

FRAKTÁLOK ZENÉJE

– AVAGY ÉRDEKES JELALAKOK OSZCILLOSKÓPON

Zhang Yu Jie – Budapesti Fazekas Mihály Gyakorló Általános Iskola és Gimnázium

Simon Ferenc – Budapesti Műszaki és Gazdaságtudományi Egyetem, Természettudományi Kar és ELKH Wigner Fizikai Kutatóközpont

Az oszcilloszkópok a fizikaszertár elengedhetetlen kellei a váltóáramú jelek láthatóvá tételére. A demonstrációk egyik kedvelt tárgya a Lissajous-görbék bemutatása. Lissajous-görbék létrejöhetnek például egy x - y síkban mozgó inga esetén, amelyből a lengés során homok folyik ki. Az általunk tárgyalt lehetőség: egy kétcsatornás függvénygenerátor jelét oszcilloszkóp két bemenetére helyezve, majd az oszcilloszkópon az x - y üzemmódot beállítva az 1. ábrán látható jelalakokat kapjuk attól függően, hogy a két csatorna frekvenciája és fázisa miként viszonyul egymáshoz. E kísérlethez a Siglent SDG1025 kétcsatornás vektor-jelgenerátort és Tektronix TBS1042 oszcilloszkópot használtuk.

A cikk elkészültét a Nemzeti Kutatási Fejlesztési és Innovációs Hivatal támogatta a K137852, a TKP2021-EGA-02, továbbá a TKP2021-NVA-02 és a V4-Japán programok által, valamint a Kulturális és Innovációs Minisztérium a Kvantuminformatika Nemzeti Laboratórium projekt (2022-2.1.1-NL-2022-00004) keretében.



Zhang Yu Jie a Budapesti Fazekas Mihály Gyakorló Általános Iskola és Gimnázium 12. osztályos tanulója, általános tantervű tagozaton. Fizikatanárai *Pintér Géza* és *Schramek Anikó*. A BME „Középiskolás Tudományos Diákköri Konferencia” program keretében végez a BME Természettudományi Karán kutatómunkát. A magyar junior gyorskorcsolya válogatott tagja.

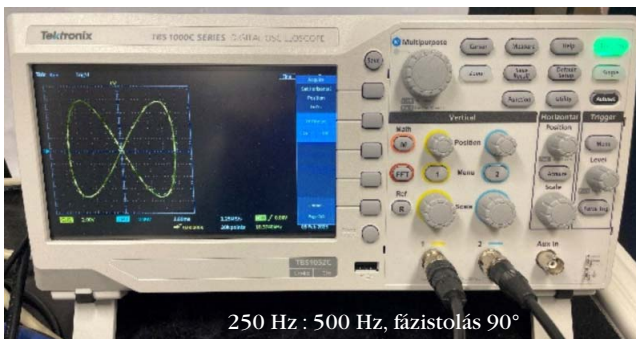
Animált Lissajous-görbéket is előállíthatunk, legkönnyebben úgy, hogy a két csatorna frekvenciája nem azonos, erre mutat egy példát a <https://drive.google.com/file/d/1eVdsku7Gu4wXWNjd7aImUL6dwc-nlSHg/view?usp=sharing> videó (lásd még a QR-kódot). Itt az egyik csatorna frekvenciája 250 Hz, a másiké 501 Hz, a fázistolás pedig 90° volt.



Felmerülhet a kérdés, hogy hasonló módszer segítségével lehetséges-e bonyolultabb jelalakok, akár animációk megjelenítése is az oszcilloszkópon. E kérdésfeltevéshez az inspirációt az adta, hogy a monitorok analóg videojelének átalakításával népszerű videojátékok megjelenítése is lehetséges oszcilloszkópon [2]. Az általunk vizsgált esetekben olyan jelalakot választottunk, ami vektorgrafikusan megjeleníthető, azaz a rajzoló elektronikus „ceruzát” folya-



Simon Ferenc fizikus, egyetemi tanár, a BME TTK dékánhelyettese. Érdeklődési területei: szilárdtest-spektroszkópia, spintronika, kvantuminformáció-elmélet és a fizika népszerűsítése. Legfontosabb eredményei: az itineráns elektronok mágneses rezonancia-jelének felfedezése új fémekben, a spinrelaxáció egyesített elméletének kidolgozása, spinnel nyomjelzett szén-nanocsövek előállítása és tripllett optikai állapotok felfedezése nanocsövekben. ERC és Lendület-pályázat vezetője.



1. ábra. Lissajous-görbék oszcilloszkóp képernyőjéről fényképezve a különböző x - y csatornára tett frekvenciaarányok és fázistolások függvényében. A mutatott görbék természetesen számítógéppel is szimulálhatók, azonban érdekesebb az oszcilloszkóp képernyőjén látható jelalakokat közvetlenül fényképezni.

matos vonal mentén tudjuk mozgatni. E demonstrációra a fraktálokat, mint érdekes geometriai objektumokat választottuk.

Fraktálok

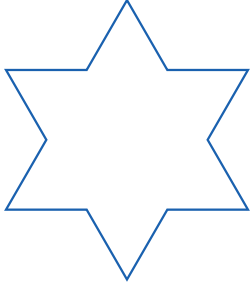
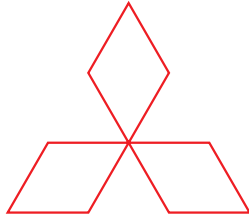
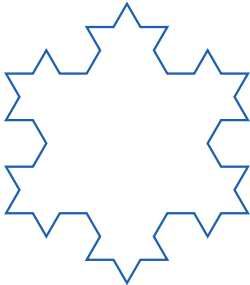
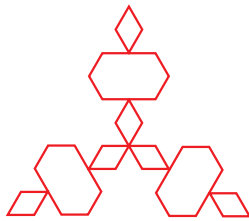
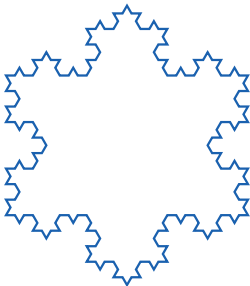
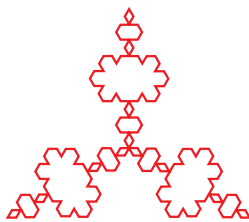
A fraktálok önhasonló objektumok, amelyeket önhasonlósági transzformációval hozhatunk létre. A természetben fraktálokat találunk a hegyek vízrajzában, tengerpartok alakján, de akár a fák lombozata, az állatok érzékszerve, tüdeje is fraktálszerkezetet mutat. Ha a diszkussziót olyan fraktálokra szűkítjük le, amelyek vektorgrafikusan kirajzolhatók, akkor ezek a képernyőn a legkényelmesebben a teknőc (*turtle*) grafika segítségével jeleníthetők meg, egy saját magát meghívó, úgynevezett rekurzív algoritmus segítségével. Mi erre a Python programozási nyelvet használtuk. A teknőcgrafika úgy fogható fel, mint egy ceruza, amit a papíron mozgathatunk (fel is emelhetjük, ha akarjuk) és vele két műveletet végezhetünk: az éppen aktuális irányba adott távolságot haladhat, majd e haladási irányhoz képest adott szöggel elforgatva egy új irányt jelölhetünk ki. Természetesen a teknőc aktuális x - y koordinátája is leolvasható, ezt a későbbiekben igénybe is vesszük.

A 2. ábrán két fraktált – az úgynevezett Koch hópolyhet és a Koch anti-hópolyhet – generáló Python-kódot mutatunk. A kódban aláhúzással jelöltük azt a két paramétert, amelyek állításával a fraktál átalakítható. A *sign* paraméter -1 értéke mellett kapjuk az anti-hópolyhet, míg $+1$ esetén a hópolyhet. A *level*

paraméter az ábrázolás mélységét-szintjét állítja be, az *N* paraméter pedig a fraktál méretét adja meg. A kódban láthatjuk, hogy ennek lényege az önmagát többször, rekurzívan meghívó *triangle* függvény. Egy fraktál elméletben végtelen „mélységig” lenne kirajzolandó, azonban – természetesen – csak néhány véges

2. ábra. Két fraktált – a Koch hópolyhet és a Koch anti-hópolyhet – generáló Python-kód.

```
import math
import turtle
sign = 1
level = 1
def triangle(turtle, size, depth):
    if depth == 0:
        turtle.forward(size)
    else:
        triangle(turtle, size/3, depth - 1)
        turtle.right(60*sign)
        triangle(turtle, size/3, depth - 1)
        turtle.left(120*sign)
        triangle(turtle, size/3, depth - 1)
        turtle.right(60*sign)
        triangle(turtle, size/3, depth - 1)
x_start = -0.5
y_start = -1 / (math.sqrt(3) * 2)
N = 600
turtle.speed(0)
turtle.hideturtle()
turtle.penup()
turtle.goto(x_start * N, y_start * N)
turtle.pendown()
for i in range(3):
    triangle(turtle, N, level)
    turtle.left(120)
turtle.done()
```

hópehely	anti-hópehely	szint
		1
		2
		3

3. ábra. A hópehely és anti-hópehely fraktál kirajzoltatva a 2. ábrán megadott Python-kód segítségével, a fraktálok különböző szintjén, amit a *level* paraméter állít.

lépésig tudjuk megjeleníteni, a 3. ábra ezen lépések mentén történő evolúciót is bemutatja a *level* paraméter változtatása mellett.

Fraktálok oszcilloszkópon

A fraktálok kirajzoltatását követően, a teknőcgrafikán elégséges azon pontok x - y koordinátáit kiolvastatni, ahol a teknőc elfordul. Ezen pontok kiolvasásával előáll az a koordinátasereg, amelyet az oszcilloszkóp x - y üzemmódja segítségével kirajzoltathatunk. A 4. ábra felső része ezeket a jeleket mutatja az idő függvényében, a Python-kódból generálva a teknőc grafika `turtle.pos()` parancsának segítségével.

Az így kapott x és y csatornaértékeket letölthetjük egy vektorfüggvény-generátornak (amennyiben ez rendelkezésre áll), ami ezt a jelet időben periodikusan kiadva az oszcilloszkóp képernyőjén fraktálunkat a megadott mélységig kirajzolja. Mi a kísérletekhez Siglent SDG1025 függvénygenerátort használtunk, azonban ez az eszköz kiváltható a számítógép audiókimenetének felhasználásával. Ehhez egy olyan kábelt

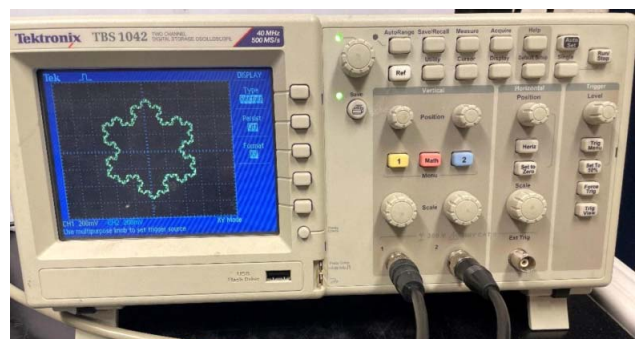
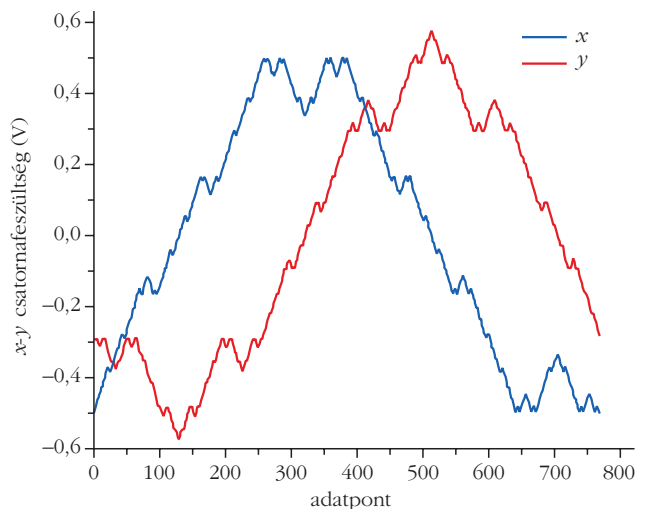
szükséges készítenünk, ami a hangszórókimenet 3,5 mm-es jack kimentéről a jobb és bal oldali hangszórók jelét külön-külön kivezetve az oszcilloszkóp BNC-bemeneteihez csatlakozik. Ezt házilag is elkészíthetjük, illetve ilyen csatlakozó kereskedelmi forgalomban is beszerezhető.

Ezt követően a jobb és bal oldali hangszórók kimeneteit felprogramozhatjuk Pythonban a 4. ábrán látható jelalakok segítségével, a *scipy.io* Python-csomagot használva. Majd az így kapott jeleket megjeleníthetjük az oszcilloszkópon, amit a 4. ábra alsó részén, fényképként mutatunk.

Az audió jelalakoknál felmerül, hogy mekkora a megjeleníthető legnagyobb frekvencia. Az időbeni jelalakokon látható, hogy gyakori váltásokat, ugrásokat tartalmaz. Az audiocsatorna programozásánál meg kell adni azt a legkisebb egységet, „bitrate”-et, amin a változó jeleket látjuk. A tipikus bitrate 44 100 Hz, azaz időben két egymást követő váltás távolsága 22,7 mikroszekundum. Ehhez a Nyquist-tétel alapján 22 kHz-es maximális kimeneti frekvencia tartozik, ami az emberi fül érzékenységi tartományához van közel, ami 20 kHz-ig terjed (elsősorban gyermekeknél). A jelek beprogramozásakor két lehetőségünk adódik:

- a fraktál egyes pontjait a 44 100 Hz-es bitrate-nek megfelelően programozzuk be, majd ezeket ismételjük.

4. ábra. Felül: a teknőcgrafika segítségével kirajzolt „level = 4” hópehelyfraktál x és y koordinátpontjaihoz tartozó jelek az idő függvényében. Alul: a fraktál fényképe az oszcilloszkópon.



Például a 3. ábrán látható 3-as szintű hópehely fraktál 193 darab pontot tartalmaz, így 4,37 milliszekundumot követően a jelet ismételtetjük, így az audiokimeneten lejátszva olyan hangot kapunk, amelynek alapfrekvenciája 228,5 Hz, ami az A₃ zenei hanghoz közeli.

– másik lehetőség, hogy a fraktál egyes pontjait „kitartjuk”, például 10 pontig, azaz 227 mikroszekundumig, így az alapharmonikus frekvenciáját szabadon változtathatjuk. Erre utal cikkünk címe, ha a jeleket az audiokimeneten hozzuk létre, akár ezek hangját is meghallgathatjuk!

Programmal történő számolás segítségével mozgó fraktálokat (de bármi más animációt) is készíthetünk, akár valós időben programozva az audiokimene- tet, akár előre generált hangfájlok segítségével. Az alábbi linkeken egy level 2 – <https://drive.google.com/file/>



d/1WRXTeOu1MYDsYwBSaqk5EEBCPLB04Li7/view?usp=sharing – és level 3 – https://drive.google.com/file/d/1u2QRW0tAEKAjCJTRE_gYCPVkuLiZwyob/view?usp=sharing – körbeforgó fraktált videóztunk le az oszcilloszkóp képernyőjéről (lásd a QR-kódokat is).

Összefoglalva elmondhatjuk, hogy a fraktálokat és konstrukciójukat informatikai eszközök segítségével is bemutathatjuk, a mi munkánk arra irányult, hogy mely módon lehet ezt kombinálni a hagyományos elektronikával, ami egyben bemutatja az oszcilloszkóp működését is. További érdekes és bonyolultabb animációk oszcilloszkópos létrehozásának csak a diák- kok képzelete szab határt.

Irodalom

1. Wikipedia <https://hu.wikipedia.org/wiki/Lissajous-görbe>
2. Doom on an Oscilloscope (Tektronix 2220), forrás: https://www.youtube.com/watch?v=OU16lIx_pC8