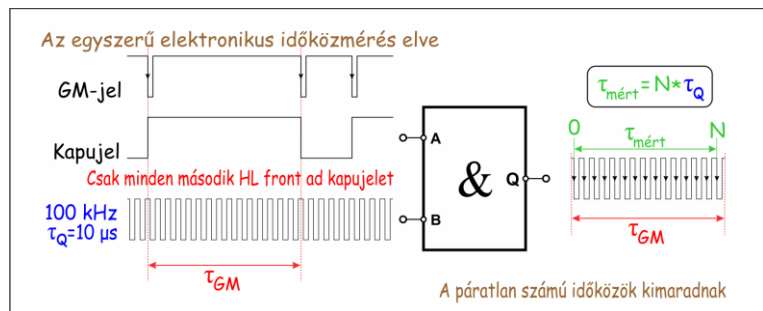


A GM cső holtidejének meghatározása CNC időközmérővel¹

Bartos-Elekes István, ADY Endre Líceum, Nagyvárad, beistvan@yahoo.com
Raics Péter Pál, Debreceni Egyetem, Kísérleti Fizikai Tanszék, raics.peter@science.unideb.hu

A radioaktív bomlás és a háttérsugárzás beütési időközei meghatározására egy CNC² időközmérőt építettem, és egy gyári Geiger-Müller csöves sugármérőhöz csatlakoztattam. Az időközmérés alapegysége a belső ciklusidő ($<1 \mu\text{s}$), ezt a PC a GM cső által adott jelközök figyelési ciklusára használja. A mért időközt ($\tau[\text{ms}]$) a belső ciklusok számával fejezzük ki, egy DWord³-ös vektorban tároljuk, és időnként elmentjük egy állományba. Az eseménynek a megfigyelés kezdetétől számított $t[\text{h}]$ időpontját az addigi DWord-ök összege adja meg. A többmillió beütést rögzítő méréseket ábrázolva⁴, a $\tau[\text{ms}]=f(t[\text{h}])$ „eseménynaptárt” kapjuk, de ez nem jó semmire. Ha az eseménynaptárt nagyon keskeny, vízszintes időközcsíkokba „vágjuk”, és meghatározzuk az ott található események számát, az időközök Poisson-eloszlási görbéjéhez jutunk. A 0,25 ms körüli „letérés” a GM cső holtideje alatt becsapódott részecskék elvesztése miatt van, ebből a „hiányból” meghatározható a GM cső holtideje. A mérésekre a Debreceni Egyetem Kísérleti Fizikai Tanszékén került sor. Az említett képek bemutatására később kerül sor.

Az elektronikus időközmérés elve. Az elektronikus időközmérők a mérendő jelek közötti időt egy kvarcetalonból származó órajel sorozattal töltik ki, majd megszámlálják, hogy hány teljes periódus fért el a mérendő időközben. A mérendő jel minden második frontja egy-egy kapujelet hoz létre (1. ábra), amelynek magas szintje megnyitja, az alacsony szintje pedig lezárja a logikai AND⁵ kaput. A kapu másik bemenetére kerül az órajel (kvarcetalon), a kimenetén pedig megjelenik a kapujel által engedélyezett impulzussorozat. Az $N+1$ számú negatív front között N egész periódus van, így a τ_{GM} időköz mért értéke $\tau_{\text{mért}}=N \cdot \tau_Q$, ahol a τ_Q az órajel periódusa. A mérés során majdnem két teljes órajelet is elveszíthetünk, ezért a legnagyobb feloldási hiba $2/N$ lesz. A kapuidő lejártakor a készülék kijelzi a mért időköz értékét, majd a következő negatív front megjelenésekor újból indul a mérés. Az elektronikus eredetű jelsorozatok időközmérésekor nincs fontossága a kimaradt időköznek, hiszen úgyis átlagolunk, de a változó időközök egyedi mérésére (például a radioaktív bomlások esetében) ez a módszer nem alkalmas, mert minden második időközmérési lehetőséget elveszítünk (nem keletkezik kapujel).



1. ábra. Az elektronikus időközmérés elve

A mérés során majdnem két teljes órajelet is elveszíthetünk, ezért a legnagyobb feloldási hiba $2/N$ lesz. A kapuidő lejártakor a készülék kijelzi a mért időköz értékét, majd a következő negatív front megjelenésekor újból indul a mérés. Az elektronikus eredetű jelsorozatok időközmérésekor nincs fontossága a kimaradt időköznek, hiszen úgyis átlagolunk, de a változó időközök egyedi mérésére (például a radioaktív bomlások esetében) ez a módszer nem alkalmas, mert minden második időközmérési lehetőséget elveszítünk (nem keletkezik kapujel).

A számítógép-vezérelt időközmérés elve. A számítógép-vezérelt időközmérés elvét még 1987-ben dolgoztam ki, amikor megépítettem a ZX81 gépen alapuló kísérleti berendezést a légpárnás ferdesíkon való gyorsuló mozgás tanulmányozására. A mérőrendszer a fotókapu sorozatos takarásai által létrejött jelek változó időközeit mérte, ezekből állapítottam meg a mozgás analitikus egyenletét. A számítógép-vezérelt időközmérés elve (2. ábra) hasonlít az időköz elektronikus

1 A debreceni Magyar Fizikus Vándorgyűlésen bemutatott előadás alapján készült dolgozat - 2013

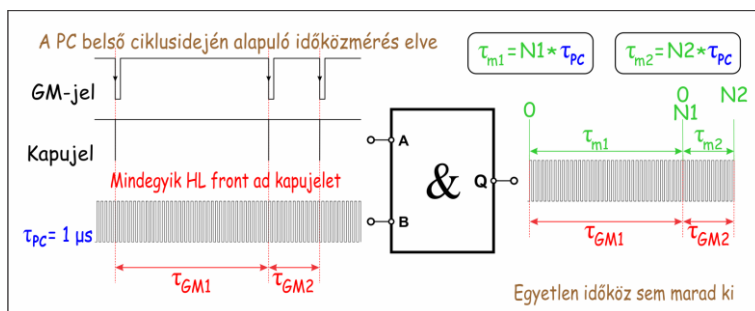
2 CNC – Computer Numerical Control – elfogadott, de van hosszabb magyar kifejezés is: számítógép-vezérelt

3 DWord – Double Word – az assembly nyelv egyik adatformája: Word (egy szó, két byte), DWord, a kettős Word, azaz két szó, négy byte

4 $\tau[\text{ms}]=f(t[\text{h}])$ – az abszcisszán a mérési időpont h -ban, az ordinátán, az ekkor kezdődött bomlási időköz van ms -ban

5 AND – logikai ÉS kapu, a digitális elektronika egyik alapvető egysége. A logikai függvénye: $Q=A \cdot B$, vagy $Q=A \& B$

mérési elvéhez, de van két lényeges, egyben nagyon előnyös különbség. Először is, az **AND** kapu csak szimbolikus, software-esen megoldott logikai egység. A virtuális kapujel a GM-jel negatív frontja megjelenésekor kezdődik, és a következő GM-jel negatív frontja megjelenéséig tart, eközben a PC egy software-es ciklusban „ragadva” várja a GM-jelet, és számlálja az eltelt PC figyelési ciklusokat (*másodpercenként több mint egymillió figyelési ciklus*). A GM-jel megjelenésekor gyakorlatilag azonnal (*néhánytized μ s*) elmenti a figyelési ciklusok számát, és újratekint az egészet. A virtuális kapujel nem egy elektronikus engedélyező jel, hanem a software-es feltétele a két GM-jel közötti belső ciklusszámlálás kezdetének, végének és a következő mérés megkezdésének. A két lényeges előny a kapujel folytonossága és az, hogy a számlálás magától történik, alig költünk rá gép-
időt. A mi esetünkben a mérés feloldóképessége 1 μ s alatt van, ez valójában a belső ciklusidő.



2. ábra. A CNC időközmérés elve

A radioaktív bomlási időköz mérőrendszere⁶. A radioaktív bomlásokból származó részecskék érzékelője egy Geiger-Müller csöves ipari sugárázmérő, amelynek teljes elektronikáját kikerültem, csak a csövet és a tápforrást használtam fel. Oszcilloszkóppal megkerestem a feldolgozásra szánt TTL szintű jelet, egy illesztő interfészen keresztül ezt vezetem a PC-be. A GM-jelek érzékelését egy assemblyben írt mérőrutin végzi el. A mérőrutin leglényegesebb része a 3. ábrán látható. A huszonhat sor két frontérzékelőből (*pozitív és negatív frontok külön-külön, ezeken belül van egy-egy DWord-ös belső ciklusszámláló is*), valamint a mérőciklus-szervezőből és az adatkimentő részből áll. A negatív frontok keresése és az erre „elköltött” belső ciklusok teljesen automatikus számlálása a *@MeasurementHL* nevű ciklusban történik. A **HL**⁷ a negatív frontok időközének mérését jelenti. A ciklusszervezés az első két sorban jön létre, beállítjuk a ciklusszámlálót a **MaxDW**⁸ számú negatív front megkeresésére. A közölt rész előtt már megtalálta az első negatív frontot, így összesen **MaxDW+1** frontot detektálunk. A jel alacsony szinten áll, most a pozitív front keresése következik (6-11 sor). Ez egy figyelési ciklus, amelyben mindig növeljük a **belső ciklusszámlálót**⁹ (*alsó Word, felső Word, 7-8 sor*), majd megnézzük a GM-jel szintjét (9-10 sor). Ha még mindig alacsony, akkor kezdi előlről, visszaugrik a 6. sorba (*ez csak jelölés, a gép tudja, hogy mi a 7. sorra gondoltunk, ezért oda ugrik, nem tölti az időt feleslegesen*). Ha magas a GM-jel szintje, akkor már megtalálta a pozitív frontot, következik a negatív front teljesen hasonló keresése és a figyelési ciklusok számlálása (12-17 sor). Amikor megtalálta a negatív frontot is, kimentti az adatokat (19-20 sor), kinullázza a belső figyelési ciklusok számlálót (21-22 sor). A GM-jel most alacsony szinten van, amennyiben még nem találta meg a MaxDW

```

1. @SetCounter:
2. mov cx,word ptr[MaxDW]
3.
4. @MeasurementHL:
5. mov ah, byte ptr[Mask]
6. @WhileALEqualLow:
7. add bx, 0001h
8. adc si, 0000h
9. in al, dx
10. test al, ah
11. jz @WhileALEqualLow
12. @WhileALEqualHigh:
13. add bx, 0001h
14. adc si, 0000h
15. in al, dx
16. test al, ah
17. jnz @WhileALEqualHigh
18. @FillData:
19. mov ax, bx; stosw
20. mov ax, si; stosw
21. xor bx, bx
22. xor si, si
23. @LoopHL:
24. loop @MeasurementHL
25.
26. @Exit:

```

3. ábra. A mérőrutin

⁶ Ezt a részt csak az assembly programozásban jártasoknak ajánlom

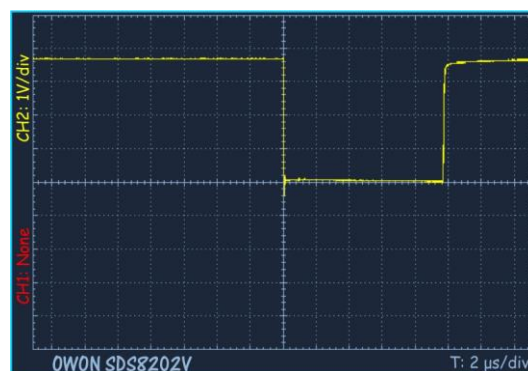
⁷ HL - High→Low a magas logikai szintről az alacsonyra való átmenet, vagyis a negatív front

⁸ MaxDW – Egy dinamikus DWord-ös tömb maximális indexe, ebben MaxDW+1 számú DWord van

⁹ belső ciklusszámlálók – a bomlási időközök legnagyobb értéke s-os nagyságrendű, ezért a μ s nagyságrendű ciklusokat DWord-ben tároljuk

számú frontot, visszaugrik a *@MeasurementHL* nevű ciklus elejére, az újabb negatív front megkeresésére. Ha megtalálta, akkor kilép, egy állományba elmentjük a MaxDW+1 számú időköz-adatot (ezek gépfüggő belső ciklusszámok), ezután, amennyiben a teljes mérési feladatot még nem végeztük el, újabb MaxDW+1 számú időköz-mérés következik. A teljes mérési feladat az esetleg előre meghatározott **Max**¹⁰ számú, fentebb leírt mérési ciklusból áll. Erre a kettős mérési ciklusra azért van szükség, mert amint lentebb látni fogjuk, a mérések alatt (a sugárzás erősségétől függően csak néhány perc) a PC teljesen kezelhetetlen, csak a mérésekkel foglalkozik, olyan, mintha lefagyott volna. Hogy a számítógépet valamennyire is kezelhetővé tegyem, a 26. sori kilépés után megmutatom neki a billentyűzet puffereiben található információkat (az utóbbi 16 billentyűzeteseemény kódjait). Ha a legutolsó billentyűzeteseemény egy **felengedett Esc**¹¹, akkor kilépést kezdeményez, ha nem akkor folytatja, a következő mérési adaggal. A **lenyomott Esc** billentyűre nem engedem reagálni, mert nem tudom pontosan akkor lenyomni, amikor ő figyel a billentyűt, legyen szíves, nézze meg ő, hogy én mit akartam.

A mérőrendszer pontosságának növelése. Első lépésként a gyári készülék GM-jelét adó monostabil áramkört 10 µs-os szélességűre változtattam (4. ábra), ez lényegesen megnövelte az eredetileg három GM csővel rendelkező sugárzásmérő érzékenységet, ugyanis sikerült az egymástól 15 µs-ra becsapódó részecskék biztonságos érzékelése, ha azok nem ugyanabba a GM csőbe csapódtak be. A holtidő meghatározásánál ennek nincs jelentősége, mert ebben az esetben, értelemszerűen csak egy GM csövet használunk. A továbbiakban meg kellett teremtenem a mérést végző számítógép teljes függetlenségét, mert a figyelési ciklus alatt nem szabad „zavarni” a PC-t, ezért erre az időre a gép minden, a méréssel nem kapcsolatos munkáját be kell fejezni, ez azt jelenti, hogy le kell tiltanunk a megszakításokat. A megszakítások letiltása¹² után a PC csak a mérőport megfelelő bitjén megjelenő GM-jellel foglalkozik, illetve számlálja a „sikertelen” figyelési ciklusokat. Ilyenkor a gép halott! Amíg ki nem lép a mérési ciklusból, illetve addig, amíg nem érkezett meg az előre megadott részecskeszám, a gép kezelhetetlen, csak a kikapcsolásra reagál. Cserébe a mérések pontossága két nagyságrenddel nőtt, ezt kvarcetalonokkal ellenőriztem. A 4 s-os impulzusok periódusát **4.209.527** belső ciklussal mérte meg, ez a szám hét számjeggyel állt órákon át, azt hittem, lefagyott a gép. Egy belső ciklus átlagideje: $4\text{ s}/4.209.527=0,95022524\cdot10^{-6}\text{ s}$, felkerekítve $\tau_{PC}=0,950226\text{ }\mu\text{s}$. A megszakítások letiltása nélkül a négymillió ciklusszám ötödik digitje is bizonytalan volt, nem felelt meg a mérési pontosság. A megszakítások letiltása alatt megáll a gép DOS órája is, a megszakítások visszaállításkor ezt figyelembe kell venni. Szerencsére, a CMOS óra tovább jár, így attól át vesszük a helyes időpontot, és beállítjuk a DOS órát is. A megszakítások letiltását csak a DOS (és a DRDOS) engedélyezi, ezért kiegészítettem a szolgáltatásait, és a mai szokásoknak megfelelő, Windowsszerű környezet teremtettem a DOS alatt. A mérések végeredményeként a felhasználó egy Excelszerű grafikont (szabványos BMP kép, lásd később), esetleg több millió sorból álló **txt**, vagy **xls** állományt kap, ezeket tetszés szerint felhasználhatja.



4. ábra. A TTL szintű GM-jel

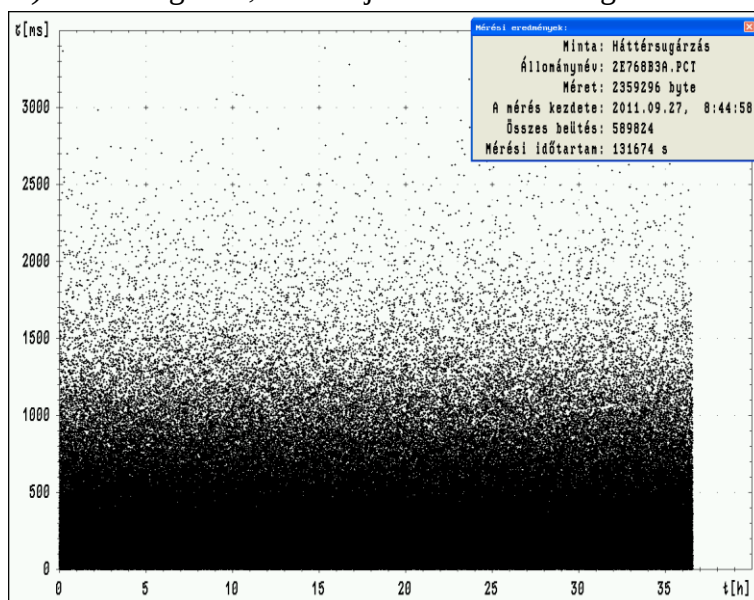
¹⁰ **Max** – a mérési adagok legnagyobb indexének jelölése: az előre megadott, vagy a kilépéskor rögzülő érték

¹¹ **Esc** – Escape billentyű. Ha a billentyűzetet a portjáról olvassuk, akkor a **lenyomott Esc** kódja **01h**, a **felengedetté**, **81h**, tehát megkülönböztethető a billentyű lenyomott és felengedett állapota. Az állapotot jelző kód csak kiolvasással, vagy billentyűművelettel szüntethető meg.

¹² A megszakítások letiltásának assembly mnemonikja (utasítása): **CLI** – Clear Interrupt Flag, a visszaállítása pedig: **STI** – Set Interrupt Flag

A mérési adatok eredeti szerkezete. Az elmentett mérési adatsor a belső ciklusok számával kifejezett bomlási időközökön kívül semmi mást nem tartalmaz, hiányzik a megszokott, legalább kétféle szerkezet (*független változó, és a készülék válasza, a függő változó*), esetleg a mérési sorszámot tartalmazó legelső mező is. Ennek ellenére, ebből az eredeti, egyvezős adatszerkezetből mindhárom mező: a sorszám, az időköz és annak mérési időpontja is megadható. Az időközök ms-ban kifejezett értékét gépállandó segítségével kapjuk meg ($\tau_{PC}=0,950226 \mu s$). Ez az érték erősen gépfüggő, egy adott gép esetében portfüggő, ezen kívül a gép BIOS beállításaitól is függ, ezért nagyon igényes kalibrálás szükséges. A gépállandót kalibrálni kell a PC-re, az abban található portokra, mindezt egy elmentett BIOS beállítás mellett. A gépet a BIOS dátuma alapján azonosítom, két azonos alaplap esetében (*ilyennel még nem találkoztam*) a merevlemez teljesen egyedi gyári sorszáma azonosíthatja a gépet, a portot pedig software-esen választja meg a felhasználó. A legjobb BIOS beállítást a kalibráláskor elmentem, a mérés megkezdésekor a mérőprogram automatikusan összehasonlítja az aktuális értékeket az elmentettel, ha a legkisebb különbséget is tapasztalja, figyelmeztetéssel megáll, betölti az eredeti értékeket, majd újraindítja a gépet (*csak ekkor hatásos a beállítások megváltoztatása*). Mindebből **a felhasználó semmit sem érzékel**, mert a program indításakor egy függvény automatikusan elvégzi a szükséges azonosításokat és visszaadja a gépállandó értékét. Az elmentett állomány **file-pointerének**¹³ állásából az **n** mérési sorszámot lehet megkapni, ennek értéke nulla és $(Max+1) \cdot (MaxDW+1) - 1$ között van, összesen $(Max+1) \cdot (MaxDW+1)$ számú DWord adat van benne. Ha a kiolvasott **n**-edik adat **DW**, akkor ennek megfelelő bomlási időköz értéke $\tau[ms]=DW \cdot \tau_{PC}[\mu s]/1000$. A bomlási időközök mellett, az egymás utáni időközök összegéből megkapjuk a mérési időpontot.

Egy radioaktív bomlás időköz-mérési adatainak feldolgozása. A módszer kidolgozása idején az Excel csak egy Word méretű (65536) adatot fogadott, ezért saját módszerrel dolgoztam fel a milliós adatsoraimat. Az állományok kezelése sajátfejlesztésű assembly függvényekkel történik. A Pascal nyelv rutinjaihoz képest óriási előnye, hogy nincs állománytípus és az adott állományban 2 GB alatt teljes szabadságom van, előre-hátra megyek az állományban, a kezelt részek nagysága nem rögzített, összetett szerkezetű állományokat is írok-olvasok, ezen kívül a file-kezelés során felmerülő szabványos hibákat a környezetnek megfelelő módon jelzi (*automatikusan, a VGA szöveges képernyőtől a VESA-ig*).

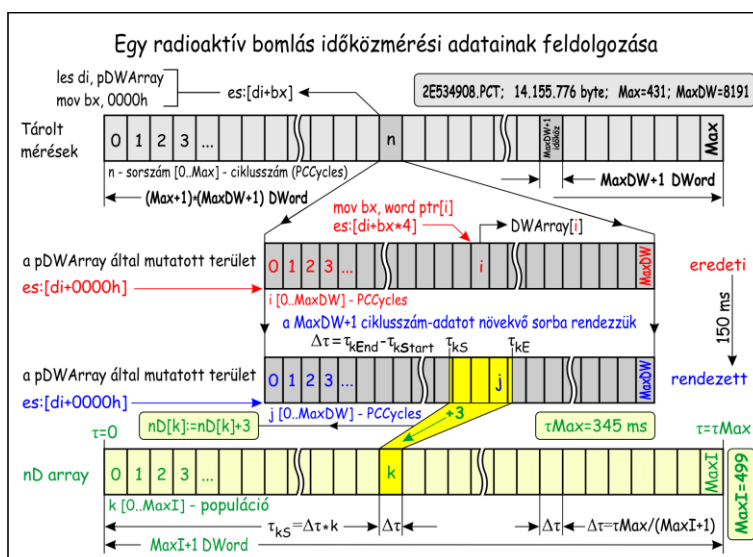


5. ábra. Háttérsugárzás. Több mint 36 óra eseménynaptára

A kísérletező fizikus a mérések befejeztével egy megfelelő grafikonon próbálja ábrázolni a méréseit. Ebben az esetben a $\tau[ms]=f(t[h])$ „görbéről” van szó. A több mint félmillió adat ábrázolása után kiderül, hogy ez nem több egy eseménynaptárnál (5. ábra), látszólag semmire se jó. Annyi talán mégis megállapítható, hogy az időközök 3500 ms és egy nagyon kicsi érték között

¹³ **file-pointer:** egy mutató, amely állomány kezdetétől számított n-edik adatalemre mutat, értéke a magas szintű nyelvekben adatalem egységekben kifejezett sorszám (tömbindex), az általam kifejlesztett file-kezelésnél byte-os szerkezetre mutat és byte-okban van megadva.

változnak, a kisebb értékek felé egyre több hasonló nagyságú időköz van. Az eseménynaptár bármely időpontja köré egy függőleges és keskeny időpontosávot húzunk, majd keresztezzük egy szintén keskeny, de vízszintes időközsávval. Ha megszámláljuk az így kialakult téglalapokba tartozó eseményeket, akkor a vízszintes időközsáv minden időpontosávjában gyakorlatilag ugyanannyi eseményt találunk. Ha azonban egy időpontosáv időközsávjaiban kialakult téglalapokban vizsgáljuk meg az eseményeket, akkor változó számú eseményt találunk, a nagyobb időközök időközsávjaitól a kisebbek felé exponenciálisan nő az események száma. Az 5. ábrán látható grafikont ezek szerint alakítjuk át. Egy keskeny időközsávban megszámláljuk a teljes mérési idő alatti eseményeket, majd folytatjuk a többi időközsávval. Létrejön egy olyan adatbázis, amely egy-egy időközsávba tartozó események számát tárolja egy helyen, de elvesztettük az esemény időpontját, csak az azonos időközsávba való tartozás alapján csoportosítjuk őket. Kialakul az $N = N_{\text{Max}} \exp(-n_v \tau)$ formájú Poisson-eloszlás, ahol N a τ időköznek megfelelő populáció, N_{Max} a populáció maximális értéke, az n_v értelmezését lásd később.

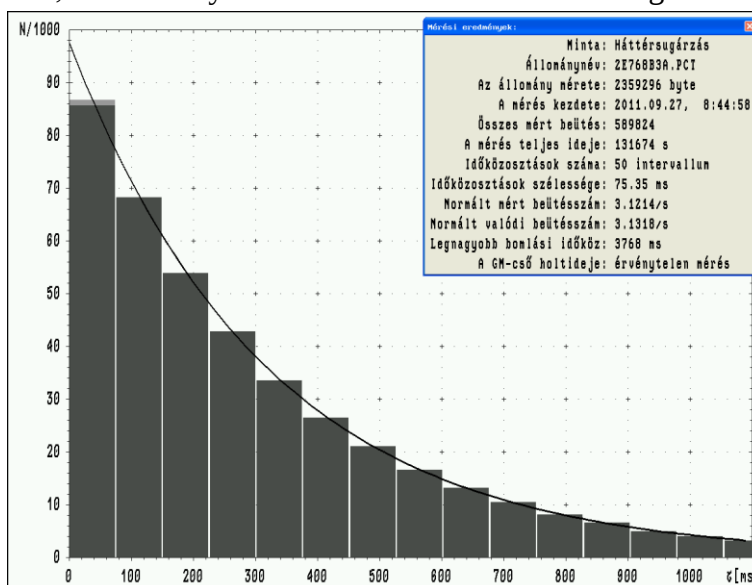


6. ábra. Az uránérc bomlási időközadatainak feldolgozása

volna mérnünk. Sokkal jobb eredményre jutottunk a nagyobb aktivitású, 180 mg-nyi uránércel, amellyel egy fél nap alatt sikerült egy 3,5 milliós adatbázis létrehozni.

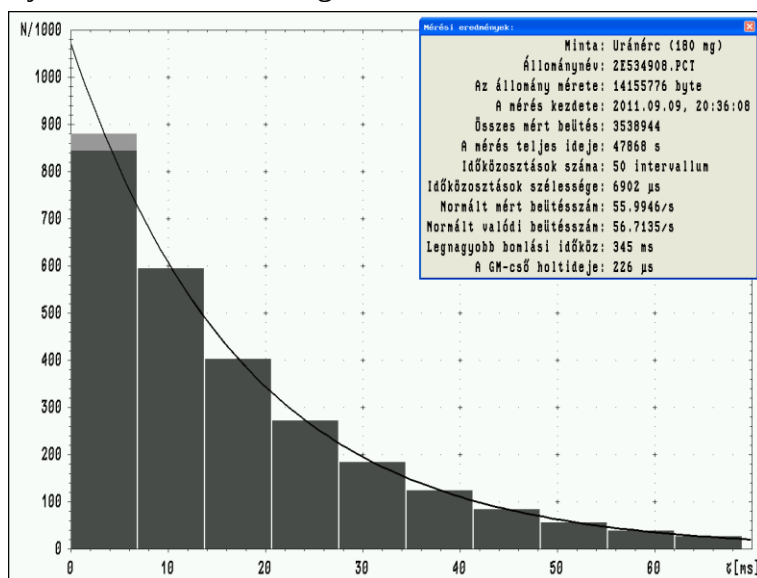
Az adatok számítógépes feldolgozását a 6. ábrán látható diagramok alapján végeztük el. $\text{Max}+1$ számú tárolt mérési adag van, egy-egy adagban egy-egy DWord formájában $\text{MaxDW}+1$ számú időköz-mérési adat található. Az említett file-kezelés lehetőségeit kihasználva a DWArray nevű dinamikus tömbbe beolvassunk egy teljes mérési adagot. A tömbre a pDWArray nevű pointer mutat. Bármikor kiolvashatjuk az i -edik elemet az ott látható kétsoros assembly programocskával. A következőkben az észlelt legnagyobb τ_{Max} bomlási időközt $\text{MaxI}+1$ számú intervallumba osztjuk, egy intervallum szélessége $\Delta\tau = \tau_{\text{Max}}/(\text{MaxI}+1)$ lesz. Létrehozunk egy **nD** tömböt, ennek $\text{MaxI}+1$ számú DWord eleme van, ezek majd a $\Delta\tau$ intervallumnak megfelelő populációt jelölik. A tömb k -adik eleme a τ_{kS} és a τ_{kE} időközértékek közötti populációt tartalmazza. A k -adik elem kezdeti időközértéke $\tau_{kS} = k \cdot \Delta\tau$, a vége pedig $\tau_{kE} = (k+1) \cdot \Delta\tau$. Az **nD** tömb feltöltése érdekében sorba rendezzük a DWArray tömb elemeit, ezek a mért figyelési ciklusok számát tartalmazzák, a rendezés mintegy 150 ms-ig tart. Sorra vesszük a rendezett terület elemeit (*az indexe j*), megállá-

pítjuk, hogy az nD tömb melyik cellájába való, majd a megtalált, k -adik cella tartalmát növeljük eggyel. Mindegyik cella tartozik valahová, a DWArray területen nem marad több feldolgozatlan információ! Az n -edik mérési adat feldolgozása után beolvassuk a következő adatot, és hasonlóan járunk el. A 13 órányi mérésből származó **3.538.944** mérési adat teljes feldolgozása mintegy két percre tart, de ezt csak egyszer kell elvégezni. Szabványosítottam az ajánlott intervallumok számát, ezek a következők: 50, 100, 150, 200, 300, 400, 500, 1000, 1500, 2000, 3000, 4000, 5000. A program az összes ajánlott intervallumra a fentiek szerint feldolgozza a mérési adatokat, és automatikusan el is menti őket. A további feldolgozáskor csak az egyik ajánlott intervallumértéket kell megadni, az ábrázolás szinte azonnali. Eltérő értékeket is meg lehet adni, de ezek jól meggondolt értékek, más intervallumok nemigen hoznak új információt. A 7. ábrán a fenti elgondolás szerint, a 36 órás mérés alapján létrejött Poisson-eloszlási görbe látható. Bár nagyon hasonlít az uránérc görbéjéhez (8. ábra), de a kevés adat miatt (589.824 mérés) nem sikerült meghatározni a holtidőt.



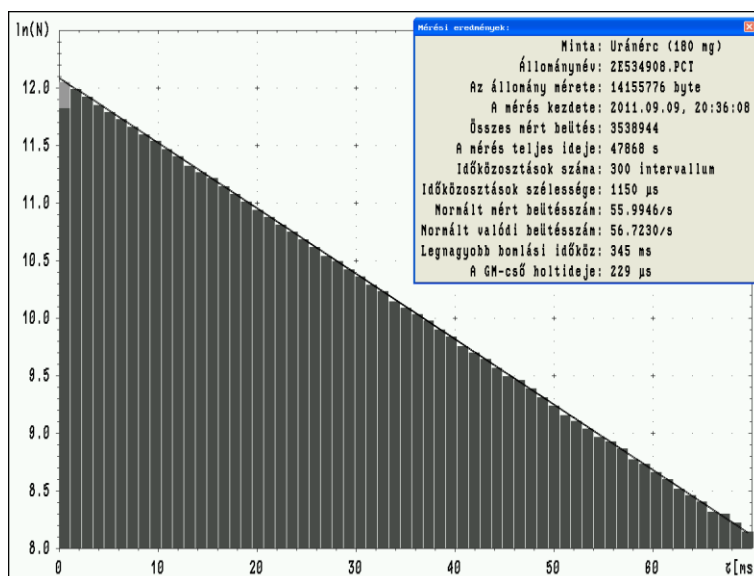
7. ábra. Exponenciális Poisson-eloszlási görbe (háttérsugárzás)

A GM cső holtidejének meghatározása. A 8. ábrán az uránérc sugárzásából származó három és fél millió időköz-mérés alapján létrejött Poisson-eloszlási görbe látható. Az abszcisszán a τ bomlási időköz van ms-ban. A teljes, 345 ms-os időközt 50 intervallumba osztottuk, de a nagy értékek felé az egyes intervallumok populációja nagyon kicsi és főleg nagy a szórás, ezért csak a legnagyobb populáció egy bizonyos hányadáig használjuk fel az adatokat. Az ordinátán az egyes intervallumok populációinak ezredrészét ($N/1000$) ábrázoltuk az intervallum szélességének megfelelő méretű téglalappal. Jól látható a téglalapok magasságának exponenciális változása, kivéve a legelső, ahonnan hiányzik egy rész, ezt kiegészítettük egy világosabb téglalappal. A hiány oka egyértelmű, az exponenciális törvényt átírhatta egy jelenség, a természet nem hibázik ekkorát. Egyedüli megoldásként a GM cső holtideje jöhet szóba, ugyanis a részecske becsapódása után a cső egy darabig nem képes fogadni újabb becsapódásokat, ezek a mérések nem jönnek létre. Ennek a hiánynak az alapján meghatározható a GM cső holtideje. Legyen N_v (v - valódi) a t idő alatt becsapódott összes ré-



8. ábra. Exponenciális Poisson-eloszlási görbe (uránérc)

szecskék száma, N_m pedig a mért részecskék száma. A nagyszámú becsapódás miatt feltételezhetjük, hogy mind a becsapódott, mind a mért részecskék száma arányosan nő az idővel, legyen k az arányossági tényező, tehát felírhatjuk, hogy $N_v = k \cdot t$. Minden mért részecske után a GM cső DT ideig nem mérhet, azaz olyan, mintha a t időből kimaradt volna az $N_m \cdot DT$ idő. A mért részecskék száma is arányosan nő az idővel, vagyis felírhatjuk, hogy $N_m = k \cdot (t - N_m \cdot DT)$. Képezzük az N_v/N_m arányt, egyszerűsítünk k -val, majd megkapjuk a holtidő képletét: $DT = t \cdot (N_v - N_m) / (N_v \cdot N_m)$. Bevezetjük az egységnyi időre normált részecskeszámokat: $n_v = N_v/t$, valamint $n_m = N_m/t$. Aláosztunk t -vel és beírjuk az új változókat, a GM cső holtidejének végső képlete így alakul: $DT = (n_v - n_m) / (n_v \cdot n_m)$. A szakirodalomban hasonló eredményt találtam [1]. Ebben a képletben az egyedüli ismeretlen az n_v , az egységnyi időre normált valódi beütésszám. A fentebb bemutatott $N = N_{\max} \exp(-n_v \tau)$ képletben szereplő n_v -t meghatározhatjuk, ha a 8. ábrán látható mérési sorra egy $N = A \cdot \exp(B\tau)$ formájú exponenciális közelítő görbét fektetünk, majd megfeleltetjük a kitevőben szereplő kifejezéseket: $n_v[1/s] = -B[1/ms] \cdot 1000$. Az ezres szorzó a s -ban kifejezett n_v és ms -ban megadott τ miatt került oda. Az n_m értékét a mért beütések száma és a teljes mérési idő hányadosaként kapjuk meg. Amennyiben az elemzések során néhány mérést kizárunk, akkor az időközzeit is le kell vonnunk a teljes mérési időből.



9. ábra. Linearizált Poisson-eloszlási görbe (uránérc)

9. ábrán látható mérési sorra egy $N = A \cdot \exp(B\tau)$ formájú exponenciális közelítő görbét fektetünk, majd megfeleltetjük a kitevőben szereplő kifejezéseket: $n_v[1/s] = -B[1/ms] \cdot 1000$. Az ezres szorzó a s -ban kifejezett n_v és ms -ban megadott τ miatt került oda. Az n_m értékét a mért beütések száma és a teljes mérési idő hányadosaként kapjuk meg. Amennyiben az elemzések során néhány mérést kizárunk, akkor az időközzeit is le kell vonnunk a teljes mérési időből.

Ha logaritmáljuk az előbbi egyenletet, akkor τ -ban egy egyenest kapunk: $\ln(N) = \ln(N_{\max}) - n_v \tau$, ez az egyenes látható a 9. ábrán. Az első időközsáv populációja kisebb a következő időközsáv populációjánál, mert az időközosztások szélessége csak **1150 µs**, ezért jól látszik a kimaradt részecskék hiánya, ezt egy szürke négyzet jelöli. Itt egy $\ln(N) = a\tau + b$ típusú egyenest illesztünk a mérési sorra. Az előbbihez hasonló módon kapjuk meg az n_v értékét: $n_v[1/s] = -a[1/ms] \cdot 1000$.

A görbék illesztésének gyakorlata. A többmilliós mérési adatokat az időközök szerint csoportosítottuk, de így is rengeteg mérőpontunk maradt. A nagyobb időközértékek felé a szórások és a nagyon kis populációjú időközsávok megnehezítik a közelítő görbék illesztését, ezért egy algoritmus szerint csak a legnagyobb populáció értékének egy előre megadott hányadáig vesszük figyelembe őket. A nagyon sok próbálkozás eredményeképpen ennek értéke **25 és 100** között van, de ez az érték rögzítésre kerül, mert az adatbázis igen enyhe módosítását jelenti. Így megkapjuk az utolsó, még figyelembe vett csoport **LastIndex** nevű indexét. Az exponenciális és a lineáris görbeillesztésnél is fontos, hogy a holtidő miatti kisebb populációjú csoportokat kizárjuk az első illesztésből, hiszen mi azt szeretnénk megtudni, hogy hány tér el a „szabályostól”. Az illesztést egy biztosan jó helyről kezdjük el, majd az ideiglenes görbe ismeretében egy algoritmussal megkeressük azokat az első csoportokat, amelyek populációja egy bizonyos kis hányaddal eltér az ideiglenes görbétől. Értelemszerűen, az első csoport mindig kimarad ebből a keresésből. A keresés eredményeként megkapjuk az első „rendes” csoport indexét, ez a **FirstIndex**. Kezdődhet a

valódi illesztés. Ha kevés mérésünk van, és vannak nagyon nagy, több másodperces bomlási időközök is, akkor a szórás nagyon nagy lesz, és nem sikerül az exponenciális illesztés. Ekkor egy negyedfokú görbével próbálkozunk, ezzel létrehozunk egy új mérési sort, amelyre már könnyedén ráilleszthetjük az exponenciális görbét. A gyakorlat azt mutatja, hogy a görbék szinte teljesen fedik egymást. A szimulált mérések esetén az így megtalált együtthatók a hatodik-hetedik digitig egyeznek. Elég nagy adatbázis esetén, ha nincsenek különösen nagy bomlási időközök, akkor az exponenciális illesztés az első próbálkozáskor sikerül. Mindebből a felhasználó semmit sem vesz észre, a program ezt automatikusan elvégzi, és szinte azonnal megrajzolja az illesztő görbét.

Hibaforrások. Itt rendre felsorolásra kerülnek azok a tényezők és tapasztalatok, amelyek befolyásolhatják a holtidő meghatározási pontosságát. Mivel a számítási képlet mindkét illesztési módszernél ugyanaz, az alábbi megállapítások mindkét módszerre vonatkoznak.

- A holtidő kiszámítási képlete két azonos nagyságrendű szám különbségéből történik. Ebben az esetben a két szám pontosságának olyannak kell lennie, hogy a különbségük legalább három digités pontosságú legyen, ekkor $\pm 1\%$ körüli meghatározási hibát érünk el.
- Az egységnyi időre normált mért beütésszám (n_m) egy rögzített érték, ennek a statisztikus szóráson kívül nincs lényeges hibatartalma, hiszen egyszerűen csak számláltuk, az időmérés hibája pedig igen kicsi.
- Az egységnyi időre normált valódi beütésszámot (n_v) az illesztő görbe alapján határozzuk meg, értéke közvetlenül csak a populáció szórásától, valójában a mérések számától függ. Csak próbából, $\pm 0,01\%$ -kal megváltoztattam a megkapott illesztő görbe kitevőjének az értékét, a holtidő majdnem $\pm 1\%$ -kal megváltozott. Ekkora pontosságú ($\pm 0,01\%$) illesztő görbe meghatározásra csak igen nagyszámú és egyben hosszú idejű mérések esetén van lehetőség. A háttérugárzás esetén ez többhetes, folytonos mérést jelentene.
- Az exponenciális illesztő görbe relatív pontossága jobb 10^{-10} -nél, ez egy sokszorosán kipróbált, majdnem 20 éve használt függvényem (*18-19 digités számokkal dolgozik*). A polinomiális illesztő görbe egykorú az exponenciálissal, elvileg a legjobb közelítést adja, hiszen a legkisebb négyzetek elve alapján jött létre. A meghatározási képlet ugyanaz, a két módszer meghatározásai közötti különbséget a szórások miatti különböző illesztéseknek tulajdonítom.
- Megvizsgáltuk az időközosztások számának hatását. A kiszámított holtidő értéke alig függ az osztások számától (*50 és 1000 között*), pedig előrelátóan 5000-ig dolgoztuk fel az adatokat. A 3,5 millió adat 5000 szakaszba való osztásánál az utolsó 2000 szakasz populációja tízes nagyságrendű, de több száz nullás populáció is van, ezeket kizártuk. Az ezer feletti osztásokat csak a többtíz milliós adatbázisokra lehet majd alkalmazni.
- A még figyelembe vett időközök száma ott áll meg, ahol az illető időközsáv populációja és a legnagyobb populáció hányadosa kisebb az előre megadott **StopQuotient** értékénél. A kis populációjú időközök kizárása csökkenti a mérések szórását és növeli az közelítő görbe együtthatóinak pontosságát. A hányados értéke nem befolyásolta lényegesen a holtidő kiszámított értéket. A méréseket 500-as időközosztásnál végeztük el, 1/10-től 1/100-ig, ötönként növelve a hányados nevezőjét, a holtidő kiszámított értéke 1/10-nél 237 μs , 1/100-nál 240 μs volt, de 1/20 után már nem változott.

Mérési eredmények. A holtidő meghatározása a fenti módszerekkel **208 μ s** és **246 μ s** közötti értékeket adott. A StopQuotient értéke 1/50, rögzített érték volt. Az időközosztást 50 és 1000 közötti ajánlott értékekből választottuk. A mérések feldolgozásakor generált BMP állományok letölthetők a *Fizikai Szemle* honlapjáról [[elérési cím](#)]. A következő eredmények születtek:

- Az exponenciális közelítéssel meghatározott holtidő értéke: **DT=225,8 μ s $\pm$ 3,47 μ s, $\pm$ 1,54%**
- A lineáris közelítéssel meghatározott holtidő értéke: **DT=234,6 μ s $\pm$ 1,90 μ s, $\pm$ 0,81%**
- A mindkét közelítéssel számolt holtidő értéke: **DT=230,2 μ s $\pm$ 2,04 μ s, $\pm$ 0,89%**

Következtetések

- Létrejött egy DOS alatti mérőrendszer, amely rugalmasan alkalmazható többféle időköz nagy pontosságú meghatározására. A mérőrendszer software-esen módosítható, ez sok alkalmazást tesz lehetővé.
- A módszer gépfüggő, de a kalibrálás után, a leírt intézkedésekkel a függőség teljesen kizárható, ebből a felhasználó már semmit sem vesz észre. A néhány száz ms-os időközöket 4-5, a nagyobbakat 5-6 digittal mérhetjük.
- A DOS látszólagos hiányait a Windowsszerű környezet megteremtésével pótoltam. Itt nem tárgyaltam, de ez egy húszezer soros, szinte teljesen assemblyben írt programcsomag.
- A többmilliós mérések feldolgozása lehetséges a DOS alatt is, a mérőprogram nagyon sok lehetőséget kínál.
- A DOS alatti, Excelszerű grafikonok, szemléletesen mutatják a holtidő miatt hiányzó méréseket, ezeket a feldolgozás után BMP szabványú képekként lehet tárolni.
- A mérőprogram az eredeti mérésekkel azonos pontosságú txt vagy xls állományokat generálhat. A gép kalibrációs állandója ismeretében ezek a DWord adatok μ s-ban kifejezett időközökké alakíthatók. A generált állományok mérete, a számábrázolás különbözősége és az elválasztó karakterek miatt majdnem háromszorosa az eredeti állomány méretének.
- A GM cső holtidejének meghatározásánál, a pontosság növelése érdekében többmilliós mérési sorozatokra van szükség, ez nagyon időigényes.
- A bemutatott, az irodalomból ismert mérési módszer csak GM cső holtideje létének bizonyítására és a nagyságrendjének meghatározására alkalmas. A létrejött grafikonok azonban szuggesztíven szemléltetik a holtidő miatt kimarad méréseket.
- Itt nem került bemutatásra, sőt említésre sem, de az általam kifejlesztett mérőrendszer különlegesen alkalmas a rövid felezési idejű, kis aktivitású minták felezési idejének nagy pontosságú meghatározására. Külön érdekessége a lehetőségnek, hogy továbbra is időköz-mérés történik, de a mérés után a CPS mérés software-esen számtalanszor újrajátszható, a virtuális kapuidő megváltoztatható. A rögzített kapuidő helyett rögzített számú időköz mérésére kerül sor, így egyetlen részecske detektálása sem vesz el, a pontosság nagyságrendekkel is megnőhet (ez egyáltalán nem túlzás!).

Irodalom

1. Glenn F. Knoll, **Radiation Detection and Measurement**, Fourth Edition, John Wiley & Sons, Inc., Michigan, 2010, page 122.