

UNIVERSITATEA "ALEXANDRU-IOAN CUZA" DIN IASI

FACULTATEA DE INFORMATICA



LUCRARE DE LICENTA

Procesare Audio pentru Invățare Automată

propusă de

Alexandru-Constantin Lazăr

Sesiunea: Iunie, 2023

Coordonator științific

Prof. Dr. Adrian Iftene

UNIVERSITATEA "ALEXANDRU-IOAN CUZA" DIN IASI

FACULTATEA DE INFORMATICA

**Procesare Audio pentru Învățare
Automată**

Alexandru-Constantin Lazăr

Sesiunea: Iunie, 2023

Scientific coordinator

Prof. Dr. Adrian Iftene

Avizat,

Îndrumător lucrare de licență,

Prof. Dr. Adrian Iftene.

Data:

Semnătura:

Declarație privind originalitatea conținutului lucrării de licență

Subsemnatul **Lazăr Alexandru-Constantin** domiciliat în **România, jud. Neamț, mun. Roman, Energiei, nr. 5**, născut la data de **01 ianuarie 2018**, identificat prin CNP **1234567891234**, absolvent al Facultății de informatică, **Facultatea de informatică** specializarea **informatică**, promoția 2018, declar pe propria răspundere cunoscând consecințele falsului în declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr. 1/2011 art. 143 al. 4 și 5 referitoare la plagiat, că lucrarea de licență cu titlul **Procesare Audio pentru Învățare Automată** elaborată sub îndrumarea domnului **Prof. Dr. Adrian Iftene**, pe care urmează să o susțin în fața comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență să fie verificată prin orice modalitate legală pentru confirmarea originalității, consumând inclusiv la introducerea conținutului ei într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări științifice în vederea facilitării falsificării de către cumpărător a calității de autor al unei lucrări de licență, de diplomă sau de disertație și în acest sens, declar pe proprie răspundere că lucrarea de față nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Data:

Semnătura:

Declarație de consimțământ

Prin prezenta declar că sunt de acord ca lucrarea de licență cu titlul **Procesare Audio pentru Învățare Automată**, codul sursă al programelor și celealte conținuturi (grafice, multimedia, date de test, etc.) care însătesc această lucrare să fie utilizate în cadrul Facultății de informatică.

De asemenea, sunt de acord ca Facultatea de informatică de la Universitatea "Alexandru-Ioan Cuza" din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Absolvent **Alexandru-Constantin Lazăr**

Data:

Semnătura:

Cuprins

| | |
|--|----------|
| Introducere | 2 |
| 1 Caracteristici Audio Elementare | 3 |
| 1.1 Sunete si forme de unda | 3 |
| 1.1.1 Frecventa | 5 |
| 1.1.2 Amplitudinea | 6 |
| 1.1.3 Timbru | 6 |
| 1.1.4 Faza | 7 |
| 1.2 Relatia dintre notele muzicale si frecventa undelor | 7 |
| 1.2.1 Inaltimea sunetelor | 7 |
| 1.2.2 Note MIDI | 8 |
| 1.2.3 Calculul frecventei in functie de nota MIDI | 9 |
| 1.3 Intensitate, volum și timbru | 9 |
| 1.3.1 Intensitate | 9 |
| 1.3.2 Volum | 9 |
| 1.3.3 Timbru | 10 |
| 1.3.4 <i>Sound Envelope</i> | 10 |
| 1.4 Sunete complexe | 12 |
| 1.5 Reprezentarea semnalelor audio in sisteme digitale | 12 |
| 1.5.1 Diferenele intre semnale analogice si semnale digitale | 12 |
| 1.5.2 <i>Sampling</i> | 13 |
| 1.5.3 Frecventa Nyquist | 13 |
| 1.5.4 Aliasing | 14 |
| 1.5.5 <i>Quantization</i> | 14 |
| 1.6 Separarea semnalelor audio in cadre | 15 |
| 1.6.1 marimea unui cadru | 15 |

| | | |
|---------------------|--|-----------|
| 1.6.2 | <i>Spectral leakage</i> | 16 |
| 1.7 | Transformarea Fourier | 18 |
| 1.7.1 | Simple Fourier Transform | 18 |
| 1.7.2 | Fourier Transform using complex numbers | 19 |
| 1.7.3 | Visualizing the Fourier Transform principles in Python | 23 |
| 2 | Titlul celui de-al doilea capitol | 26 |
| 2.1 | Titlul secțiunii 1 | 26 |
| 2.2 | Titlul secțiunii 2 | 27 |
| 2.3 | Titlul secțiunii 3 | 27 |
| 3 | Titlul celui de-al treilea capitol | 28 |
| 3.1 | Titlul secțiunii 1 | 28 |
| 3.2 | Titlul secțiunii 2 | 29 |
| Concluzii | | 30 |
| Bibliografie | | 31 |

Introducere

Elaborarea lucrării de licență a luat naștere dintr-o necesitate avută de mulți producători sau ingineri audio. În momentul analizării unei compozitii muzicale, ar fi foarte la îndemână dacă am ști mai multe informații precum, acordurile respectiv cheia în care a fost scrisă melodia. Mai mult, dacă am reuși să exportăm toate aceste informații într-un format convenient, am putea să recreem sau să modificăm compozitia. Astfel, plecând de la această idee, am setat obiectivele acestei lucrări de licență, și anume creearea unui program de tip plug-in care este compatibil cu o stație de lucru digitală, cu ajutorul căruia am analiza notele muzicale ale unor date de intrare audio, urmând să le exportăm sub un format *MIDI*.

Lucrarea se compune din două parti. Prima parte cuprinde o analiza științifica ce acopera caracteristicile semnalelor audio, respectiv stocarea și interpretarea datelor de intrare audio.

A doua parte cuprinde descrierea arhitecturii unui program demonstrativ.

Capitolul 1

Caracteristici Audio Elementare

In cadrul acestui capitol voi expica mai multe notiuni audio elementare care vor fi folosite de-a lungul intregii lucrari de licenta. Multe dintre aceste caracteristici audio vor fi puncte de plecare pentru recunoasterea sunetelor in cadrul modelelor de intelligența artificiala sau invatare automata.

1.1 Sunete si forme de unda

Un semnal audio poate fi reprezentat folosind forme de undă, care, prin natura sa, se aseamana cu o diagrama de presiune. O astfel de reprezentare este folosita pentru a reprezenta o compozitie muzicala, dar are capacitatea de a reprezenta multe alte informatii, din aproape orice domeniu.

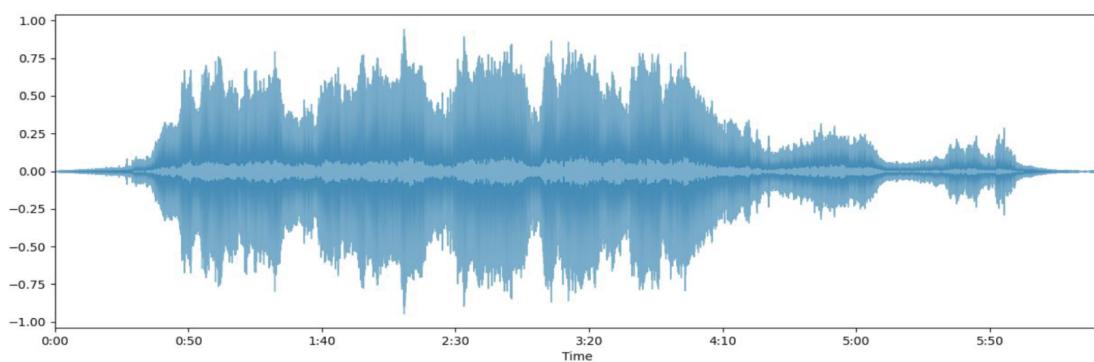


Figura 1.1: Forma de unda a unei compozitii de pian

Formele de unda, in engleza *Waveforms*, pot fi clasificate, drept Periodice si Aperiodice. Cele periodice pot fi descrise prin functii sinus. Undele simple sunt formate

dintron un singur semnal sinusoidal, pe cand cele complexe fiind formate din mai multe functii sinusoidale inmultite. Acestea sunt specifice instrumentelor si vocii umane. Frecventele acestor unde indica notele muzicale.

Formele de unda aperiodice sunt, de asemenea, de doua tipuri: continue si transiente. Cele continue sunt compuse din mai multe unde aleatorii, pe cand cele transiente reprezinta un puls. Undele transiente sunt specifice instrumentelor de percutie.

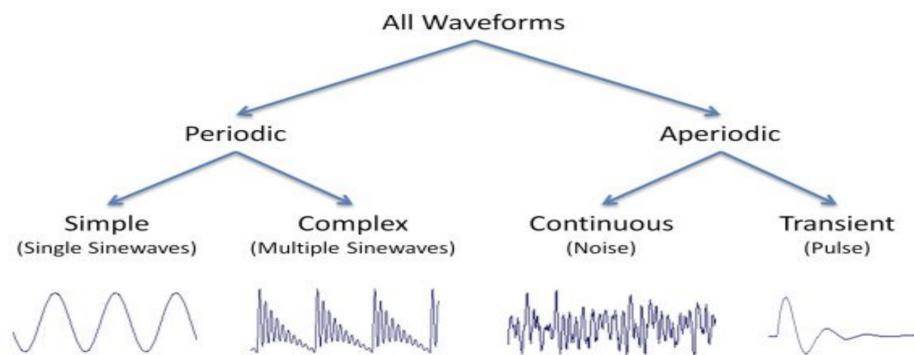


Figura 1.2: Tipuri de forme de unda

O forma de unda este caracterizata de:

- Frecventa
- Amplitudine
- Timbru
- Faza

1.1.1 Frecventa

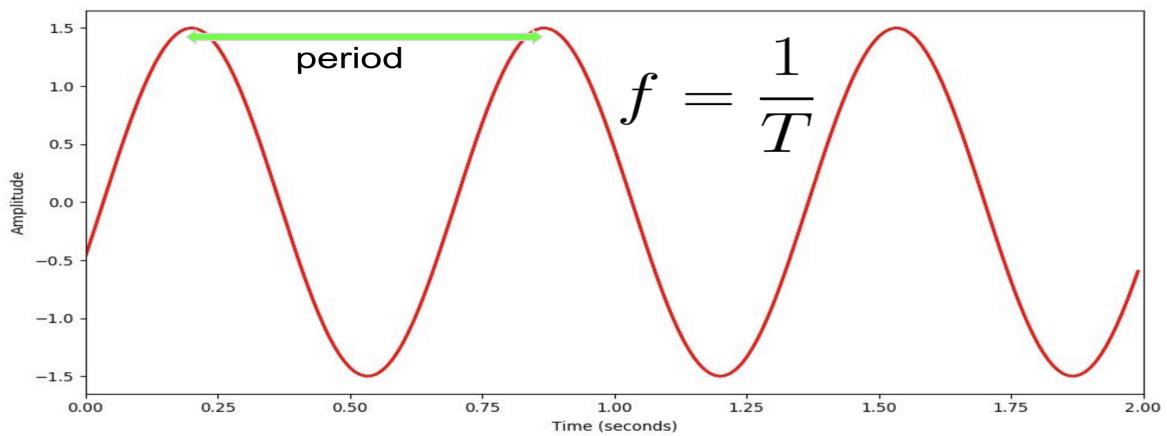
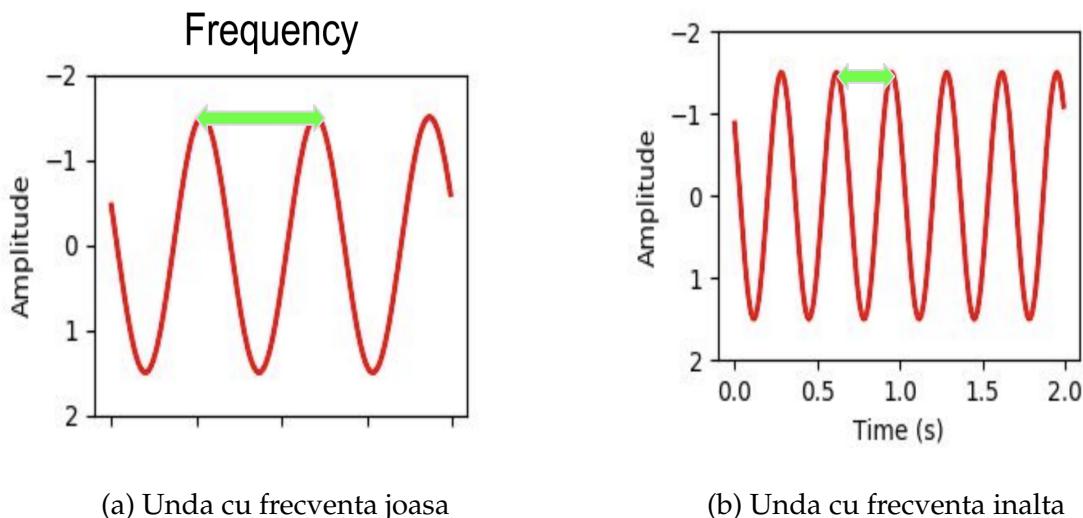


Figura 1.3: Perioada unei unde simple

Frecventa, sau inaltimea unui sunet, indica atat nota muzicala cat si octava specifica unui sunet. O frecventa mai inalta este specifica unui sunet ascutit, adica dintr-o octava mai mare, pe cand o frecventa joasa este specifica unui sunet "gros", adica dintr-o octava mai mica.



(a) Unda cu frecventa joasa

(b) Unda cu frecventa inalta

Figura 1.4: Reprezentarea grafica a doua unde cu frecvente diferite

1.1.2 Amplitudinea

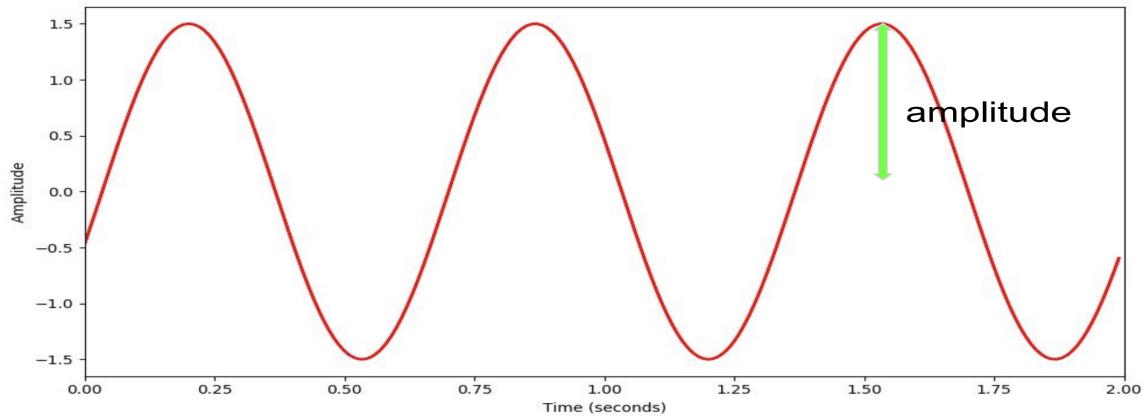
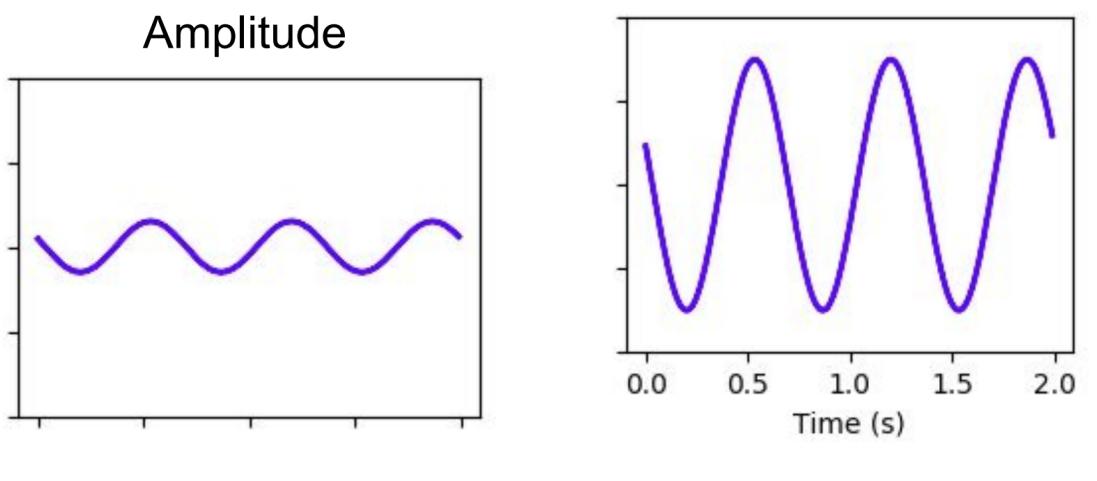


Figura 1.5: Amplitudinea unei unde simple

Amplitudinea unei unde indica intensitatea sunetului. Intensitatea poate fi măsurată în dB .



(a) Unda cu amplitudine mica

(b) Unda cu amplitudine mare

Figura 1.6: Reprezentarea grafică a două unde cu amplitudini diferite

1.1.3 Timbru

Timbrul unei forme de undă este un concept complex. Desi nu există o definitie completă a timbrului unei unde, aceasta este rezultatul mai multor caracteristici a unei unde, printre care frecvențele ce creează armonii în interiorul sunetului.

1.1.4 Faza

Faza unei unde indica un decalaj al inceperii semnalului. Poate fi calculat drept diferența dintre valoarea semnalului la momentul initial și valoarea 0 pe axa Ox .

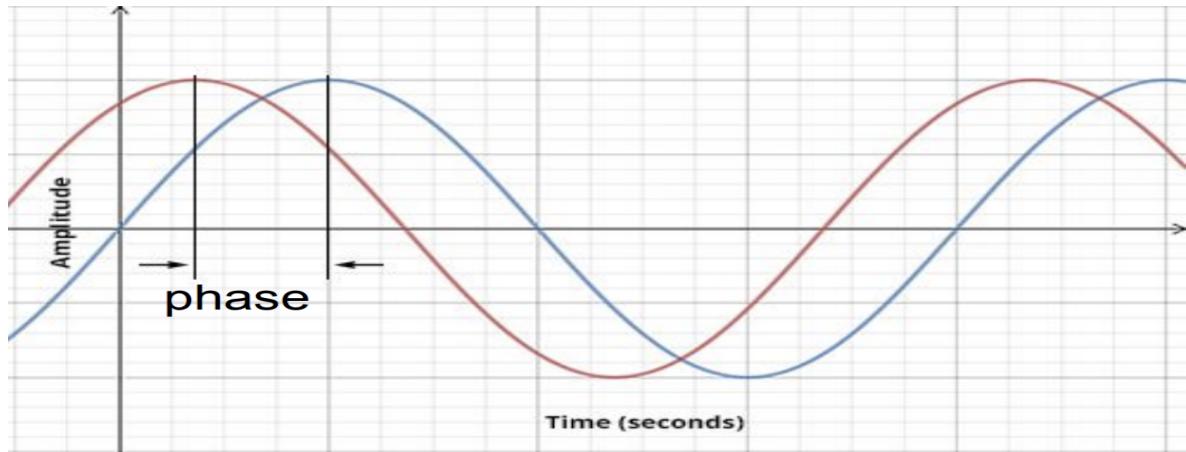


Figura 1.7: Reprezentarea grafica a doua unde cu faze diferite

1.2 Relatia dintre notele muzicale si frecventa undelor

1.2.1 Inaltimea sunetelor

Inaltimea sunetelor este calculata in Hz . Sunetele sunt percepute de urechea umana intr-un mod logaritmic, doua sunete fiind similare daca difera prin o putere a lui 2. De exemplu, nota La din octava a 4-a a unui pian, scrisa drept A4 conform notatiei muzicale pe litere, are frecventa $440\ Hz$. Nota A5, adica nota la din a 5-a octava, are frecventa de $880\ Hz$. Aceleasi note doar ca din octave diferite sunt percepute drept similare de urechea umana. Acestea difera perfect prin puteri a lui 2, astfel fiind percepute intr-un mod logaritmic.

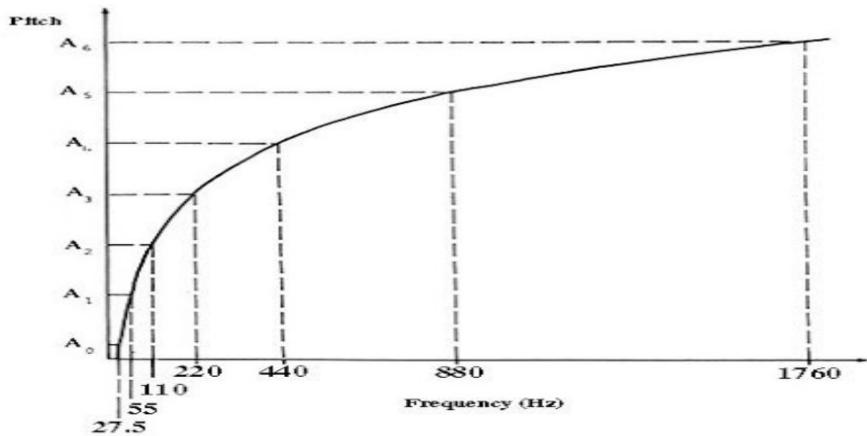


Figura 1.8: Maparea logaritmica a frecventei unei unde cu notele corespunzatoare

1.2.2 Note MIDI

MIDI, sau *Musical Instrument Digital Notation*, este un standard tehnic pentru a descrie protocoale de comunicare, interfata digitala si conectorii electrici care conectaza o varietate de instrumente muzicale.

Un instrument MIDI ingregistreaza orice interacțiune a unei persoane cu un buton, clapa sau petala si transforma aceste informatii intr-o reprezentare digitala care specifica intensitatea, inaltimea si momentul in care a fost cantat un sunet.

Vom folosi aceast standard tehnic pentru a memora digital toate informatiile despre opere muzicale, prelucrate de modelul de intelectual artificial si invatare automata.

| Note name | A0# | B0 | C1# | D1# | E1 | F1# | G1# | A1# | G2# | F2# | E2 | D2# | C2# | B2 | A2# | G3# | F3# | D3# | C3# | B3 | A3# | G3# | F3# | E4 | D4 | C4# | B4 | A4# | G4# | F4# | E5 | D5# | C5# | B5 | A5# | G5# | F5# | E6 | D6# | C6# | B6 | A6# | G6# | F6# | E7 | D7# | C7# | B7 | A7# | G7# | F7# | E8 | | | | | | | | | | | | | | | | | | | | | | | | |
|-------------|-----|----|-----|-----|----|-----|-----|-----|-----|-----|----|-----|-----|----|-----|-----|-----|-----|-----|----|-----|-----|-----|----|----|-----|----|-----|-----|-----|----|-----|-----|----|-----|-----|-----|----|-----|-----|----|-----|-----|-----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Midi number | 21 | 20 | 23 | 29 | 31 | 30 | 32 | 34 | 44 | 42 | 41 | 40 | 38 | 37 | 46 | 56 | 54 | 51 | 49 | 60 | 62 | 64 | 66 | 68 | 70 | 72 | 74 | 76 | 77 | 79 | 81 | 82 | 80 | 78 | 85 | 84 | 83 | 81 | 80 | 78 | 91 | 92 | 93 | 94 | 95 | 96 | 98 | 100 | 101 | 103 | 104 | 102 | 106 | 105 | 107 | 108 | 109 | 110 | 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 | 128 |
| Note name | A0# | B0 | C1# | D1# | E1 | F1# | G1# | A1# | G2# | F2# | E2 | D2# | C2# | B2 | A2# | G3# | F3# | D3# | C3# | B3 | A3# | G3# | F3# | E4 | D4 | C4# | B4 | A4# | G4# | F4# | E5 | D5# | C5# | B5 | A5# | G5# | F5# | E6 | D6# | C6# | B6 | A6# | G6# | F6# | E7 | D7# | C7# | B7 | A7# | G7# | F7# | E8 | | | | | | | | | | | | | | | | | | | | | | | | |

440 Hz 880 Hz

Figura 1.9: Note Midi impreuna cu frecventele asociate

1.2.3 Calculul frecventei in functie de nota MIDI

Fiecare nota MIDI are asociat un numar, dupa cum se poate observa si in Figura 1.9. De exemplu, nota A4 are asociata numarul 69. Putem folosi urmatoarea formula pentru a determina frecventa unei note MIDI:

$$F(p) = 2^{\frac{p-69}{12}} * 440 \text{ , unde } p \text{ este numarul asociat unei note MIDI}$$

$$F(69) = 2^{\frac{69-69}{12}} * 440 = 2^{\frac{0}{12}} * 440 = 440 Hz$$

1.3 Intensitate, volum și timbru

1.3.1 Intensitate

Intensitatea unui sunet este masurata in $\frac{W}{m^2}$. Urechea umana poate percepe sunete de o intensitate foarte mica. Cea mai mica intensitate pe care o putem auzi este notata cu *TOH*, abreviere pentru *Threshold of Hearing*.

$$TOH = 10^{-12} * \frac{W}{m^2}$$

Aceasta valoare este folosita drept un punct de referinta pentru calcului intensitatii unui sunet in decibeli.

$$dB(I) = 10 * \log_{10}\left(\frac{I}{I_{TOH}}\right)$$

$$dB(I_{TOH}) = 10 * \log_{10}\left(\frac{I_{TOH}}{I_{TOH}}\right) = 10 * \log_{10}(1) = 0$$

Dupa cum se poate observa si din formula, Intensitatea este o unitate de masura cu o scara logaritmica. Odata la circa $3dB$, intensitatea se dubleaza.

1.3.2 Volum

Volumul este interpretarea subiectiva a intensitatii. Este masurat in foni. Doua sunete cu aceeasi intensitate pot fi percepute de urechea umana drept avand volume diferite, daca sunetele difera in frecventa sau durata.

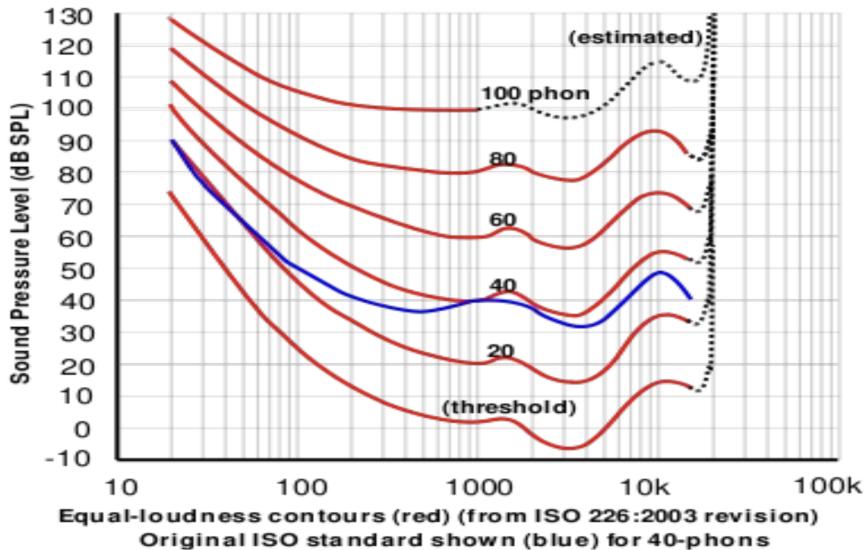


Figura 1.10: Ilustrarea relatiei dintre decibeli si foni pentru urechea umana

1.3.3 Timbru

Chiar daca nu exista o definitie exacta a timbrului, il putem numi drept diferența intre 2 note cu aceeasi intensitate, frecventa si durata. In practica, timbrul este diferența intre nota A4 cantata la pian si aceeasi nota cantata la saxofon.

Timbrul este o caracteristica multidimensională, acesta fiind rezultatul la mai multe atributii ale sunetului. O parte din aceste atributii sunt: *Sound Envelope*, continutul armonic, modulatia amplitudinii / frecventei.

1.3.4 Sound Envelope

Sound envelope-ul este una dintre cele mai importante caracteristici ale unui sunet periodic. Aceasta poate defini tipul unui sunet: clape, instrument cu coarde și arcuș, instrument cu coarde ciupite etcetera. Acesta este format din 4 variabile:

- *Attack* (atac): descrie durata necesara ca un sunet sa ajunga la amplitudinea maxima.
- *Decay* (decadere): descrie durata in care amplitudinea sunetului scade pana intr-un punct constant.
- *Sustain* (sustinere): descrie durata in care sunetul isi pastreaza o amplitudine constanta

- *Release* (eliberare): descrie durata in care amplitudinea sunetului scade pana la 0dB.

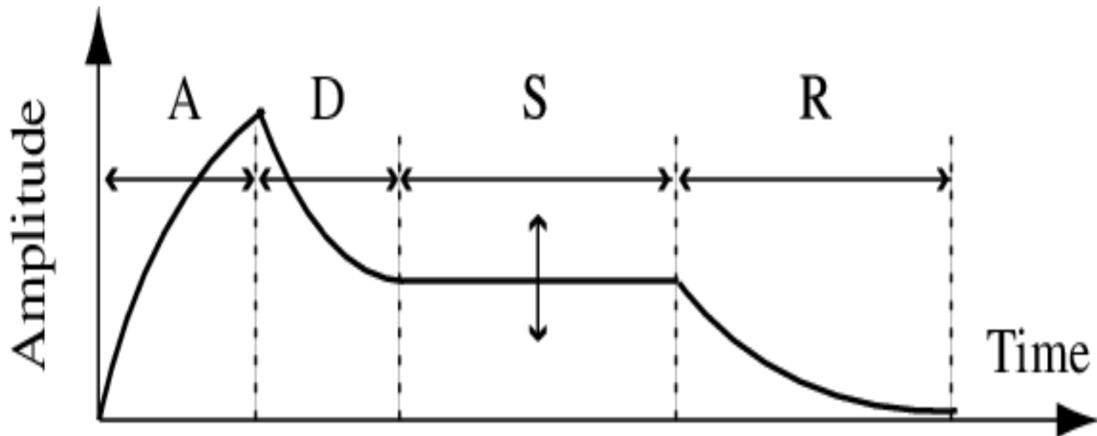


Figura 1.11: *Sound Envelope* bazat pe modelul *Attack-Decay-Sustain-Release*

Sound Envelope-ul ne poate oferi foarte multe informatii, care ne pot ajuta sa diferentiem instrumente intre ele, in cazul in care acestea se suprapun. De exemplu un instrument cu clape va avea mereu un atac, decadere si eliberare scurta, cu o sustinere lunga si liniara. Pe de alta parte, un instrument cu coarde si arcus care canta o nota folosind tehnica *tremollo*, adica o variatie periodica in amplitudine, va avea un atac lung, o decadere foarte mica, o sustinere ce sufera modulatii, si o decadere ce poate varia in functie de cum este cantata nota de catre instrumentalist.

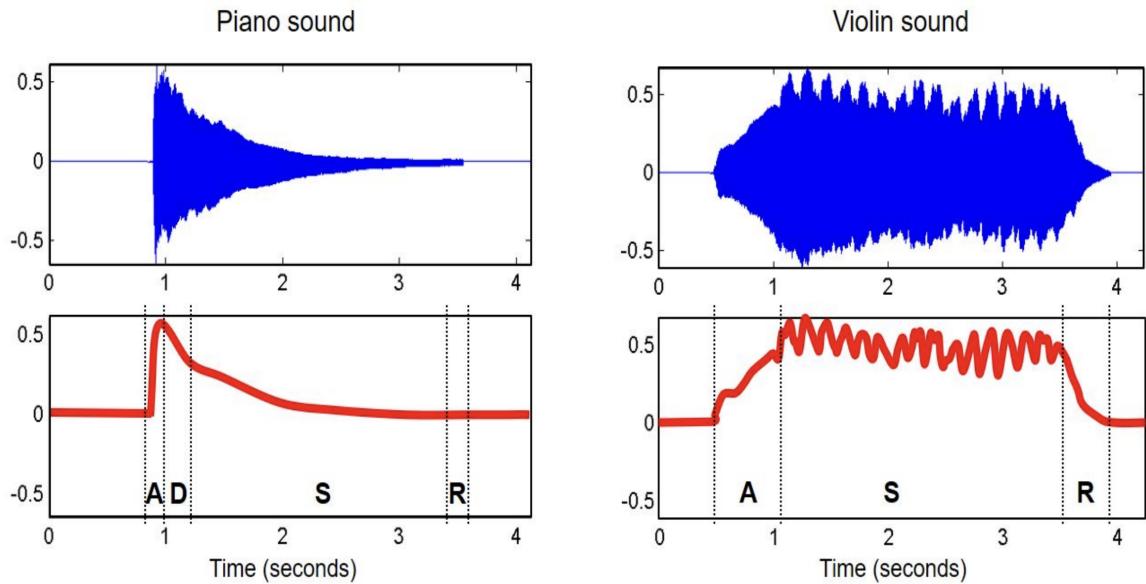


Figura 1.12: Ilustrarea diferenței între *Sound Envelope*-ul corespunzător unui pian și cel corespunzător unei vioare

1.4 Sunete complexe

Sunetele complexe sunt o superpozitie de sunete simple (sinusoide). Un sinusoid care intra în componenta unui sunet complex se numește **partial** și acest sinusoid descrie sunetul. Partialul cu frecvența cea mai joasă se numește **frecvența fundamentală** a sunetului. Un partial armonic este o frecvență care este un multiplu al frecvenței fundamentale. O dezarmonie între 2 sinusoide indică o deviere de la partialul armonic.

1.5 Reprezentarea semnalelor audio în sisteme digitale

1.5.1 Diferențele între semnalele analogice și semnalele digitale

Pentru a înțelege cum sunt stocate semnalele audio într-un sistem digital trebuie să înțelegem diferența între semnalele audio analogice și cele digitale. Semnalele analogice au valori continue pentru timp și amplitudine. Deoarece semnalele digitale nu pot avea valori infinite pentru timp sau amplitudine, va trebui să avem o secvență discretă de valori pentru acestea. Stocarea semnalelor audio analogice se face prin 2 metode: *Sampling* și *Quantization*.

1.5.2 Sampling

Procesul de Sampling implica stocarea datelor despre un semnal prin citirea, o data la o perioada T , a amplitudinii semnalului analogic. Valoarea obtinuta prin citirea mesajului se numeste *Sample point*.

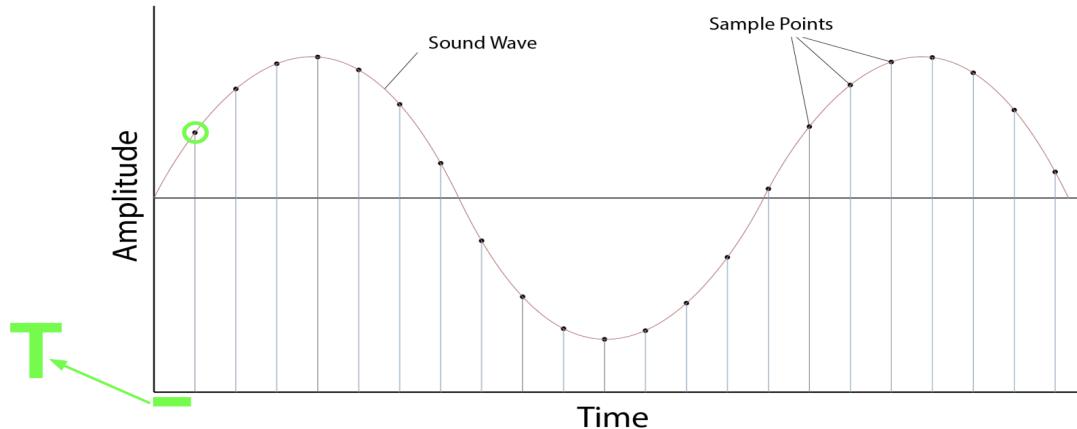


Figura 1.13: Punctele returnate de procesul de *sampling*

Sampling rate reprezinta frecventa citirilor de *Sample Points* intr-o secunda. Este calculata in Hz dupa urmatoarea formula:

$$S_r = \frac{1}{T}$$

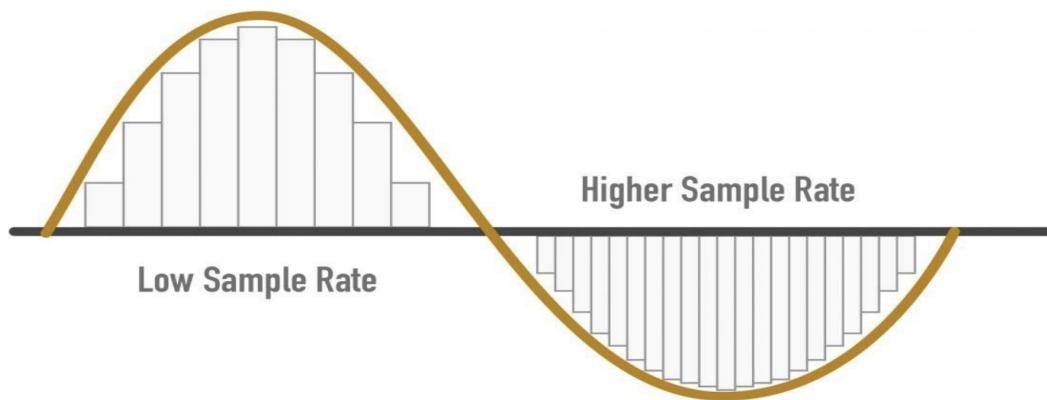


Figura 1.14: Diferenta intre un *Sample Rate* mic si unul mare

1.5.3 Frecventa Nyquist

Frecventa Nyquist se refera la cea mai intalta frecventa care poate fi reprodusa fara erori pentru un anumit *Sample Rate*. Aceasta se calculeaza dupa urmatoarea formula:

$$f_N = \frac{S_r}{2}$$

Ca atare, pentru a alege un sampling rate al semnalelor folosite, trebuie sa avem in vedere care va fi cea mai intalita frecventa pe care o putem reproduce fara erori. Urechea umana este sensibila la vibratii ale aerului cu frecvente intre $20Hz$ si $20000Hz$. Ca atare, pentru a reproduce sunete cu frecventa in intervalul de mai sus, vom avea nevoie de o frecventa Nyquist mai mare decat $20000Hz$, adica un *Sampling Rate* mai mare decat $40000Hz$.

Asadar, vom folosi un *Sampling Rate* cu valoarea de $44100Hz$, frecventa standard a CD-urilor, intrucat aceasta valoare pentru *sampling rate* are o frecventa Nyquist mai mare decat $20Hz$.

$$f_N = \frac{S_r}{2} = \frac{44100}{2} = 22000Hz > 20000Hz$$

1.5.4 Aliasing

O problema a metodei de *sampling* este *Aliasing*-ul. Acesta este un fenomen care se intampla atunci cand aplicam metoda de *sampling* asupra unui sunet cu frecventa mai mare decat frecventa Nyquist. In figura de mai jos este exemplificat cum, folosind aceasta metoda asupra unei frecvente mai mari decat f_N , putem obtine un sunet cu o fază identica, dar cu o frecvență de 5 ori mai mică.

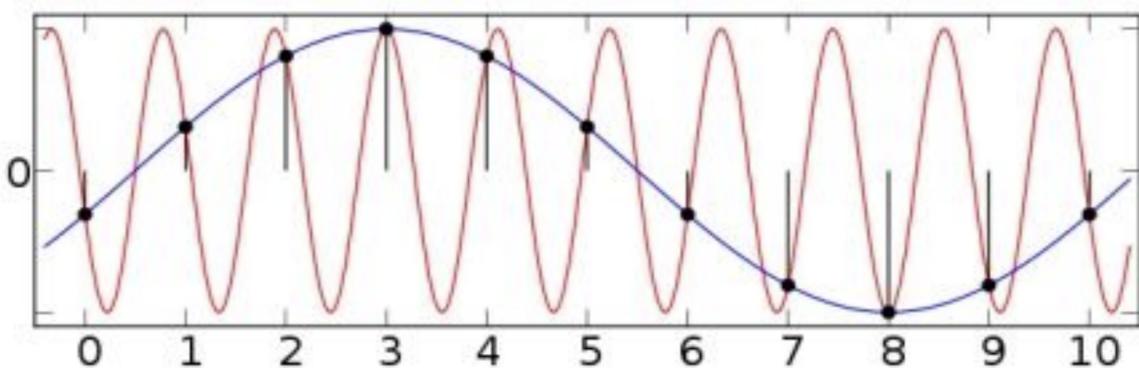


Figura 1.15: Fenomenul de *Aliasing* exemplificat

1.5.5 Quantization

Metoda *Quantization* este folosita pentru a aproxima valoarea amplitudinii la momentul t unui semnal analog continuu la cea mai apropiata valoare ce poate fi stocata

digital. *Bit depth* indica rezolutia pe care o poate avea amplitudinea unui semnal. De exemplu, pentru un *Bit depth* cu valoarea 4, vom avea o rezolutie de 16 valori.

$$Res = 2^{B_d} = 2^4 = 16, \text{ unde } Res \text{ este notatia pentru rezolutie si } B_d \text{ este notatia pentru Bit depth.}$$

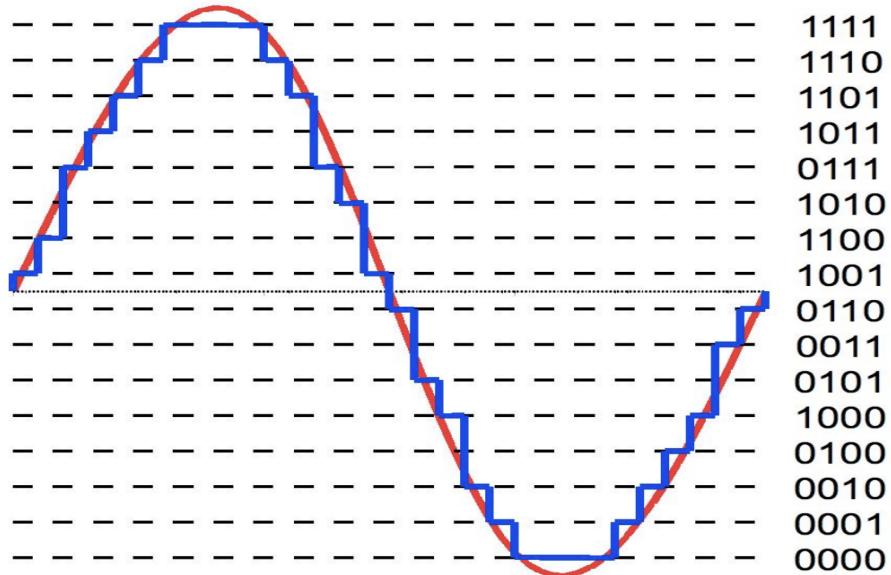


Figura 1.16: Procesul de *Quantization* cu un *Bit depth* de 4 biti.

1.6 Separarea semnalelor audio in cadre

1.6.1 marimea unui cadru

Un *sample* ofera foarte putine informatii referitoare la un semnal audio. Durata unui *sample* dintr-un semnal ce are un *Sample Rate* de 44100Hz este de $\frac{1}{44100} = 0.0227\text{ms}$. Astfel, pentru a analiza un semnal pentru a-i obtine proprietatile, este mai de folos sa il impartim in mai multe cadre. Un cadru este o multime de cadre. De obicei marimea unui cadru este o putere a lui 2. Cele mai des folosite valori pentru marimea unui cadru sunt puteri ale lui 2 din intervalul [256, 8192].

Deoarece urechea umana poate percepe semnale audio cu durata minima de circa 10ms , vom alege valoarea minima de 512 *sample*-uri pentru marimea cadrelor.

$$d_f = \frac{1}{S_r} * K = \frac{1}{44100} * 512 = 11.6\text{ms} > 10\text{ms}$$

unde d_f este durata unui cadru, S_r este *Sample rate*-ul si K este marimea unui cadru.

1.6.2 Spectral leakage

In momentul in care divizam un semnal audio in cadre, vom avea capetele cadrului discontinue. Discontinuitatile apar drept sunete cu frecvențe înalte care nu sunt prezente in semnalul original.

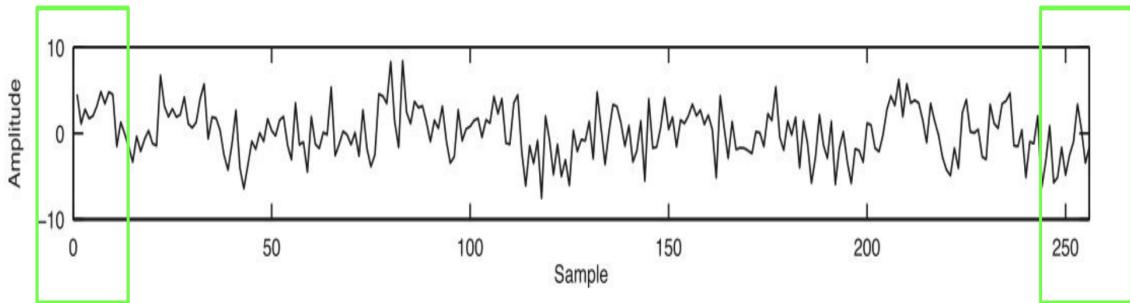


Figura 1.17: Cadru cu capete discontinue

Putem rezolva aceasta problema prin aplicarea unei functii de *windowing*. Una dintre cele mai folosite functii de acest tip este *Hann's windowing function*:

$$w(k) = 0.5 * (1 - \cos(\frac{2\pi k}{K-1})), k = 1 \dots K,$$

unde K este marimea unui cadru

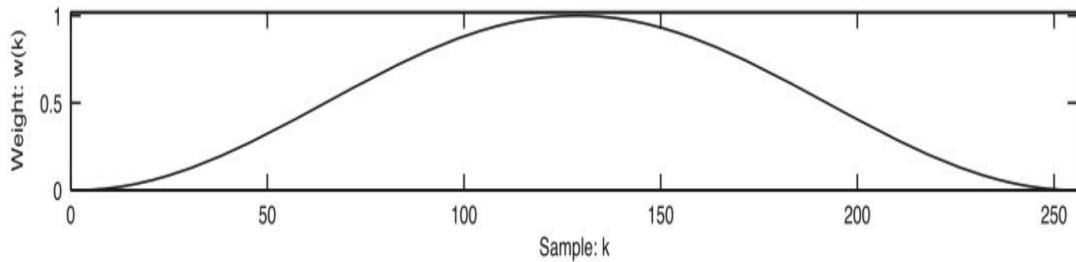


Figura 1.18: Graficul functiei de *windowing* Hann

Astfel, aplicand functia aceasta functie, vom obtine un cadru cu capetele cu amplitudini nule si cu un mijloc cu amplitudini identice cu cadrul original. Pentru a aplica functia de *windowing* vom aplica urmatoarea functie:

$$s_w(k) = s(k) * w(k), k = 1 \dots K,$$

unde K este marimea unui cadru

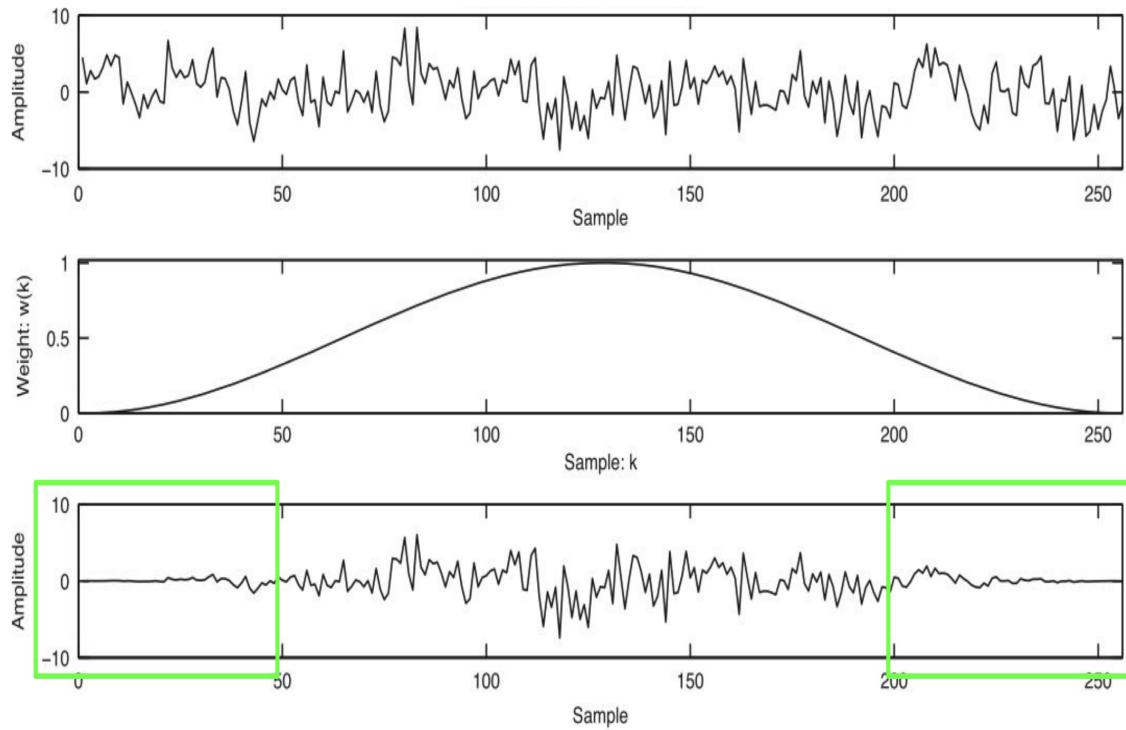


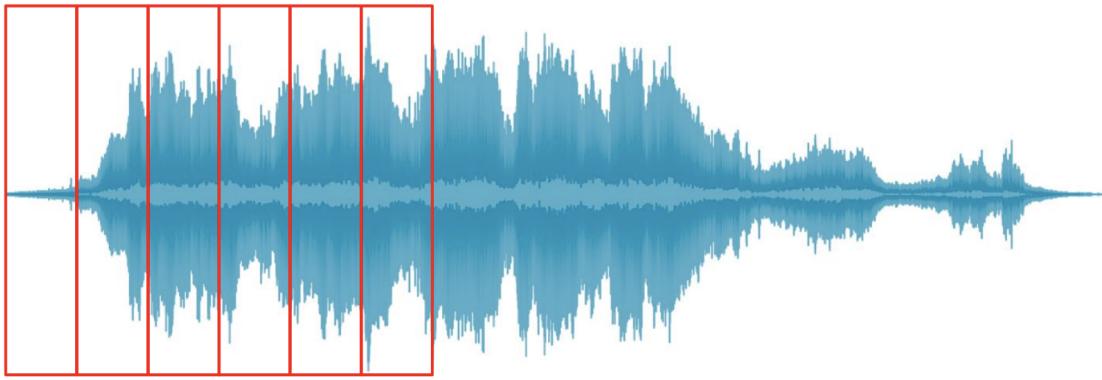
Figura 1.19: Cadru cu capete discontinue, obtinut dupa ce am aplicat functia Hann de *windowing*.

Totusi, functia de windowing "sterge" informatiile de la inceputul unui cadru. Astfel, daca impartim semnalul in cadre si analizam cadrele consecutiv vom pierde informatii.

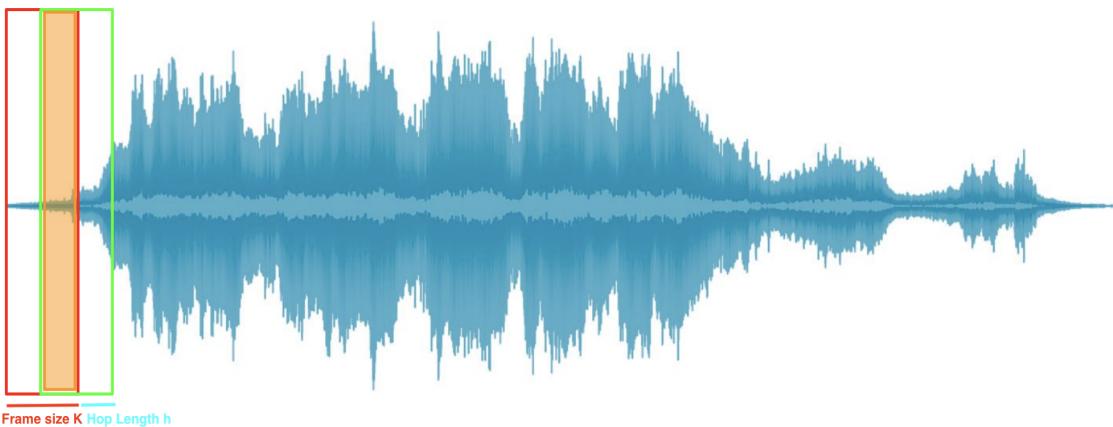


Figura 1.20: Cadre analizate consecutiv, dupa aplicarea functiei Hann de *windowing*.

Pentru a rezolva aceasta problema, vom analiza cadrele prin suprapunerea lor. Numarul de *sample*-uri pe care le sarim pentru a trece la urmatorul cadru se numeste *hop length*. De obicei *hop length*-ul este egal cu jumata din marimea unui cadru.



(a) Cadre analizate consecutiv



(b) Cadre analizate prin suprapunere

1.7 Transformarea Fourier

1.7.1 Simple Fourier Transform

Transformarea Fourier descompune un sunet in multe sinusoidale ce se afla in componenta sa. Aceasta transformare functioneaza prin compararea mai multor sinusoidale de frecvente diferite cu sunetul original. Pentru fiecare sinusoidal obtinem o magnitudine si o faza. O magnitudine mare indica similaritate intre sinusoidal si sunetul original. Pasii Transformarii Fourier sunt:

1. Alegerea unei frecvente
2. Optmizarea fazei sinusoidului corespunzator frecventei
3. Calculul magnitudinii
4. Repetarea acestor pasi pana am parcurs tot spectrul de frecvente.

Using the following equation, we can calculate the value of the amplitude of a sinusoidal at any given time t , for a function with the frequency f and the phase φ :

$$H(t) = \sin(2\pi * (f * t - \varphi))$$

We can calculate the best phase that a sinusoidal can have, in order to maximise the magnitude, using the following formula:

$$\varphi_f = \operatorname{argmax}_{\varphi \in [0,1)} (\int s(t) * H(t) * dt),$$

unde $s(t)$ este valoarea amplitudinii semnalului la momentul t , iar $f t$ valoarea sinusoidului corespunzator frecventei f .

Prin inmultirea sinusoidului cu valoarea semnalului, vom obtine o noua functie sinusoidala, care va avea aria, adica valoarea integralei, maxima daca atat functia corespunzatoare semnalului si a sinusoidului sunt identice. Astfel, parcurgand toate fazele semnalului posibile, vom afla valoarea lui φ pentru care sinusoidul este cel mai similar cu semnalul.

Formula pentru calculul magnitudinei maxime a unui sinusoid, in relatie cu un semnal :

$$d_f = \max_{\varphi \in [0, 1)} (\int s(t) * H(t) * dt)$$

unde $s(t)$ este valoarea amplitudinii semnalului la momentul t , iar $f t$ valoarea sinusoidului corespunzator frecventei f .

1.7.2 Fourier Transform using complex numbers

Every time we apply a Fourier transform we end up with 2 variables, phase and magnitude, for each pure sinusoidal. We would like to encode both of these variables into a single complex number, in order to more easily interpret it in a graphical plane, where we plot the real part of the complex number on the Ox axis, and the imaginary part on the Oy axis. But, before we can get into that, we first need to get familiar with complex numbers.

Complex numbers

First of all, a complex number is a number that is made up of a real and imaginary part.

The formula for a complex number is the following:

$$c = a + b * i \text{ , where } a, b \in \mathbb{R} \text{ and } i \text{ is a imaginary unit with the value } i = \sqrt{-1}$$

Plotting complex numbers

We can plot a complex number in a plane with the Ox axis corresponding to the Real part of the number, and the Oy axis corresponding to the imaginary one, as can be seen in the following image:

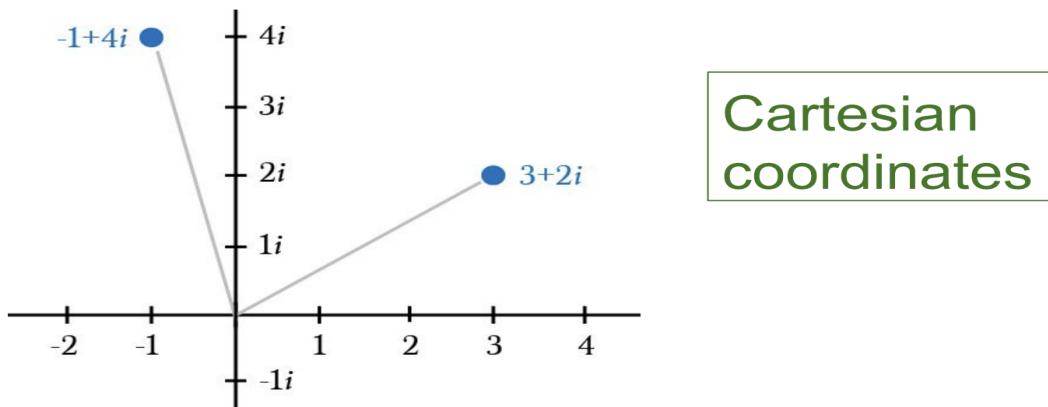


Figura 1.22: Plotting complex numbers

Polar coordinate representation

Another way to represent complex numbers is by using polar coordinates, which introduces two new parameters, the distance of the complex number from the origin and the angle γ , as can be seen in the figure below:

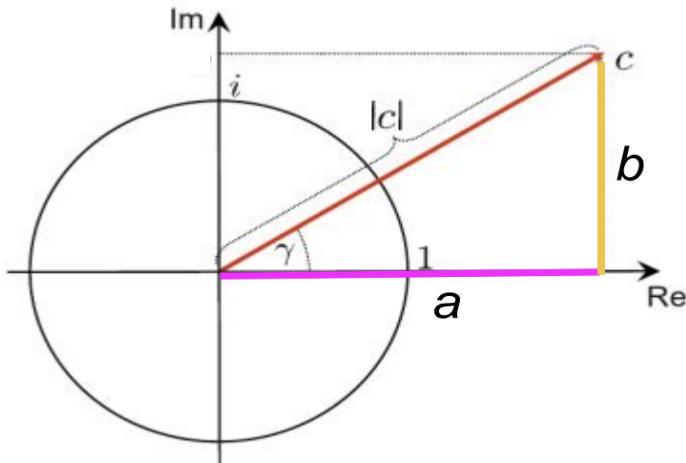


Figura 1.23: Representation of complex numbers using polar coordinates

We have the following formulas:

$$\gamma = \arctan\left(\frac{b}{a}\right)$$

$$\cos(\gamma) = \frac{a}{|c|} \Rightarrow a = |c| * \cos(\gamma) \quad (1)$$

$$\sin(\gamma) = \frac{b}{|c|} \Rightarrow b = |c| * \sin(\gamma) \quad (2)$$

$$c = a + b * i \quad (3)$$

$$(1), (2), (3) \Rightarrow c = |c| * \cos(\gamma) + i(|c| * \sin(\gamma)) \Rightarrow$$

$$\Rightarrow c = |c| * (\cos(\gamma) + i * \sin(\gamma)) \quad (4)$$

Euler's formula: $e^{i\gamma} = \cos(\gamma) + i * \sin(\gamma) \quad (5)$

$$(4), (5) \Rightarrow c = |c| * e^{i\gamma}$$

We now have the representation of a complex number as 2 components, $|c|$, which denotes the distance from the origin, and $e^{i\gamma}$, which denotes the direction of the number in the complex plane.

Continuous audio signal

We can define a continuous audio signal as the following function:

$$g : \mathbb{R} \Rightarrow \mathbb{R}$$

$$g(t) = \text{amplitude level at the moment } t$$

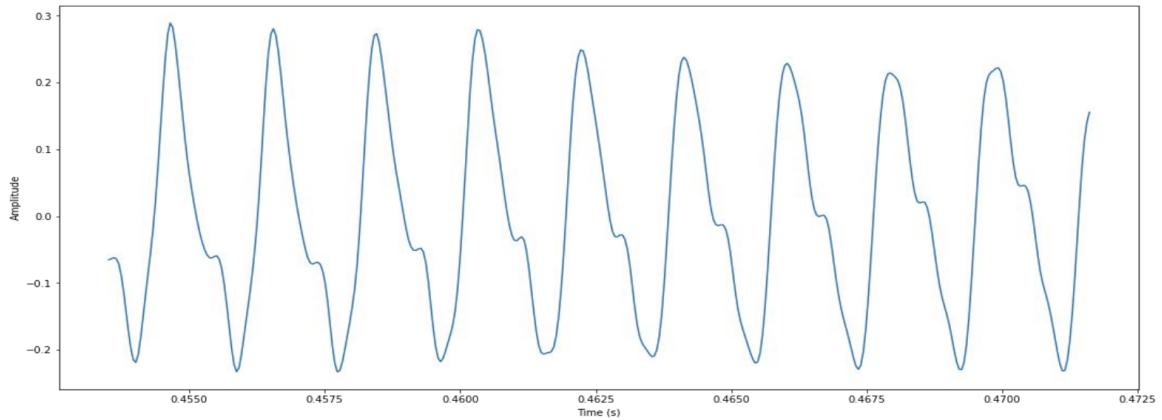


Figura 1.24: Continuous audio signal

Complex Fourier Transform Coefficients

The complex number representation for a sinusoidal signal of frequency f is:

$c_f = \frac{d_f}{\sqrt{2}} * e^{-i2\pi\varphi f}$ where the amplitude is $\frac{d_f}{\sqrt{2}}$, $\sqrt{2}$ being a normalization standard and $e^{-i2\pi\varphi f}$ being the direction of the complex number. We can notice from ⑤ that $\gamma = -2\pi\varphi f$, which is the formula for a sinusoidal function of frequency f .

Complex Fourier Transform

For each sinusoidal signal corresponding to the frequency f that we compare against the signal, the Fourier Transform will return a complex number that encodes the magnitude and phase. We can define the function for this as following:

$$\hat{g} : \mathbb{R} \Rightarrow \mathbb{C}$$

$$\hat{g}(f) = c_f = \int g(t) * e^{-i2\pi ft} dt$$

If we decompose this using **Euler's formula** we get:

$$\hat{g}(f) = \int g(t) * \cos(-2\pi ft) dt + i \int g(t) \sin(-2\pi ft) dt$$

Magnitude and phase are related to each other. $\int g(t) * \cos(-2\pi ft)$ calculates the magnitude between the 2 signals by multiplying the initial signal with the sinusoidal function of frequency f and calculating the integral of the new obtained function. If the 2 signals are opposing, the integral will be 0. If the 2 signals are similar, the integral will have a high magnitude. $\int g(t) * \cos(-2\pi ft)$ calculates the phase. If the 2 signals are similar, meaning that the sinusoidal function of frequency f is apart of the original signal, this will return the value for phase that we need to subtract from $-2\pi ft$ in order to maximize the magnitude.

1.7.3 Visualizing the Fourier Transform principles in Python

Python provides us with many tools to easily visualise and represent complex mathematical functions and operations. We will use this language to visually explain the how the Fourier Transform calculates the magnitude and phase in the formula for \hat{g} .

In the following lines of code we will observe the relation of sounds which are related, meaning that one of the sounds is part of the other. In order to do so in python3, we will first need to import the following libraries:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
```

The *numpy* library is used for performing logical and mathematical calculations on arrays and matrices and *pyplot* is used to graphically plot mathematical functions.

```
1 def create_sinusoidal_function(frequency, time, phase):
2     angle = -2 * np.pi * (frequency * time + phase)
3     return np.sin(angle)
```

As mentioned in previous chapters, the function for a pure tone is $\sin(2\pi * (f * t + \varphi))$. The function *create_sinusoidal_function* returns the value for such a signal over a period of time t .

```
1 time = np.linspace(0, 2, 10000)
```

This line of code creates an array that takes up values from 0 to 2, containing 10000 samples.

```
1
2 signal1 = create_sinusoidal_function(frequency=1, time = time, phase = 0 )
    * 2
3 signal2 = create_sinusoidal_function(frequency=2, time = time, phase = 0)
4 signal3 = signal1 * signal2
```

Here we created 3 signals. *signal1* and *signal2* are 2 simple sinusoidal functions of frequencies of 1Hz , respectively of 2Hz . The *signal1* has its amplitude multiplied by 2, in order to more easily differentiate the 2 signals between each other. *signal3* is the multiplication of the 2 frequencies.

```

1 plt.figure(figsize=(18, 8))
2 plt.plot(time, signal1)
3 plt.plot(time, signal2, color="r")
4
5 plt.fill_between(time, signal3, color="y")
6
7 plt.xlabel("Time (s)")
8 plt.ylabel("Amplitude")
9 plt.show()

```

In the lines above we plotted the 3 signals. *signal3* is plotted as the area between the value of the amplitude and the Ox axis.

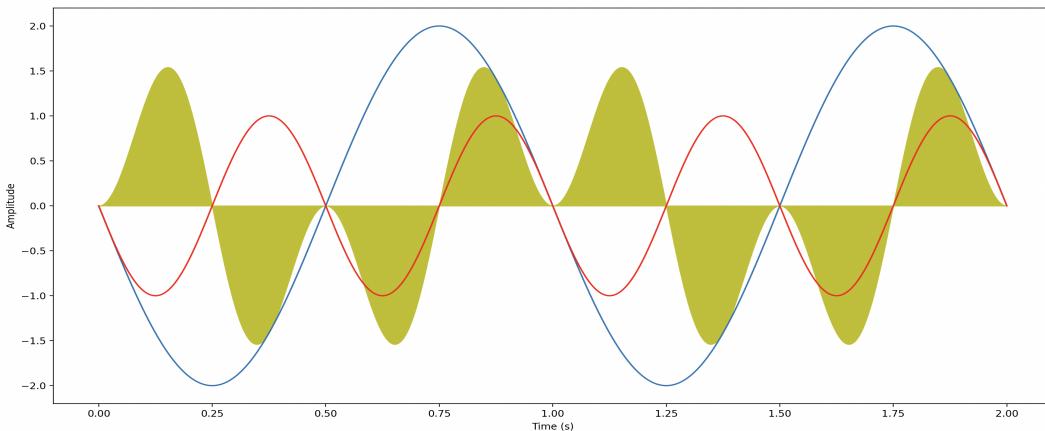


Figura 1.25: Integral of 2 unrelated frequencies multiplied

We can observe that if we were to calculate the integral of *signal3*, which is graphically represented in the figure above as the yellow area, it would be 0.

Now, let's change the values of *signal1* and *signal2* to be of 2 frequencies which are multiples of one another and see how the value of the integral changes.

```

1
2 signal1 = create_sinusoidal_function(frequency=1, time = time, phase = 0 )
    + create_sinusoidal_function(frequency=2, time = time, phase = 0)
3 signal2 = create_sinusoidal_function(frequency=1, time = time, phase = 0)
4 signal3 = signal1 * signal2

```

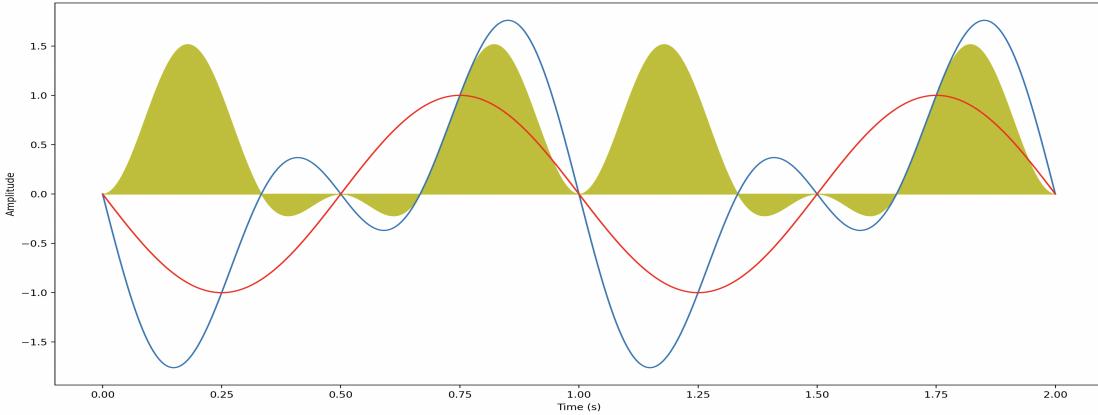


Figura 1.26: Integral of 2 related frequencies multiplied

What we can observe is that the integral is no longer equal to 0. This relates to the \hat{g} function. The value of this integral is the value of the magnitude.

Discrete Fourier Transform

The formulas used above are continuous, but we deal with discrete values in the digital world. These are the discrete counterparts for the continuous functions:

$$\hat{g} \text{ becomes } \hat{x}(f) = \sum_{n=0}^{N-1} x(n) * e^{-i2\pi ft}$$

Observation: the Discrete Fourier Transform produces artifacts above the Nyquist frequency. Thus, we must choose a sample rate big enough to cover the whole desired spectrum.

Capitolul 2

Titlul celui de-al doilea capitol

Facilisi nullam vehicula ipsum a arcu. Purus semper eget duis at tellus at. Adipiscing tristique risus nec feugiat. Eu volutpat odio facilisis mauris sit. Porta nibh venenatis cras sed. Penatibus et magnis dis parturient. Sollicitudin aliquam ultrices sagittis orci a. Senectus et netus et malesuada fames ac turpis egestas integer. Cras tincidunt lobortis feugiat vivamus at augue eget arcu dictum. Leo vel fringilla est ullamcorper eget nulla facilisi etiam dignissim. Nulla aliquet enim tortor at auctor urna nunc id cursus. Elit duis tristique sollicitudin nibh. Sagittis nisl rhoncus mattis rhoncus urna neque viverra. Convallis posuere morbi leo urna molestie at. Quisque egestas diam in arcu cursus euismod.

2.1 Titlul secțiunii 1

A diam sollicitudin tempor id eu nisl. Hac habitasse platea dictumst vestibulum. Integer enim neque volutpat ac tincidunt. Facilisi nullam vehicula ipsum a arcu cursus vitae congue. Vel turpis nunc eget lorem. Vestibulum mattis ullamcorper velit sed ullamcorper morbi tincidunt ornare. Nunc sed blandit libero volutpat. Sit amet luctus venenatis lectus magna fringilla urna porttitor. Hac habitasse platea dictumst quisque sagittis purus. Sed faucibus turpis in eu mi bibendum neque egestas. Vel orci porta non pulvinar neque laoreet suspendisse interdum consectetur. Erat nam at lectus urna duis convallis convallis tellus id. Tristique sollicitudin nibh sit amet commodo nulla facilisi nullam vehicula. Etiam dignissim diam quis enim lobortis scelerisque. Nunc congue nisi vitae suscipit tellus mauris a diam maecenas. Lacus viverra vitae congue eu consequat ac felis donec. Mauris sit amet massa vitae tortor condimentum. Mauris

augue neque gravida in. Lorem ipsum dolor sit amet. Arcu dui vivamus arcu felis bibendum ut tristique et.

2.2 Titlul secțiunii 2

Sit amet mauris commodo quis imperdiet massa tincidunt nunc pulvinar. Ligula ullamcorper malesuada proin libero nunc consequat interdum. Mauris a diam maece-nas sed enim ut. Ut sem nulla pharetra diam sit amet nisl suscipit adipiscing. Leo duis ut diam quam nulla. Neque ornare aenean euismod elementum. Vitae sapien pellen-tesque habitant morbi tristique senectus. Lectus magna fringilla urna porttitor rhoncus dolor purus non enim. Egestas sed sed risus pretium quam vulputate dignissim sus-pendisse in. At quis risus sed vulputate odio ut enim. Hac habitasse platea dictumst quisque sagittis. Lectus vestibulum mattis ullamcorper velit sed. Massa vitae tortor condimentum lacinia quis vel eros donec ac. Vulputate dignissim suspendisse in est ante. Sed faucibus turpis in eu mi bibendum neque. Enim eu turpis egestas pretium aenean pharetra magna. Tellus mauris a diam maecenas.

2.3 Titlul secțiunii 3

Faucibus ornare suspendisse sed nisi lacus sed. Mi in nulla posuere sollicitudin aliquam ultrices. Lacus suspendisse faucibus interdum posuere lorem ipsum dolor sit amet. Odio tempor orci dapibus ultrices in iaculis nunc sed augue. Congue eu conse-sequat ac felis donec et odio. Enim ut sem viverra aliquet eget sit amet. Sit amet con-sectetur adipiscing elit duis tristique sollicitudin. Quis blandit turpis cursus in. Cras fermentum odio eu feugiat pretium nibh ipsum consequat nisl. Non curabitur gravida arcu ac tortor dignissim convallis aenean. Porta non pulvinar neque laoreet suspen-disse interdum consectetur libero id. Lacus viverra vitae congue eu consequat ac felis. Vulputate dignissim suspendisse in est ante in nibh mauris. Amet mauris commodo quis imperdiet massa. Varius sit amet mattis vulputate enim nulla aliquet. Pellen-tesque diam volutpat commodo sed egestas egestas. Amet est placerat in egestas erat imperdiet sed euismod. Scelerisque varius morbi enim nunc faucibus a pellentesque sit. Ut sem viverra aliquet eget sit amet tellus cras. Sem integer vitae justo eget magna fermentum iaculis eu.

Capitolul 3

Titlul celui de-al treilea capitol

Amet venenatis urna cursus eget. Quam vulputate dignissim suspendisse in est ante. Proin nibh nisl condimentum id. Egestas maecenas pharetra convallis posuere morbi. Risus viverra adipiscing at in. Vulputate eu scelerisque felis imperdiet. Cras adipiscing enim eu turpis egestas pretium aenean pharetra. In aliquam sem fringilla ut morbi tincidunt augue. Montes nascetur ridiculus mus mauris. Viverra accumsan in nisl nisi scelerisque eu ultrices vitae. In nibh mauris cursus mattis molestie a iaculis. Interdum consectetur libero id faucibus nisl tincidunt eget. Gravida in fermentum et sollicitudin ac orci. Suscipit adipiscing bibendum est ultricies. Etiam non quam lacus suspendisse. Leo urna molestie at elementum eu facilisis sed odio morbi. Egestas congue quisque egestas diam in arcu cursus. Amet consectetur adipiscing elit ut aliquam purus.

3.1 Titlul secțiunii 1

Eros donec ac odio tempor. Facilisi morbi tempus iaculis urna id volutpat. Faucibus in ornare quam viverra orci sagittis eu. Amet tellus cras adipiscing enim eu turpis egestas. Integer feugiat scelerisque varius morbi. Platea dictumst vestibulum rhoncus est pellentesque elit ullamcorper dignissim. Bibendum arcu vitae elementum curabitur. Eu nisl nunc mi ipsum faucibus. Id aliquet lectus proin nibh nisl condimentum id venenatis a. Cras adipiscing enim eu turpis egestas pretium. Quisque non tellus orci ac auctor augue mauris augue. Malesuada pellentesque elit eget gravida cum. Ut lectus arcu bibendum at. Massa id neque aliquam vestibulum morbi blandit. Posuere ac ut consequat semper viverra nam. Viverra adipiscing at in tellus integer feugiat

scelerisque varius morbi. Morbi enim nunc faucibus a pellentesque sit amet porttitor eget. Eu feugiat pretium nibh ipsum consequat nisl vel. Nisl purus in mollis nunc sed.

3.2 Titlul secțiunii 2

Elementum sagittis vitae et leo duis ut diam quam nulla. Purus sit amet volutpat consequat mauris nunc. Tincidunt augue interdum velit euismod in pellentesque massa. Nunc sed augue lacus viverra vitae congue. Porttitor leo a diam sollicitudin. Faucibus pulvinar elementum integer enim. Adipiscing bibendum est ultricies integer quis auctor elit. Blandit aliquam etiam erat velit scelerisque in. A iaculis at erat pellentesque adipiscing commodo elit at. Erat nam at lectus urna duis. Consequat ac felis donec et. Fermentum posuere urna nec tincidunt praesent semper feugiat nibh sed. Proin gravida hendrerit lectus a. Pretium viverra suspendisse potenti nullam ac tortor vitae purus. Arcu cursus euismod quis viverra nibh cras pulvinar mattis. Gravida arcu ac tortor dignissim convallis aenean. Quam nulla porttitor massa id neque aliquam vestibulum morbi. Sed viverra ipsum nunc aliquet. Quis enim lobortis scelerisque fermentum dui faucibus in.

Concluzii

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Nunc mattis enim ut tellus elementum sagittis vitae et. Placerat in egestas erat imperdiet sed euismod. Urna id volutpat lacus laoreet non curabitur gravida. Blandit turpis cursus in hac habitasse platea. Eget nunc lobortis mattis aliquam faucibus. Est pellentesque elit ullamcorper dignissim cras tincidunt lobortis feugiat. Viverra maecenas accumsan lacus vel facilisis volutpat est. Non odio euismod lacinia at quis risus sed vulputate odio. Consequat ac felis donec et odio pellentesque diam volutpat commodo. Etiam sit amet nisl purus in. Tortor condimentum lacinia quis vel eros donec. Phasellus egestas tellus rutrum tellus pellentesque eu tincidunt. Aliquam id diam maecenas ultricies mi eget mauris pharetra. Enim eu turpis egestas pretium.

Bibliografie

- Author1, *Book1*, 2018
- Author2, *Boook2*, 2017
- <https://en.wikipedia.org/wiki/MIDI>