltételeknek köszönhetően.



8. ábra. Referenciamérés elrendezése a DÖME berendezésnél.

Ahhoz, hogy meg tudjuk állapítani, hogy mekkora változást okoznak a müonhozamban az abszorbensek, szükségünk van referenciamérésekre, amelyek során megmérjük, hogy abszorbensek nélkül milyen lenne a müonfluxus. A detektort ehhez elhelyeztük a DÖME tetején, és két napig gyűjtöttük az adatokat.

Ezután a detektort különböző pozíciókban és különböző abszorbens vastagságok alatt elhelyezve (mérések összefoglalását ld. a 1. táblázatban) további adatokat gyűjtöttünk. Az abszorbens anyag mennyiségét nem csak a DÖME adta, hanem néhány későbbi mérésben ólomtéglákat is elhelyeztünk a detektor felett, növelve a sűrűséghosszat⁸.

⁸Sűrűséghossz jele: ρ_L , mértékegysége: $\frac{g}{cm^3} \cdot cm$, azaz az abszorbens anyag sűrűsége szorozva a hosszával

Mérés sorszáma	Mérés ideje	Detektor helyzete	
320	2012.07.17 07.19.	DÖME tetején	
321	2012.07.19 07.23.	DÖMÉ-n belül	
322	2012.07.23 07.24.	DÖME alatt, középen	
323	2012.07.24 07.27.	DÖME alatt, középen	
324	2012.07.27 08.02.	$\rm D\ddot{O}ME$ alatt, 0,5 cm Al és 5 cm Pb réteg	
325	2012.08.02 08.07.	$D\ddot{O}ME$ alatt, 15 cm Pb réteg	
326	2012.08.07 08.08.	DÖME alatt rétegek nélkül	
327	2012.08.08 08.10.	DÖME oldala alatt	
328	2012.08.10 08.14.	DÖME oldala alatt, 45 fokkal elforgatva	
329	2012.08.14 08.17.	DÖME tetején	
330	2012.08.17 08.21.	DÖME belsejében ferdén	

1. táblázat. A DÖMÉ-nél végzett mérések összefoglaló táblázata

A mérések nagyjából 5 hétig folytak, 2012. július 17-től augusztus 21-ig. Ezután következett az adatanalízis, amelyet az általam készített programmal készítettem el. Megjegyzem, hogy ellenőrzésnek rendelkezésemre állt a korábbi offline adatanalizáló program is, amellyel szintén megvizsgáltam a mérési adatokat összehasonlításként. Az adatokat a szerver oldalon futó alkalmazás offline módban dolgozta fel, ahogy azt a 6. fejezetben később bemutatom. A program kimeneteleként kapott eredményeimet és ábráimat az alábbiakban összegzem.

4.1. Analóg adatok a DÖMÉ-nél

Az analóg adatok mérésére az előző fejezetben részletesen kitértem. A 9. ábrán a környezeti változókat, valamint a tápfeszültség változását láthatjuk az idő függvényében az első, azaz a DÖME tetején végzett 47 órás referenciamérésben. Jól látszik például, hogy 20 óra elteltével a páratartalom a detektor házán kívül mintegy 20%-ot emelkedett, ugyanekkor a detektoron belül gyakorlatilag állandó maradt.



9. ábra. Analóg adatok a Run 320-as mérésnél.

A 10. ábrán a frekvenciákat láthatjuk, negyedórákra átlagolva. Piros színnel a triggerelt események gyakoriságát ábrázoltam. Ez annak felel meg, hogy a detektor milyen gyakran rögzített adatot. Zöld színnel pedig az illesztett események gyakoriságát ábrázoltam. Ez utóbbi adatsor már csak azoknak az eseményeknek a gyakoriságát mutatja, amelyek megfelelnek az egyenesillesztési kritériumnak, azaz, hogy csak egy klaszter legyen kamránként és hogy a két szélső kamrában legyen klaszter.



10. ábra. A triggerelt események és az illesztések gyakorisága.

	Átlag [Hz]	Hiba
Triggerelt	8,685	$\pm 0,014$
Illesztett	6,537	$\pm 0,012$

2. táblázat. A triggerelt és az illesztett események frekvenciája

4.2. Az elnyelődés vizsgálata

A folyamatok jobb megismerése miatt fontos volt, hogy megnézzük, hogy az átlagos müonfluxus hogyan változik a sűrűséghosszal. Erről van elméleti görbe, amivel szintén össze lehet vetni a méréseinket [24]. Látható, hogy igen jól illeszkednek az elméleti görbére a mért adataink.



11. ábra. A mért fluxus a sűrűséghossz függvényében: a DÖME mérésnél, földalatti méréseknél (Ajándék-barlang, Jánossy-akna) összevetve az elméleti szimulációkból kapott görbével [24]

4.3. A DÖME leképezése kozmikus részecskékkel

A mérés másik célja az volt, hogy mevizsgáljuk: valóban lehetséges-e egy nagyméretű objektum leképezése a müontomgráfia módszerével. A leképezés elkészítéséhez két, abszorbensek nélküli mérést végeztünk el: a referenciamérést, valamint egy további 4 napos mérést. Ekkor a detektor a DÖME fala alatt helyezkedett el, 45 fokkal elforgatva a berendezés falához képest. A leképezéshez az időegység alatt a detektor felületére merőlegesen (zenitben) érkező kozmikus müonok arányát számoltuk ki minden felületelemre (pixelre), mind a berendezés alatti, mind pedig a felette történt mérésekre. Ezt követően felületelemenként kiszámoltuk, hogy a detektorunk feletti berendezésben hány százalékot csökken a müonhozam.

A 12. ábrán jól megfigyelhető három elkülönülő tartomány:

- a bal felső sarokban a berendezés teteje és alja (250 \pm 11 $\frac{\rm g}{\rm cm^2})$ okozta átlagosan 30 %-os hozamcsökkenés,
- középen a berendezés fala alatti (630 \pm 18 $\frac{\rm g}{\rm cm^2})$ átlagosan 50 %-os hozamcsökkenés,
- jobb alsó sarokban a berendezés alól kilógó $(0 \pm 0, 06 \frac{g}{cm^2})$ detektorfelület által mért átlagosan 0 %-os hozamcsökkenés.



12. ábra. A merőleges müonhozam csökkenése a DÖME oldalfalánál mérve (százalékban megadva) [4]. Az ábrán jól megfigyelhet a detektorhoz képest 45°-ban álló fal struktúrájának hatása, amely effektus már néhány napos mérés után láthatóvá vált.



13. ábra. DÖME berendezés leképezése. Az ábrán jól látszik a berendezés fala (kék "L" alakú folt), valamint a kihajtott ajtaja [4].

A 13. ábrán a kozmikus müonok kétdimenziós zenit- és azimutszög-eloszlását láthatjuk $sr^{-1}s^{-1}$ egységekben a DÖME berendezés alól mérve. Az origó a detektor mértani középpontja, a vízszintes és függőleges tengelyeken az x és y irányú zenitszög-vetületek láthatók. Az eloszláson kékkel színeztük az alacsonyabb, pirossal a magasabb hozamú térszögeket. A kék "L-alakú" rész mutatja a DÖME falait és a jobb oldali zárt ajtó által kitakart térszögeket. A müontomográf a DÖME alatt aszimetrikusan, balra helyezkedett el. A mérés során a berendezés bal oldali ajtaja nyitva, a jobb oldali zárva volt. Ez utóbbi vetülete látszik az ábra jobb alsó térnegyedében. Ezek alapján a detektor egyértelműen képes nagy sűrűségű anyagok károsodásmentes leképezésére kozmikus müonok segítségével.

5. Adatfeldolgozás

A szakdolgozat megírása során a feladatom a már korábban kipróbált adatgyűjtési, -tárolási és -feldolgozási struktúra átültetése egy online felületre. Ennek célja kettős: egyfelől szolgálja ez az adatok gyors megfigyelését, és felügyeletét. Másrészt távlati célok közt szerepel egy ilyen rendszer oktatási célokra való felhasználása is. Most a DÖME mérés kapcsán fogom bemutatni, hogy az offline, nem valós idejű adatanalízis hogyan zajlik.

5.1. Az adatok szerkezete

A csoport az évek során már kidolgozott egy jól működő adatkezelési módot. Az adatokat két részre lehet osztani, az analóg adatokra illetve a szálak és padek digitális adataira.



14. ábra. Az adatszerkezet

A 14. ábrán látható egy adatfájl szerkezete. Az eseményt leíró adatok közül az első az esemény sorszáma. A számozás nullától indul. Ezt követi az előző esemény óta eltelt idő, mikroszekundum egységekben. Ezután következnek a csatornák adatai. A kiolvasás sorrendje szerint előbb a szálakat, aztán a padeket olvassuk ki. Egy eseményhez tartozó beütési mintázatot egy nullákból és egyesekből álló számsor kódolja, ahol az egyes, ha "megszólalt" a szál és nullás, ha nem. A csatorna-adatok az átláthatóság és a helymegtakarítás miatt 10-es számrendszerbe átváltva kerülnek eltárolásra. Ez úgy történik, hogy kettes számrendszerben írt számként értelmezzük és átváltjuk tizes számrendszerbe (így például a 65535 megfelel az 1111 1111 1111 1111 nek). Ezzel legalább 70%-al csökkentjük a kimeneti fájl méretét, hiszen 16 karakter helyett legfeljebb 5 kerül eltárolásra. Így a kimeneti fájlban egy szám 16 csatornát, azaz 16, nulla vagy egy értékű számot reprezentál. Minden detektor réteghez 160 csatorna tartozik (80 szál és 80 pad irányú), így ha négy kamránk van, mint például a DÖME mérésénél, akkor összesen $\frac{4\times 2\times 80}{16} = 40$ db szám reprezentálja a beütési adatokat.

Az analóg adatok a berendezésbe épített mérőeszközökből származnak. Ezek közül három a hőmérséklet, három pedig a páratartalom mérésére szolgál. Az eszközök úgy kerültek elhelyezésre, hogy egy a detektoron kívül, egy a berendezés házában, egy pedig az alaplapon gyűjti az adatokat. Továbbá, a detektor adatgyűjtő rendszere eltárolja az akkumulátor feszültséget, a nagyfeszültség-modulon folyó áramot illetve a nyomás értékeket. A DAQ eltárolja azt is, hogy egy eseményben mely kamrák szólaltak meg, és melyek nem. Az első szám a kamra adatok után (ld. a 14 ábrán, trigger mintázat) tartalmazza a trigger adatokat, méghozzá úgy, hogy az abból kettes számrendszerbe átváltott szám első 8 bitjét kell nézni. A trigger mintát az első 5 bit (hogyha 5 darab kamra van), tartalmazza, ezt pedig követi egy lezáró 101 bitsor. Ez jelzi, hogy vége van a trigger mintának.

6. Az online feldolgozó és megjelenítő keretrendszer

Az előző fejezetben láthattuk, hogy a detektor képes a korábban részletezett módon az események rögzítésére. A detektor továbbá rendelkezik egy soros port kimenettel is. Ekkor a nyers adatok megfelelő fogadása után sor kerülhet a fájlok feldolgozására. A teljes folyamat sematikusan a 15. ábrán látható, ahol már több detektor hálózatát rajzoltam fel.



15. ábra. Az adatfeldolgozás folyamata

6.1. A szerver oldali feldolgozás

A detektor az eseményeket az 5.1 fejezetben leírt adatstruktúrában továbbítja. Ennek fogadását egy C++ programnyelvben írt, online adatfeldolgozó program kezeli. Azért esett a választásom erre a nyelvre, mert ebben jól megoldott a TCP/IP kommunikáció, könnyen kezelhető könyvtárak állnak rendelkezésre az implementációhoz. A nyelv objektumorientáltságát jól ki lehetett használni, hogy megfelelően modulárisan tudjam felépíteni a programot.

Az adatok feldolgozása esemény alapú. Ez azt jelenti, hogy az egyes triggerelt eseményeket tekintem egységnek, és ezen végzem a manipulációkat. A feldolgozás során az adatok egy új, dátummal megjelölt fájlban kerülnek eltárolásra. Minden egyes fájl 24 óra adatait tartalmazza, ami később a megjelenítés szempontjából lesz fontos. Az adatokból a részecskepályára vonatkozó információk is kinyerhetők, ezt szintén elvégezi a program. Összefoglalva tehát szerver feladatai a következők:

- adatok fogadása és tárolása,
- adatfeldolgozás: klaszterezés és egyenesillesztés,
- kliens kiszolgálása.

Az alábbiakban ezeket fejtem ki részletesen.

Az adatok fogadása és tárolása

A detektor valamilyen csatornán keresztül továbbította az adatokat a szerver felé. A szerver oldalon egy program fut folyamatosan, ami figyeli a bejövő TCP/IP alapú forgalmat egy adott porton. Kiválasztottam a "12345" portot, mert azt nagyon ritkán szokták használni, ezzel a konfliktus lehetőségét minimálisra csökkentettem.

Az adatküldés és fogadás alapegysége az úgynevezett socket. A socket lényegében egy fájl, ám nem a klasszikus értelemben, mint például egy szöveges dokumentum a merevlemezen, hanem egy úgy nevezett fájl leíró (file descriptor). Egy socketet úgy lehet elképzelni nagyjából, mint egy csatornát, ami tartalmazza a kapcsolatra vonatkozó legfontosabb információkat, így azt, hogy milyen IP⁹ cím tartozik a sockethez, hogy milyen IP családba tartozik (IPv4 vagy IPv6), a hoszt nevét és a port nevét. A kommunikáció az egyes programok socketei között zajlik. A szerver létrehoz egy socketet a saját IP címével, és ezt kapcsolja a porthoz, és figyelésre állítja be. Amikor kapcsolódási kérelem érkezik, a program létrehoz egy új child-processt. Ennek a child-processnek a feladata, hogy lemásolja a figyelő socketet, összekapcsolja a detektor socketével és kinyerje az adatot. Amikor az adatkinyerés befejeződött, a child-process befejeződik (terminál), a program ezután képes újabb adat fogadására.

A következő lépésben néhány egyszerű adatmanipulációt kell végrehajtani, mert a "raw" (nyers) adatok mértékegysége nem minden esetben megfelelő ábrázoláshoz. A hőmérsékleteket és a páratartalmat például a DAQ egy 1000-es szorzóval tárolja el¹⁰. Az adatok átskálázását a program elvégzi, majd beírja egy fájlba, amit dátum alapján nevez el¹¹. Egy-egy feldolgozott 24 órás adatfájl körülbelül 50-60 megabyte. Amikor az összegzést végzi a kliens oldali alkalmazás, akkor ezt az adatmennyiséget kellene gyakorlatilag rögtön a felhasználó rendelkezésére bocsátani. Mivel ez gyakorlatilag lehetetlen, az Internet korlátozott átviteli sebessége miatt, ezért a szerver program maga is elvégez egy 15 perces binelést, és így is eltárolja az adatokat egy **év_hónap_nap.hist** nevű fájlba. Ennek a mérete (1 Hz eseménygyakoriság mellett) körülbelül 1/900-ada lesz az eredeti fájlnak. Ez már könnyebben mozgatható adatmennyiség. A folyamat végén a "raw" digitális adatok már nem kerülnek eltárolásra, azokat csak az egyenesillesztéshez használja fel a program.

⁹Internet Prokoll

 $^{^{10}98,2~\%}$ páratartalom az adatsorban 98200-ként szerepel és így tovább.

¹¹2013 május 29-én keletkezett fájl a 2013_05_29.txt nevet kapja.

A klaszterezés és az egyenesillesztés menete

A szerveren futó program egyenesillesztést is végez. Az egyenesillesztés első lépése a beütési "foltok", az ún. klaszterek megkeresése. A detektorban az áthaladó részecske nem egyes szálakat (illetve padeket) szólaltat meg, hanem néhányat egy kis tértartományban. A klaszterek mérete az esetek nagy részében (99%-ban) nem haladja meg a 4×4 pont nagyságot, ahogy azt korábban már láttuk [25, 26]. Minden egyes kamrán megjelenik véges számú klaszter, az áthaladó részecske pályáját pedig a kamrák klasztereire történő illesztésből határozhatjuk meg. Mivel 1+1 dimenzióban történik a kiolvasás, ezért először csak a pálya két vetületét kapjuk meg, és ezután geometriai számolással adódik a pálya meredeksége és annak hibája. Az illesztett egyenesek adatai a program szintén hozzácsatolja az esemény adataihoz, ezt fogja majd felhasználni az online megjelenítő alkalmazás, hogy hisztogramokat készítsen a pályák meredekségeiről.

Jelenleg csak azokkal az eseményekkel foglalkozom, amikor minden kamrában pontosan egy klaszter jelenik meg. Ekkor a pálya meredekségét a két szélső beütés alapján határozom meg. Ez egy igen egyszerű eljárás, nagy hibával és kis hatásfokkal, azonban gyors, ugyanis a feltételek teljesülése esetén egy ciklus fut végig. Természetesen az algoritmus fejlesztés alatt áll, több klasztert tartalmazó esetekre is.

Ugyanis egy eseményben több klaszter is azonosítható egy adott kamrában, és ekkor minden lehetséges esetre egyenest kell illeszteni valamint ezeket osztályozni az egyenesillesztés hibája alapján. Ez az, ami alapján el lehet majd dönteni, hogy melyik a legvalószínűbb egyenes.

Kliens kiszolgálása

Az utolsó feladat a kliens kiszolgálása. Az előző fejezetben leírt módon a szerver oldalon futó program archiválja az adatokat, és így azok eltárolásra kerülnek a szerveren. A kliens program ezeket a fájlokat kéri le, a szokásos, IP alapú hoszt-kliens struktúrában. Természetesen egyszerre több klienst is ki tud szolgálni a szerver.

Osztálystruktúra

A részecskefizikában használt, adatok elemzésére tipikusan jellemző az esemény alapú feldolgozás. Ennek megfelelően írtam meg az "Esemény" osztályt (Event class). Az Esemény osztály példányosodik minden egyes bejövő adatnál. Ezen a ponton lehet majd lehetőségünk a későbbiekben elhelyezni olyan szűrést, ami például az egyes detektorok adatainak időkülönbségét vizsgálja, és ez alapján korrelációs osztályokat állít fel. Jelenleg minden eseményhez hozzátartoznak az analóg adatok, valamint egy lista néhány kamrával, amik a "Kamra" osztály példányai (Chamber class). A kamrák tartalmazzák a csatornák digitális információit, tehát azokat az információkat, amik a pályaillesztéshez szükségesek lesznek. Ha később több detektort akarunk egyszerre vizsgálni, itt ide kerül egy közbeeső osztály, amely a Detektorok egyedi tulajdonságait tartalmazza, például a geometriát, a trigger információkat vagy az analóg adatokat.

Miután ezek megvannak, a kamrákra megtörténik a "klaszterezés", azaz a klaszterek keresése és azonosítása, ami létrehoz egy-egy klaszterekből álló listát minden egyes kamrához, a "Klaszter" osztály (Cluster class) példányaival. A programban ezekre történik az eseményenkénti egyenesillesztés, és ezeknek az illesztett egyeneseknek a meredeksége lesz az, ami mint információ hozzáíródik a kimeneti fájlhoz.



16. ábra. Az adatkezelő program sémája

6.2. Kliens oldali feldolgozás

A kliens oldalon a célom egy olyan program létrehozása volt, ami

- weben keresztül is használható,
- jórészt platform független,
- könnyen kezelhető és áttekinthető,
- "látványos", abban az értelemben, hogy a számunkra fontos eredményeket rögtön leolvashatóak legyenek, és következtetéseket lehessen belőlük levonni.

A program megírására a Java nyelvet választottam. A Java hasonlóan a C++-hoz szintén egy objektumorientált nyelv, ennek előnyeit kihasználtam a program megírásakor. A nyelv könnyen kezelhető és jó eszközöket kínál a megjelenítéshez, és az interaktív online felületek létrehozásához, továbbá jól használható grafikon megjelenítő csomagok is léteznek hozzá. Én ezek közül a jfreechartot¹² választottam, amely nagyon jól beleillett az általam elképzelt vizualizációs koncepcióba. Az is fontos szempont volt, hogy a program webes környezetbe beágyazva működjön, emiatt egy Java appletet készítettem¹³.

A szerver oldalon futó program egy dedikált gépen fut, ezért ott a kompatibilitási problémák minimálisak, mert tudtam, hogy milyen gépen tervezem futtatni a programot. Más a helyzet az applikációval, hiszen itt a felhasználók sokféle operációs rendszerrel és böngészővel rendelkezhetnek és ki kell tudni szolgálni a lehető legtöbbféle platformot. Épp ezért is tűnt jó választásnak a Java, aminek a Bytecode-ja¹⁴ operációs rendszertől függetlenül futtatható. A fejlesztés során mind Windows, mind Linux alapú operációs rendszereken teszteltem, különböző böngészőkkel (például Firefox, Chrome; a teljes támogatási lista az alkalmazás honlapján [28].).

Az újdonság az eddigi utólagos offline adatfeldolgozási módszerhez képest a valós idejű online feldolgozás lehetősége. Ezt úgy érem el, hogy a szerver oldali feldolgozó alkalmazásnak, valamint egy kliensnek egyaránt hozzáférést biztosítok az adatokhoz. Mivel az egyik irányból csak írás, a másik irányból pedig csak olvasás történik, a két művelet sorrendjét a szerver oldali operációs rendszer irányítja, és természetesen a menetközben a pufferelést is elvégzi.

A kliens oldalán az alkalmazás szintén esemény alapon végzi az adatfeldolgozást, azaz az egyes eseményeket egyesével olvassa be, és eszerint végzi az adatmanipulációt. A legfontosabb kliens oldali feladat az adatokból hisztogramok készítése. Ez akkor jó választás,

¹²A jfreechart egy LGPL licenszelt, szabadon felhasználható Java grafikon készítő könyvtár [27].

¹³A Java applet egy olyan Java nyelvben írt program, ami csak egy adott weboldalba beágyazva működik. Speciális biztonsági megkötései vannak, ugyanakkor a böngészőkkel való együttműködése sokkal jobb, mint egy átlagos Java alkalmazásnak.

¹⁴A Bytecode a Java nyelv platform független futtatható állománya.

ha a program online figyelésre van beállítva, ugyanis ekkor a sávszélesség nem hátráltató tényező. Ugyanakkor, ha összegezni szeretnénk, akkor a program automatikusan átáll a szerver oldalon futó program által negyedórás intervallumokra átlagolt, kisebb méretű fájlokra.

A programnak alapvetően két funkciója van, a valós idejű adatmonitorozás és a már archivált adatokban történő vizsgálódás.

6.3. Online monitorozás

Az online adatelemző alkalmazás fontos módja ez, hiszen ez az, amiben más, mint az eddigi adatelemzési módszer. Ez a mód lehetővé teszi, hogy a detektor működését folyamatosan ellenőrizzük, és ha valamilyen változtatást kell végrehajtani, azt rögtön észrevegyük. Ilyen lehet például, ha az akkumulátor lemerül vagy ha a túl magas páratartalom miatt nem biztosított a hatékony detektálás. Ezek a feszültség vagy páratartalom értékekben azonnal láthatóak, ám ha mondjuk a detektor két hétig egy barlangban mér, akkor lehet, hogy csak napokkal később derül fény erre a hibára, így a mérést meg kell ismételni. Ekkor lehet hasznos ezen változók online, valós idejű, folyamatos nyomonkövetése.

A 17. ábrán a futó alkalmazást láthatjuk. A működésbe való beleszólás korlátozott, csak az egyes változókat tudjuk nyomon követni. Ennél a módnál a sebesség nem állítható, az időkésleltetés úgy történik, hogy az n – edik esemény n-1 – edik esemény óta eltelt idejét állítom be az n+1 – edik esemény késleltetésére. Ez azt jelenti, hogy az események egymást követése egy lépés csúszásban van a valós időhöz képest, ám úgy ítéltem meg, hogy ez nem számottevő hiba mondjuk egy egész napos idősorhoz képest. Jelenleg nincs detektor kötve hálózatra, így csak az archivált adatok elérhetők.



17. ábra. Az alkalmazás online felülete, analóg adatok

6.4. Archivált adatok vizsgálata

A program lehetőséget ad a már korábban lezajlott események visszamenőleges vizsgálatára is. Ekkor egy naptár jellegű táblázatból választhatjuk ki, hogy mely nap eseményei érdekelnek minket és ezeket, hasonlóan mint az online módnál, "kvázi-online", eseményről-eseményre (event-by-event) futtathatjuk, ami azt jelenti, hogy mint egy videófelvételt, lejátszhatjuk, gyorsíthatjuk, lassíthatjuk, megállíthatjuk. Az offline módban készült ábrák láthatóak a 19. képen. A felső menüsorban az irányításért felelős kapcsolók vannak.

- A "Paraméterek" menüpontban lehet kiválasztani, hogy melyik környezeti változót, illetve melyik kamra paramétert rajzoltatjuk ki (triggerelési hatásfok, triggerelt, illetve illesztett események gyakorisága).
- A "Fluxus" segítségével a nézet átvált a pad illetve szál irányú meredekségek hisztogramjaira. Ez, ellentétben a paraméterekkel, kezdettől betöltődik, és a folyamatos feldolgozásnak köszönhetően akkor is kumulálódik, amikor épp nem azt vizsgáljuk.
- A "Play" funkcióval a naptárban kiválasztott fájlt kezdjük el lejátszani. Ha a lejátszást többször nyomjuk meg, mindig újra indul.
- A "Pause" funkcióval felfüggeszthetjük a lejátszást.
- A "Stop" funkcióval megállíthatjuk a lejátszást.
- A "Összegzés" funkcióval azonnal feldolgozhatjuk a fájlt, és megjelenik a teljes idősorra összegzett adatsor, valamint a teljes meredekség hisztogramok.

A fájlokat a lent látható naptár segítségével válaszhatjuk ki. Az eredeti forrást a [29] weboldalról vettem, ezt módosítottam az igényeknek megfelelően. A naptárban azt is láthatjuk, hogy mely napokról vannak részecske adatok, ezek a gombok jelennek csak meg kéken, amely napról nincs adat, azok szürke színűek.

A dátum választó panelen láthatóak még a következő funkciók:

- A "Dátum kiválasztása" gombbal erősíthetjük meg a dátum kiválasztását.
- A "Sebesség" csuszkával gyorsíthatjuk illetve lassíthatjuk a lejátszás sebességét (az alapértelmezett 10ms-onként tölt be egy új eseményt)
- A "Trigger hatásfok" mezőkben pedig a triggerelési minta alapján kiszámolt triggerelési hatásfokok láthatóak, balról a legfölső kamrával kezdve



18. ábra. Az alkalmazás offline felülete, analóg adatok



19. ábra. Az alkalmazás offline felülete, fluxus hisztogramok

6.5. További tervek, kitekintés

Későbbiekben szeretnék a részecskepályák korrelációival is foglalkozni. A 2.6. fejezetben láthattuk, hogy létezik már kozmikus részecske adatsorok online feldolgozása és vizsgálata, ám ott még nem foglalkoztak korrelációkkal, mert inkább az oktatás, és a fiatalok bevonása az elsődleges cél. Lehetséges lenne azonban több detektort együtt hálózatra kötni és hasonlóan a Pierre Auger Obszervatóriumhoz, kozmikus részecskezáporok egyes elemeit megvizsgálni. Ehhez szükség van a detektorok pontos időmérésére, és a megfelelő pontosságú térbeli koordinátákra. Ezután az időbélyeg alapján lehet megállapítani, hogy mely részecske pályák tartoznak ugyanahhoz a keltő primer részecskéhez.

Ehhez természetesen különféle módosítások szükségesek a programban, mind szerver, mind kliens oldalon, sőt magában a detektorban is. A programokat, ahol csak lehetett úgy írtam meg, hogy ne függjön a detektor tulajdonságaitól, például a kamrák számától, vagy az egyes kamrán található csatornák számától. Így többféle detektort "be tud fogadni" a rendszer. Fejlesztési lehetőségek, szerver oldalon

- Későbbi, korrelációs vizsgálatokhoz szükséges lesz az Event és a Chamber osztály közé beilleszteni egy Detektor osztályt is. Így lehetőség nyílik egy eseményhez több detektort is hozzárendelni. Ekkor első lépésként kiszűröm, hogy melyik detektoroktól származó adatokat kell egy esemény elemeinek tekinteni. Miután ez megtörténik, utána szétosztom a detektoroknak az adataikat.
- Az adatfogadó rész egyelőre egyszerre csak egy kapcsolódási kérést és adatcsomagot tud kezelni. Ha túl hamar érkeznek a következő esemény adatai, akkor azok elvesznek és pufferelésre nincs lehetőség, mert maga a program nem képes egyidejúleg fogadni két beérkező kérést. Ez probléma lehet, hiszen korreláló részecskék – ha közel van a két detektor – valószínűleg mikro- vagy még inkább nanoszekundumos különbséggel jelennek majd meg a detektorokban.
- Az egyenesillesztési és klaszterezési algoritmus hatásfokán is lehet javítani.

Fejlesztési lehetőségek, kliens oldalon

- A naptár nézetben az egyes napokra kattintva részletesebb adatokat is lehetne kapni, mint hogy hol zajlott a mérés, milyen hosszan és így tovább.
- A meredekség hisztogramokat az eddigi két darab egy dimenziós helyett egy darab két dimenziós hisztogramon ábrázolni. Ezzel sokkal átláthatóbbá válnak a meredekség adatok.
- Amint részecske korrelációs vizsgálatokra is képes lesz a szerver oldal, ugyanezt a lehetőséget a kliens oldalnak is meg kell adnia.

7. Összefoglalás

Szakdolgozatom készítése során bekapcsolódhattam a MTA Wigner FK és az Eötvös Loránd Tudományegyetem közös, gáztöltésű detektorok kutatásával és fejlesztésével foglalkozó REGARD csoport munkájába. Feladatom az volt, hogy elkészítsek a csoport egyik fejlesztéséhez, a müontomográfhoz egy olyan online adatfogadó és -feldolgozó keretrendszert, amely szerver-kliens alapon lehetővé teszi a detektor által szolgáltatott adatok online és offline monitorozását valamint analízisét. A feladatot sikeresen elvégeztem, elkészítve egy olyan platformot, ami jó kiindulási alap lehet a későbbi fejlesztésekhez vagy adott esetben más alkalmazásokhoz, online analízishez, detektormegfigyeléshez.

Az általam elkészített alkalmazás a http://people.inf.elte.hu/kadsaat/OnlineApplet/index.html weboldalon érhető el.

A feladat elvégzése során lehetőségem nyílt a csoport más projektjeibe is bekapcsolódni. A 2012 nyarán során egy nagyobb érzékeny felületű detektorkamra építésében segédkeztem, továbbá megismerkedtem a REGARD csoport által épített gáztöltésű detektorok alapvető működésévél.

E munka részeként a csoport DÖME berendezésnél végzett müontomográfiás mérésében is részt vettem, aminek eredményeiből egy cikk is született a Nukleon folyóiratban [4].

A szakdolgozat során elkészített keretrendszer már működőképes, ugyanakkor még jócskán továbbfejleszthető. Sikeres MSc felvételi esetén egy TDK dolgozat keretein belül szeretném folytatni a megkezdett munkát és később a kozmikus részecskék korrelációinak vizsgálatára felhasználni a kidolgozott rendszert. A továbbiakban az ehhez szükséges szoftveres háttér megteremtése lenne az elsődleges célom.

Köszönetnyilvánítás

Szeretném megköszönni családomnak és barátaimnak, hogy segítségemre voltak és támogattak a szakdolgozat elkészítése során.

Köszönet illeti a Wigner Fizikai Kutatóintézetet, az Eötvös Loránd Tudományegyetemet és az Energiatudományi Kutatóntézetet is, mert lehetővé tették a szakdolgozat és az ahhoz kapcsolódó mérések létrejöttét.

Külön köszönet jár továbbá témavezetőmnek Barnaföldi Gergely Gábornak, belső konzulensemnek Varga Dezsőnek, Hamar Gergőnek és Oláh Lászlónak a Wigner FK RMKI dolgozóinak hasznos tanácsaikért, jó meglátásaikért és a számomra nyújtott nagy segítségükért. Szeretném még megköszönni a DÖME mérésben nyújtott segítséget az Energiatudományi Kutatóintézet igazgatójának, Horváth Ákosnak, valamint Belgya Tamásnak, Kis Zoltánnak és Takács Kálmánnak, továbbá a cikkben résztvevő szerzőknek, Melegh Hunor Gergelynek és Surányi Gergelynek, és az ELTE részéről Horváth Ákosnak a közreműködésüket.

Kutatásunkat az OTKA NK778816, NK106119, H07-C 74164, és K104260 pályázatok, valamint részben az OTKA-KTIA 77719, 77815, 77816, NK 106119, NIH TET 10-1 2011-0061, ZA-15/2009 és NAP VENEUS (OMFB-00184/2006) pályázatok támogatták. Té-mavezetőim BGG és VD külön köszönetet mondanak az MTA Bolyai János tudományos kutatási ösztöndíjának.

Hivatkozások

- D. Varga, G. Hamar, G. Kiss, Gy. Bencédi Close Cathode Chamber: Low material budget MWPC, Nucl. Instrum. Meth. A 698 (2013) 11-18;
- [2] G. G. Barnaföldi, G. Hamar, H. G. Melegh, G. Surányi, L. Oláh, D. Varga, Portable Cosmic Muon Telescope for Environmental Applications, Nucl. Instrum. Meth. A 689 (2012) 60-69;
- [3] Barnaföldi G. G. és mtsai., Kincskeresés kozmikus müonokkal, Fizikai Szemle LXI 12. (2011) 401.;
- [4] Kálmán D. és mtsai., Kozmikus müonok elnyelődése vas és ólom abszorbensben, Nukleon V. évf. 5 (2012) 1-5;
- [5] V. Hess, Penetration Radiation in Seven Free Balloon Flights, PZ (1912) 13;
- [6] http://top.gae.ucm.es/auger/Images/auger_design.jpg
- [7] https://kicp.uchicago.edu/depot/images/auger_surface_array_large.jpg
- [8] Pierre Auger Collaboration , Properties and performance of the prototype instrument for the Pierre Auger Observatory, Nucl. Instrum. Meth. A 523 (2004) 50-95;
- J. A. Abraham et al. [Pierre Auger Collaboration], Operations of and Future Plans for the Pierre Auger Observatory, arXiv:0906.2354 [astro-ph.IM].
- [10] T. DeYoung [IceCube Collaboration], Nucl. Instrum. Meth. A 692 (2012) 180

- [11] http://newscenter.lbl.gov/wp-content/uploads/IceCube-schema.jpg
- [12] I. I. Yashin et al., Muon diagnostics of magnetosphere and atmosphere of the Earth, Proc. 20th ECRS, Lisboa, (2006);
- [13] A. Petrukhin, Muon diagnostics of near-terrestrial space, Round Table 2 Italy-Russia@Dubna on Space Physics and Biology (2010);
- [14] L. W. Alvarez et al., Search for Hidden Chambers in the Pyramids, Science 167 (1970) 832-639;
- [15] K. Nagamine et al., Method of probing inner-structure of geophysical substance with the horizontal cosmic-ray muons and possible application to volcanic eruption prediction, Nucl. Instrum. Meth. A 523 (2004), 50-95;
- [16] H. Tanaka et al.,

High resolution imaging in the inhomogeneous crust with cosmic-ray muon radiography: The density structure below the volcanic crater floor of Mt. Asama, Japan, Earth and Planetary Science Letters 263 (2007), 104-113;

- [17] K. N. Borozdin, L. J. Schultz et al, Radiographic Imaging with Cosmic Ray Muons, Nature 422 (2003), 277
- [18] Pierre van Baal, Jan-Willem van Holten, HiSPARC
 EuroPhysicsNews Volume 35, Number 1 (2005) 26-27;
- [19] http://www.hisparc.nl/typo3temp/pics/33ccba67b7.jpg
- [20] L. Oláh et al.,
 Cosmic Background Measurements at a Proposed Underground Laboratory in Felsenkeller (Dresden, Germany) by the REGARD Muontomograph, Nuclear Physics in Astrophysics VI, Lisboa, Portugal (2013);
- [21] G. Charpak,
 Evolution of the Automatic Spark Chambers,
 Annual Review of Nuclear Sciences 20 (1970), 195-254;
- [22] Kiss G.,

Sokszálas proporcionális kamrák fejlesztése részecskefizikai detektorokhoz, BSc Szakdolgozat, ELTE TTK Komplex Rendszerek Fizikája Tanszék (2010)

[23] Melegh H. G.,

Müontomográf adatgyűjtó berendezésének fejlesztése, Önálló laboratórium 1-2. dolgozat, BME Irányítástechnika és Informatika Tanszék, (2011);

[24] L. Malqvist et al,

Theoretical studies of in-situ rock density determinations using underground cosmicray muon intensity measurements with application in mining geophysics, GEOPHYSICS 44 (1979), 9, 1549-1569;

[25] Oláh L.,

Földalatti üregek vizsgálata kozmikus részecskék segítségével, BSc szakdolgozat, ELTE TTK Komplex Rendszerek Fizikája Tanszék, (2010);

[26] Oláh L.,

Szerkezetvizsgálat kozmikus részecskék segítségével, MSc diplomamunka, ELTE TTK Komplex Rendszerek Fizikája Tanszék, (2012);

- [27] http://www.jfree.org/jfreechart/
- [28] http://people.inf.elte.hu/kadsaat/OnlineApplet/index.html
- [29] http://javacodespot.blogspot.hu/2010/05/java-calendar.html

Függelék

Forráskód

Java Applet

import java.applet.Applet; import java.awt.BorderLayout; import java.awt.BorderLayou import java.awt.Graphics; import java.awt.Graphics2D; import java.awt.Color; import java.io.File; import java.net.URL; import java.util.Calendar; import java.util.Hashtable; import javax.swing.JButton; import javax.swing.JOptionPane; import javax.swing.JApplet; import javax.swing.JLabel; import javax.swing.JLayeredPane; import javax.swing.JList; import javax.swing.JPanel; import javax.swing.JRootPane; import javax.swing.JScrollPane; import javax.swing.JSplitPane; import javax.swing.JTextField; import javax.swing.ListSelectionModel; import javax.swing.SpringLayout; import javax.swing.SwingUtilities; import javax.swing.Timer; import javax.swing.JSlider; import java.awt.event.ActionListener; import java.awt.event.ActionEvent; import javax.swing.event.ChangeListener; import javax.swing.event.ChangeEvent; import javax.swing.JPopupMenu; import javax.swing.JMenuItem; import java.awt.event.*; import java.awt.*; import org.jfree.chart.ChartFactory; import org.jfree.chart.ChartPanel; import org.jfree.chart.ChartUtilities; import org.jfree.chart.JFreeChart; import org.jfree.chart.plot.PlotOrientation; import org.jfree.chart.plot.XYPlot; import org.jfree.data.xy.XYSeries; import org.jfree.data.xy.XYSeriesCollection; public class OnlineApplet extends JApplet { private static final long serialVersionUID = 1L; //Megjelenítés elemei public JPanel track_panel; public JPanel panel2; public JPanel plot; public JPanel rács; //Interakció elemei public JButton play; public JButton cumulate; public JButton save; public JButton pause; public JButton stop; public JSlider csuszka; public JButton online_button;

```
public int sebesseg = 1;
public static double hour;
public static boolean search = true;
public static boolean trigger = false;
public static boolean online = false;
public static String date = new String("42_42_42");
public static String date_hist = new String("137_137");
public static String date_mhist = new String("137_42");
public static String present_date = new String("42_42_42");
public static String present_date_hist = new String("137_137");
public static String present_date_mhist = new String("137_42");
public JPopupMenu popup;
public JButton menu;
public JButton 11;
public JButton 12;
public JButton 13;
public JButton 14;
public JButton date_choose;
public JButton calendar;
public JButton switcher;
//Feldolgozás elemei
private int dim=7;
public Timer timer;
public Timer online_timer;
public static URL url;
public Event event:
public int event_num=1;
public boolean tick bool;
public boolean update_bool=true;
public JFreeChart chart:
public XYPlot chart_plot;
public JFreeChart hist_pad_chart;
public XYPlot hist_pad_plot;
public JFreeChart hist_wire_chart;
public XYPlot hist_wire_plot;
public JFreeChart saved_chart;
//public ChartPanel plot;
public double data_sum;
public int bin_count;
public int mode=3;
public double Dt=0, bin_time=10 , bin=0;
public double T_elapsed=0;
public XYSeries data_point;
public XYSeries hist_point_pad;
public XYSeries hist_point_wire;
public XYSeriesCollection dataset;
public XYSeriesCollection hist_pad_set;
public XYSeriesCollection hist_wire_set;
public int datamax = 50;
public String[] y_axis_felirat = {"ID","Frekvencia","valami","Hõmérséklet","Páratartalom", "valami", "Feszültség"};
public static int chamber_number = 5;
JPanel content_pane;
public ChartPanel plot_chart_panel;
public ChartPanel pad_chart_panel;
public ChartPanel wire_chart_panel;
public JPanel trigger_panel;
public boolean hist_view = false;
JLabel csuszka_label = new JLabel("Sebesség");
public int pic_num = 0;
public static JTextField[] trigger_fields = new JTextField[chamber_number] ;
public double[] trigger_eff = {0,0,0,0,0,0};
public int slope_bin = 50;
public URL base;
public JLayeredPane layered_pane;
JPanel date_choose_panel;
public int date_year;
public int date_month;
public int date_day;
JPanel csuszka_panel = new JPanel();
JList year_list;
JList month_list;
Solid add list;
String[] years = {"12","13","14","15","16"};
String[] months = {"01","02","03","04","05","06","07","08","09","10","11","12"};
String[] days = new String[31];
/*-----*/
```

public void start(){ //Az ablak beállítása

this.resize(800,600); content_pane = new JPanel(); content_pane = (JPanel)this.getContentPane(); this.setBackground (Color.white); BorderLayout mainLayout = new BorderLayout(); mainLayout.maximumLayoutSize(content_pane); content_pane.setLayout(mainLayout); //layered_pane = this.getLayeredPane();

//A léptetéshez szükséges timer beállításai

timer = new Timer(10, new ActionListener() {
 public void actionPerformed(ActionEvent evt) {
 tick(mode);
 }
}):

timer.setRepeats(true);

online_timer = new Timer(10, new ActionListener() {
 public void actionPerformed(ActionEvent evt) {
 onlineTick(mode);
 }
});
online_timer.setRepeats(true);

//Panelek beállítása

rács = new JPanel(); date_choose_panel = new JPanel();

//GridLayout grid = new GridLayout(1,2); rács.setLayout(new GridLayout(1,1)); rács.setBackground(Color.white);

//Alpanelek beállítása

//Kezelöfelület
trigger_panel = new JPanel();
JLabel trigger_label = new JLabel("Trigger hatásfok");

pane12 = new JPane1(); pane12.setBackground(Color.blue); play = new JButton("Play"); save = new JButton("Kép mentése");

stop = new JButton("Stop"); pause = new JButton("Pause"); menu = new JButton("Paraméterek"); cumulate = new JButton("Öszegzés"); date_choose = new JButton("Dátum kiválasztása"); calendar = new JButton("Naptár"); switcher = new JButton("Fluxus"); csuszka = new JSlider(1,3); online_button = new JButton("Offline");

Hashtable labelTable = new Hashtable();

labelTable.put(new Integer(3), new JLabel("10")); labelTable.put(new Integer(2), new JLabel("1")); labelTable.put(new Integer(1), new JLabel("1/10"));

csuszka.setLabelTable(labelTable); csuszka.setPaintTicks(true); csuszka.setPaintLabels(true);

//Kezelöfelület: legördülö menü popup = new JPopupMenu("Legördül"); JMenuItem 11 = new JMenuItem("Hömérsékletek"); JMenuItem 12 = new JMenuItem("Páratartalom"); JMenuItem 13 = new JMenuItem("Feszültség"); JMenuItem 14 = new JMenuItem("Frekvencia"); Dimension panel2_dim = new Dimension(this.getWidth(),50); panel2.setPreferredSize(panel2_dim);

//panel2.setLayout(grid_fenn);

panel2.add(menu);
//panel2.add(save);
panel2.add(play);
panel2.add(play);

panel2.add(stop); panel2.add(cumulate); panel2.add(switcher); panel2.add(online_button); csuszka_panel.add(csuszka_label); csuszka_panel.add(csuszka); trigger_panel.add(trigger_label); crigger_pane.aut(trigger_iduel); for(int i = 0; i < chamber_number ; i++) { trigger_fields[i] = new JTextField("0.000"); trigger_fields[i].setPreferredSize(new Dimension(50,20)); trigger_panel.add(trigger_fields[i]); 3 date_choose_panel.setLayout(new GridLayout(1,2)); date_choose_panel.setPreferredSize(new Dimension(200,170)); content_pane.add(date_choose_panel, BorderLayout.PAGE_END); content_pane.add(panel2,BorderLayout.PAGE_START); JPanel panelke = new JPanel(); panelke.setSize(new Dimension(100,100)); panelke.setBackground(Color.green); panelke.setLayout(new FlowLayout());
//Celendar.date_set(date_choose_panel); base = getCodeBase(); Days d = new Days(base); date_choose_panel.add(d); date_choose_panel.add(panelke); panelke.add(trigger_panel); panelke.add(csuszka_panel);
panelke.add(date_choose);
//panelke.add(calendar); //Legördülő menü hozzáadása panel2.add(popup); popup.add(11);
popup.add(12); popup.add(13); popup.add(14); //Az ábrázolt adatok deklarálása data_point = new XYSeries("XYGraph"); data_point.setMaximumItemCount(datamax); dataset = new XYSeriesCollection(data_point); chart = ChartFactory.createXYLineChart("Offline adatok", // Title "Idõ", // x-axis Label y_axis_felirat[mode], // y-axis Label dataset, // Dataset Plot Orientation.VERTICAL, // Plot Orientation false, // Show Legend true, // Use tooltips false // Configure chart to generate URLs?); chart_plot = chart.getXYPlot(); hist_point_pad = new XYSeries("PadSlopeHist"); hist_point_wire = new XYSeries("WireSlopeHist"); for(double i = -1.53; i < 1.53; i+=0.06){</pre> hist_point_pad.add(i,0); hist_point_pad.add(i+0.06,0); hist_point_wire.add(i,0); hist_point_wire.add(i+0.06,0); hist_pad_set = new XYSeriesCollection(hist_point_pad); hist_wire_set = new XYSeriesCollection(hist_point_wire); hist_pad_chart = ChartFactory.createXYLineChart("Meredekség hisztogram", // Title "Pad irányú meredekség", // x-axis Label "Gyakoriság", // y-axis Label hist_pad_set, // Dataset Plot Orientation.VERTICAL, // Plot Orientation false, // Show Legend
true, // Use tooltips false // Configure chart to generate URLs?

hist_wire_chart = ChartFactory.createXYLineChart("Meredekség hisztogram", // Title

);

"Szál irányú meredekség", // x-axis Label "Gyakoriság", // y-axis Label hist_wire_set, // Dataset PlotDrientation.HORIZONTAL, // Plot Orientation false, // Show Legend true, // Use tooltips false // Configure chart to generate URLs?);

//hist_pad_plot = hist_pad_chart.getXYPlot(); plot_chart_panel = new ChartPanel(chart); pad_chart_panel = new ChartPanel(hist_pad_chart); wire_chart_panel = new ChartPanel(hist_wire_chart); //plot = new ChartPanel(chart);

//Az első event létrehozása
event = new Event(url,dim,chamber_number);
plot_chart_panel.getChart().getXYPlot().getRenderer().setSeriesStroke(0, new BasicStroke(3.0f));
pad_chart_panel.getChart().getXYPlot().getRenderer().setSeriesStroke(0, new BasicStroke(3.0f));
wire_chart_panel.getChart().getXYPlot().getRenderer().setSeriesStroke(0, new BasicStroke(3.0f));

//Ábrázolás elemei

//g.drawString("Event"+" "+event_num,100,300); //g.drawString("Temp"+" "+event.data[4],120,320); /*plot = new JPanel(){ private static final long serialVersionUID = 1L;

public void paint(Graphics g){
if(event_num!=0){
//Plot kirajzolás

chart_plot.getRangeAxis().setRange(data_point.getMinY()*0.99,data_point.getMaxY()*1.01); chart.setAntiAlias(true);

if (chart_plot !=null) {
 chart_plot.draw((Graphics2D)g,plot.getBounds(),null,null, null); //repaints the whole chart
 }
}

};
};*/
//plot.setPreferredSize(plot_size);

//Az alpanelek hozzáadása

track_panel = new JPanel(); track_panel.setSize(new Dimension((int)(content_pane.getHeight()*0.8),content_pane.getWidth()));

JPanel small_square = new JPanel();
JPanel big_square = new JPanel();

track_panel.setLayout(new GridLayout(2,2));

big_square.setSize(new Dimension(250,250));
small_square.setSize(new Dimension(75,75));

track_panel.add(big_square);

track_panel.add(wire_chart_panel);

track_panel.add(pad_chart_panel);
track_panel.add(small_square);

rács.add(plot_chart_panel);

content_pane.add(rács,BorderLayout.CENTER);
//Legördülő menü létrehozása

```
//Gombok függvényeinek a megadása
play.addActionListener(new ActionListener() {
    public void actionPerformed (ActionEvent e) {
    try{
        url = new URL(base,date);
        event.open(url);
        }catch (Exception f){
        }
    if (event.open(url)){
```

if (event.open(uri))
data_sum=0;
bin=0;
T_elapsed=0;
Dt=0;
timer.start();

```
data_point.clear();
rács.repaint();
else{
JOptionPane warning = new JOptionPane();
warning.showMessageDialog(OnlineApplet.this, "Erősítse meg dátum választását a dátum kiválasztása gombbal!");
}
}
});
pause.addActionListener(new ActionListener() {
    public void actionPerformed (ActionEvent e) {
    if(timer.isRunning())}
timer.stop();
}
else
timer.start();
     }
});
online_button.addActionListener(new ActionListener() {
      public void actionPerformed (ActionEvent e) {
  if (online == false){
     online = true;
search = true;
      rács.remove(plot_chart_panel);
      timer.stop();
      Calendar rightNow = Calendar.getInstance();
      hour = rightNow.get(Calendar.HOUR_OF_DAY);
System.out.println(hour);
      try{
url = new URL(base,present_date_hist);
      event.open(url);
}catch (Exception f){
      3
      panel2.removeAll();
      chart.setTitle("Online adatok");
      content_pane.remove(date_choose_panel);
      content_pane.remove(panel2);
      repaint();
      panel2.add(menu);
      //panel2.add(save);
      panel2.add(switcher);
panel2.add(online_button);
      panel2.add(popup);
      panel2.add(trigger_panel);
content_pane.add(panel2,BorderLayout.PAGE_START);
      if(hist_view)
rács.add(track_panel);
      else{
      rács.add(plot_chart_panel);
      3
      online_button.setText("Offline");
     online_timer.start();
repaint();
      Ъ
      else {
      online = false;
      online_button.setText("Online");
      chart.setTitle("Offline adatok");
panel2.removeAll();
      panel2.add(menu);
      //panel2.add(save);
panel2.add(play);
      panel2.add(pause);
      panel2.add(stop);
panel2.add(cumulate);
      panel2.add(switcher);
      panel2.add(online_button);
      repaint();
      if(hist_view){
     rács.remove(plot_chart_panel);
rács.add(track_panel);
      3
      else{
      rács.remove(track_panel);
      rács.add(plot_chart_panel);
}if(hist_view==false)
      content_pane.add(date_choose_panel, BorderLayout.PAGE_END);
      repaint();
//System.out.println("Adat nézet");
```

```
}
       }
});
stop.addActionListener(new ActionListener() {
    public void actionPerformed (ActionEvent e) {
    timer.stop();
event_num=0;
event.stopReadIn();
3
;
});
cumulate.addActionListener(new ActionListener() {
    public void actionPerformed (ActionEvent e) {
    timer.stop();
       event.stopReadIn();
       data_sum=0;
bin=0;
T_elapsed=0;
Dt=0;
try{
      base = getCodeBase();
url = new URL(base,date_hist);
       event.open(url);
} catch (Exception evt){
System.out.println("HAJJAJ");
       data point.clear();
       data_point.setMaximumItemCount(100000);
try{
tick_bool=event.forcedReadIn(data_point, mode+1);
url = new URL(base, date_mhist);
event.open(url);
tick_bool=event.forcedHistReadIn(hist_point_wire, hist_point_pad);
} catch (Exception baj){
System.out.println("Error in Forced Read In!");
3
racs.repaint();
for(int i = 0; i < chamber_number ; i++ ){</pre>
trigger_fields[i].setText(String.format("%.3g%n ",event.trigger_eff[i]));
bin_count=0;
}
});
date_choose.addActionListener(new ActionListener() {
       public void actionPerformed (ActionEvent e) \{
      public void action of the mode (.ect-action of the mode (.ect-action of the mode (.ect-action of the mode);
//int month = month_list.getSelectedIndex();
//int day = day_list.getSelectedIndex();
//date = years[year]+"_"+months[month]+"_"+days[day]+".txt";
       try{
      base = getCodeBase();
url = new URL(base,date);
} catch (Exception evt){
       System.out.println("HAJJAJ");
       //layered_pane.remove(0);
       repaint();
}
});
switcher.addActionListener(new ActionListener() {
       public void actionPerformed (ActionEvent e) {
    if (hist_view == false){
        hist_view = true;
       rács.remove(plot_chart_panel);
       content_pane.remove(date_choose_panel);
switcher.setText("Adatok");
       rács.add(track_panel);
       repaint();
       //System.out.println("Fluxus nézet");
```

```
else {
             hist_view = false;
switcher.setText("Fluxus");
               rács.remove(track_panel);
              rács.add(plot_chart_panel);
if(online==false)
               content_pane.add(date_choose_panel, BorderLayout.PAGE_END);
              repaint();
              //System.out.println("Adat nézet");
              }
}
});
menu.addMouseListener(new MouseAdapter() {
public void mouseClicked (MouseEvent e) {
  popup.show(e.getComponent(), 0, 10);
});
 l1.addActionListener(new ActionListener() {
    public void actionPerformed (ActionEvent e)
{
mode=3;
data sum=0:
bin=0;
Jan State Sta
data_point.clear();
chart_plot.getRangeAxis().setLabel(y_axis_felirat[mode]);
repaint();
3
);
12.addActionListener(new ActionListener() {
          public void actionPerformed (ActionEvent e)
{
mode=4;
data_sum=0;
bin=0;
T_elapsed+=Dt;
lata_point.clear();
chart_plot.getRangeAxis().setLabel(y_axis_felirat[mode]);
repaint();
3
}
);
 13.addActionListener(new ActionListener() {
    public void actionPerformed (ActionEvent e)

ſ
              mode=6:
             data_sum=0;
bin=0;
T_elapsed+=Dt;
 data_point.clear();
chart_plot.getRangeAxis().setLabel(y_axis_felirat[mode]);
repaint();
3
);
 14.addActionListener(new ActionListener() {
    public void actionPerformed (ActionEvent e)
{
             mode=1:
data_sum=0;
bin=0;
T_elapsed+=Dt;
data_point.clear();
chart_plot.getRangeAxis().setLabel(y_axis_felirat[mode]);
repaint();
);
csuszka.addChangeListener(new ChangeListener() {
  public void stateChanged(ChangeEvent e){
if(csuszka.getValue()==1)sebesseg=100;
  if(csuszka.getValue()==2)sebesseg=10;
```

ł

```
//if(sebesseg<0)sebesseg=Math.abs(1/sebesseg);
//if(sebesseg==0)sebesseg=1;</pre>
 });
}
tick_bool=event.readIn();
if (mode==1)
data_sum+=1000000/event.data[mode];
else
data_sum+=event.data[mode];
Dt += ev ent . data [1] /1000000;
++event_num;
++bin;
hist_set();
pad_chart_panel.repaint();
wire_chart_panel.repaint();
if(trigger==true){
for(int i = 0; i < chamber_number ; i++ ){</pre>
trigger_eff[i] +=event.data[12+i];
}
}
,
try{
if (Dt > bin_time) {
data_point.add(T_elapsed/3600,data_sum/bin);
T_elapsed+=Dt;
data_point.add(T_elapsed/3600,data_sum/bin);
Dt =0:
if (trigger==true) {
for(int i = 0; i < chamber_number ; i++ ){</pre>
trigger_fields[i].setText(String.format("%.3g%n ", trigger_eff[i]/event_num));
3
3
bin_count++;
data_sum=0;
bin=0:
rács.repaint();
}
} catch (Exception e) {
}
if(tick_bool==false) {
data_point.add(T_elapsed,data_sum/bin);
T_elapsed+=Dt;
data_point.add(T_elapsed, data_sum/bin);
Dt =0;
bin_count++;
data_sum=0;
bin=0;
timer.stop();
event.stopReadIn();
rács.repaint();
}
}
public void onlineTick(int mode){
online_timer.setDelay(1000);
tick_bool=event.onlineReadIn();
pad_chart_panel.repaint();
price control of print();
wire_chart_panel.repaint();
if (trigger==true) {
for(int i = 0; i < chamber_number ; i++ ){
trigger_eff[i] +=event.data[12+i];
}
if (tick_bool==true)
try{
data_point.add(event.data[0],event.data[mode+1]);
data_point.add(event.data[1],event.data[mode+1]);
Dt =0;
if (trigger==true) {
for(int i = 0; i < chamber_number ; i++ ){</pre>
trigger_fields[i].setText(String.format("%.3g%n ", trigger_eff[i]/event_num));
}
```

if(csuszka.getValue()==3)sebesseg=1;

```
}
rács.repaint();
} catch (Exception e){
}
if(tick_bool==false) {
online_timer.stop();
event.stopReadIn();
rács.repaint();
}
```

}

```
public void hist_set(){
    double slope = event.data[10];
    if (slope != 42){
    int bin_number=(int)Math.round(slope/0.06)+25;
    double y_temp = hist_point_pad.getY(2*bin_number-1).doubleValue();
    hist_point_pad.updateByIndex(2*bin_number-1, y_temp + 1 );
    hist_point_pad.updateByIndex(2*bin_number-2, y_temp + 1 );
    slope = event.data[11];
    if (slope != 42){
    int bin_number=(int)Math.round(slope/0.06)+25;
    double y_temp = hist_point_wire.getY(2*bin_number-1).doubleValue();
    hist_point_wire.updateByIndex(2*bin_number-1, y_temp + 1 );
    hist_point_wire.updateByIndex(2*bin_number-1, y_temp + 1 );
    hist_point_wire.updateByIndex(2*bin_number-2, y_temp + 1 );
}
```

} };