

# ■ 3 ■

## Soinuaren prozesamendu digitalerako sarrera

---

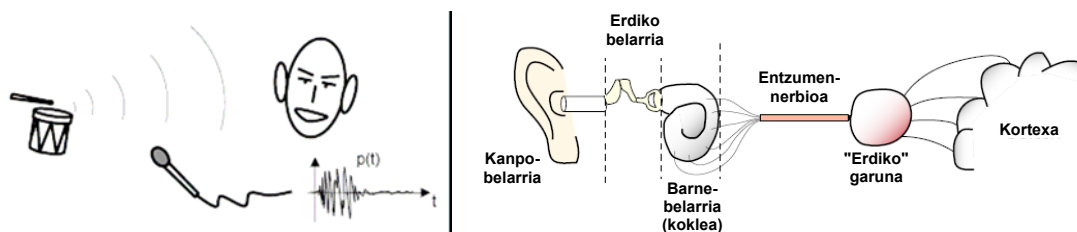
### SARRERA

Gaur egungo hainbat sistema digitalek soinuak prozesatzen dituzte. Laborategi-saio honetan, MATLAB programa erabiliko dugu; programa horrek soinu-seinaleak eraikitzea, entzutea eta irudikatzea ahalbidetzen ditu. Hala, soinu-seinaleak sortu eta grabatu egingo ditugu, eta nota musikalak identifikatu.

### DESKRIBAPENA

#### 1. Soinua: zer da? nola gorde daiteke soinua konputagailu batean?

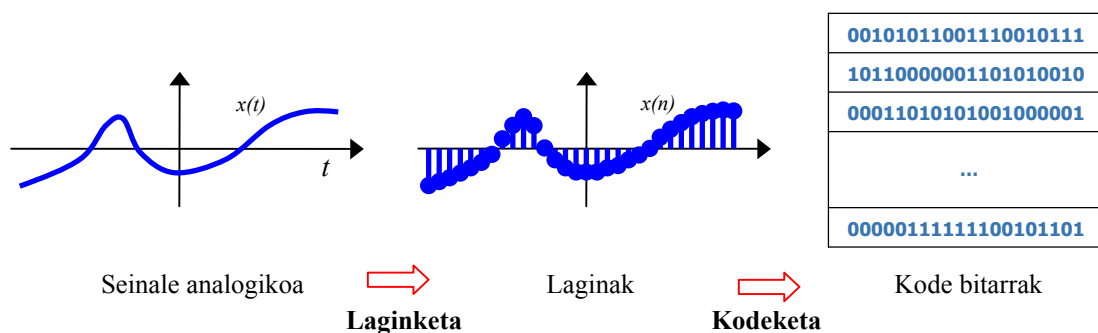
Haizearen mugimenduak presio-uhinak (uhin akustikoak), soinua alegia, sortzen ditu. Soinua da entzumenaren bidez gizakiok aditzen duguna, eta horri ezker gure ingurunetik hainbat informazio jasotzen dugu.



**1. irudia.** Soinuaren prozesatzea.

Gizakien entzumen-sistema oso konplexua da. Irudian ageri den moduan, entzumenaren kanpo-organoek (kanpo- eta erdiko belarriak) jasotzen dituzte uhin akustikoak, estimulu elektriko bihurtzen dira frekuentzia edo maiztasun deskonposaketa baten bidez (koklean), eta burmuinera transmititzen dira, entzumen-nerbioari esker, non prozesatu egiten baitira. Burmuinak soinuak prozesatzeko duen gaitasuna izugarria da, eta sistema artifizialek ezin dute funtzionamendu hori guztiz imitatu.

Mikrofonoak erabiltzen dira soinuak jasotzeko, uhin akustikoak (mugimenduak) seinale elektriko bihurtzen baitute. Soinua sortzeko, berriz, bozgorailuak erabiltzen dira, alderantzizko eragiketa egiten baitute: seinale elektrikoak uhin akustiko bihurtzea. Konputagailuak mikrofono batetik jasotako seinale elektrikoaren prozesua dezan, seinale analogikoa seinale digital bihurtu behar da; horretarako, seinale analogikoaren hainbat **lagin** hartzen dira eta, gero, kode bitarrak esleitzen zaizkie laginei.



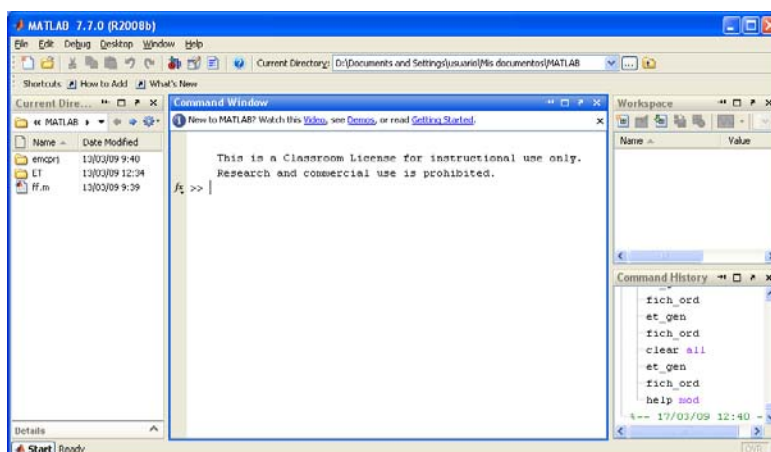
**2. irudia.** Seinale analogikoa seinale digital bihurtzeko prozesua.

Bihurketa digitalak bi ezaugarri nagusi ditu: lagintze-maiztasuna –zenbat lagin hartzen diren segundoko– eta laginak kodetzeko erabiltzen den bit kopurua. Eraitza gisa, konputagailu batek prozesu dezakeen kode bitarren sekuentzia bat lortzen da. Soinua estereo bada, bi seinale digitalizatu behar dira (ezkerreko eta eskuineko kanaletakoak, alegia).

## 2. MATLAB: zer da?

MATLAB (MATrix LABoratory) datuak konputagailuetan tratatzeko tresna matematiko oso ahaltsua da, batez ere matrizeekin kalkulu konplexuak egiteko. Praktika honetan, 7.7.0 bertsioaren seinale-prozesamendurako aurredefinitutako funtzio-sorta (*toolbox*) erabiliko dugu. MATLABen funtzioak banan-banan exekuta daitezke, modu interaktiboan, komando batzuen bidez, edo haiekin programak sortu. Egokia da, beraz, eskaintzen dituen funtzioak ezagutzera (“on line” laguntza ere badago).

MATLAB PC/Windows sistema batean abiatzeko, nahikoa da idazmahaiari dagoen ikono hau sakatzeko. MATLABi dagokion leihoa irekiko da, eta barruan komando-leihoa:



Hortxe idaztiko ditugu banan-banan exekutatu nahi ditugun komandoak; » ikurrak programa komando baten zain dagoela adierazten du. Hori egiaztatzeko, egin ezazu proba hau:

```
» 2+2
ans =
4
```

Beraz, MATLABek batzen badaki! Hala ere, laborategi-saio honetan, soilik soinu-aplikazio simple batzuk aztertuko ditugu, ez kalkulurako aukerak.

## 3. Soinua jasotzea, irudikatzea eta entzutea, MATLAB erabiliz.

Windows sistema eragileak soinuak grabatzea ahalbidetzen du, soinu-txartela instalatuta badugu eta mikrofono bat konektatzen badugu. Sakatu horretarako Hasi/Programak/"Complementos"/Denbora-pasa/"Grabadora de sonidos", eta ahotsa erabiliz soinu seinale bat graba daiteke. Zenbait kasutan, konfiguratu eta aktibatu behar dira mikrofonoaren

sarrera eta bozgorailuaren irteera Bolumenaren kontrola erabiliz. Grabatutako soinua `.wav` fitxategi batean gorde daiteke. Egiazta ezazu hori guztia.

Baina beste aukera da MATLAB erabiltzea soinuak grabatzeko. Adibidez, mikrofono bat konektatu eta exekutatu honako hau:



```
>> fs = 8000;
>> esaldia = wavrecord(5*fs, fs, 'int16');
>> wavplay(esaldia, fs)
```

`wavrecord` komandoa erabiliz 5 segundoko soinu bat grabatu duzu, lagintze-maiztasuna 8.000 Hz izan da, eta informazioa 16 bitetan kodetu da; eta `wavplay` komandoaren bidez, soinu hori bozgorailuetara bidali duzu. Grabatu duzun seinalea "ikus" nahi baduzu, exekutatu:

```
>> plot(esaldia)
```

Leiho grafiko bat irekitzen da, non grabatutako soinua grafikoki ageri baita. Grafikoa, MATLABek seinalearen laginak lotu egiten ditu marra baten bidez, eta X ardatzean laginaren zenbakia adierazten da. X ardatzean denbora adieraztea nahi baduzu, tekleatu:

```
>> N = 0:(length(esaldia)-1);
>> tt = N/fs;
>> plot(tt, esaldia)
```

Seinalearen zati bat handitu nahi izanez gero, *zoom* egin daiteke menu-barrako   ikonoak sakatuz.

Azkenik, batzuetan interesgarria izan daiteke laginak bananduta irudikatzea. Adibidez:

```
>> plot(tt, esaldia, '.')
```

edo bi gauzak aldi berean:

```
>> plot(tt, esaldia, '.-')
```

MATLABek `.wav` fitxategiak irakurtzea eta idaztea ahalbidetzen du. Formatu hori soinua gordetzeko windows-en formatu arrunta da, eta erabili behar diren komandoak `wavwrite` eta `wavread` dira. Fitxategi horietan, seinalearen laginez gain informazio gehiago gordetzen da, hala nola: lagintze-maiztasuna, bereizmena, etab. Hurrengo adibidean, musika soinu zati bat (piano batez sortuta) irakurtzen dugu, eta gero irudikatu eta entzun egiten dugu:

```
>> [melo fs bits] = wavread('melodia.wav');
>> fs
>> bits
>> plot(melo)
>> wavplay(melo, fs)
```

## 4. Soinuaren sintesia eta pertzepzioa.

Atal honetan, soinu-seinale batzuk artifizialki sortuko ditugu, eta gizakiok soinu- seinaleak nola aditzen ditugun aztertuko dugu. Laginei balioak esleituko dizkiegu, seinale erreal bat digitalizatzen denean egiten den moduan; beraz, aurretik lagintze-maiztasuna finkatu behar dugu.

### A. Seinale sinusoidal hutsak

Adibidez: segundo bateko 440 Hz-eko seinale sinusoidal bat sortzeko, seinale hori aditzeko lagintze-maiztasuna 22.050 Hz-ekoa dela kontuan harturik, honako komando hauek exekutatu behar dira:

```
>> tt = 0:1/22050:1;           Lagintze-uneen bektorea
>> s = sin(2*pi*440*tt);      Seinale sinusoidal: f=440 Hz eta fs=220.50 Hz
>> plot(tt,s)                 Seinale osoa irudikatzen dugu.
>> plot(tt(1:500),s(1:500))  Seinale zati bat irudikatzen dugu.
>> wavplay(s,22050)          Soinua aditzen dugu.
```

Proba gehiago modu erosoagoan egiteko, antzeko eragiketak egiten dituen beste funtzio bat erabiliko dugu. Funtzio hori `tono.m` deitzen da (MATLAB editorearen bidez idatzi dugu eta ikasleen karpetan utzi dugu), eta komando-lerrotik exekuta daiteke. Hona hemen funtzio horren kodea:

```
function y = tono(f, d)
% TONO: sinusoidal huts bat sortzen duen funtzioa
% y = tono(f, d) seinale sinusoidal baten laginak sortzen ditu,
% f seinalearen maiztasuna da, Hz-etan, eta d iraupena, segundotan
% lagintze-maiztasuna fs = 22.050 Hz da
%
% tono(f, d) exekutatzen bada (hau da, irteera ez bada gordetzen)
% seinalea aditu egiten da

fs = 22050;
tt = 0:1/fs:d;
s = cos(2*pi*f*tt);

if nargin == 0,
    wavplay(s, fs)
else
    y = s;
end
```

Orain, exekuta ezazu:

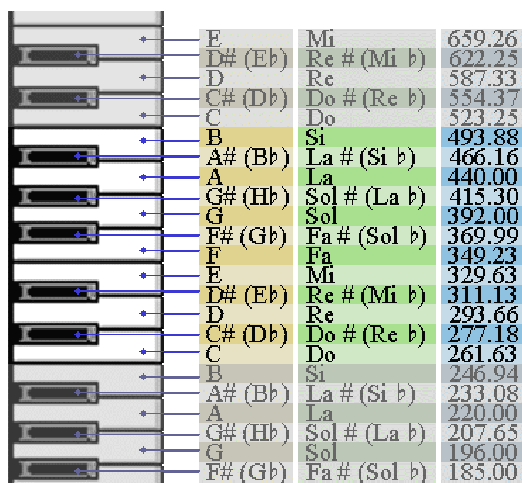
```
>> help tono
>> tono(440,1)
```

Sortu eta aditu maiztasun desberdineko seinaleak **tono** funtzioa erabiliz; adibidez: 50, 100, 500, 1000, 2000... Hz-eko seinaleak. Soinuaren hainbat ezaugarri nabaritu duzu:

- Aditzen duzun tonua edo "soinuaren altuera" seinalearen maiztasunaren arabera da.
- Seinale-maiztasunaren arabera, soinuaren bolumena desberdina da, nahiz eta anplitudea beti bera den (eskuarki, gizakiok hobe aditzen ditugun soinuak 1 eta 5 kHz-en artean daude).
- Maiztasuna handiagoa den heinean, maiztasun gehikuntzek handiagoak izan behar dute antzeko tonu-igoerak nabaritzeko.

Maiztasuna ezin da  $f_s/2$  baino handiago izan, *aliasing* efektua agertzen delako; hau da, erabili dugun lagintze-maiztasuna ez da nahikoa seinalea ondo digitalizatzeko, eta aditzen dugun maiztasuna, handiago izan beharrean, txikiagoa izango da.

Ezagutzen ditugun musika-notei maiztasun jakineko uhinak dagozkie (adibidez, LA notari 440 Hz dagoio). Irudian ageri da eskala bateko zazpi notei dagozkien maiztasunak.



E	Mi	659.26
D# (Eb)	Re # (Mi b)	622.25
D	Re	587.33
C# (Db)	Do # (Re b)	554.37
C	Do	523.25
B	Si	493.88
A# (Bb)	La # (Si b)	466.16
A	La	440.00
G# (Gb)	Sol # (La b)	415.30
G	Sol	392.00
F# (Gb)	Fa # (Sol b)	369.99
F	Fa	349.23
E	Mi	329.63
D# (Eb)	Re # (Mi b)	311.13
D	Re	293.66
C# (Db)	Do # (Re b)	277.18
C	Do	261.63
B	Si	246.94
A# (Bb)	La # (Si b)	233.08
A	La	220.00
G# (Gb)	Sol # (La b)	207.65
G	Sol	196.00
F# (Gb)	Fa # (Sol b)	185.00

**3. irudia.** Musika-notei dagozkien maiztasunak.

Irakur ezazu ikasle-karpetan dauden `ejemplo1.m` eta `ejemplo2.m` fitxategiak. Exekuta itzazu eta ikusiko duzu zein erraz sor daitekeen musika.

## B. Harmonikoen konbinazioak

`Ejemplo3.m` fitxategian idatzita dauden komandoek  $f_0 = 440$  Hz-eko **s1** seinalea sortzen dute, eta baita bere **harmonikoak** ere, hau da,  $2xf_0 = 880$  Hz-eko,  $3xf_0 = 1320$  Hz-eko..., **s2**, **s3**..., seinale sinusoidalak. Seinale horiek bi modutan entzuten dira: banan-banan, eta batuta, seinale bakoitzari pisu desberdinak emanez.

Ikusiko duzu seinale konbinatua periodikoa dela, eta haren periodoa seinale nagusiarena (s1-ena) dela. Aditzerakoan bi seinale horien tonua berdina da, baina tinbrea desberdina da. Dena den, sinplifikazio handiak egiten ari gara. Oinarrizko maiztasunaren pertzepzioa (*pitch*) entzule bakoitzaren araberakoa da; eskuarki, uhinean pisu gehien duen maiztasuna da baina batzuetan ez da hori gertatzen, eta, gainera, entzuleak maiztasun bat baino gehiago entzun ditzake.

## 5. Melodia baten notak identifikatzea

Aurreko atalean, soinu-seinalearen sintesia egin dugu. Orain, alderantzizko prozesuari ekingo diogu, hau da, **soinu-seinale baten analisia**, haren ezaugarriak zehazteko. Ikusi dugunez, aditzen dugun tonua seinalearen maiztasun nagusiaren mende dago. Tonu hori oso garrantzitsua da musika arloan, nota bakoitza maiztasun jakin bati dagokio eta. Hori egiaztatzeko, aurretik erabili dugun **melodia.wav** soinu-seinalea erabiliko dugu, hots, pianoz sortutako musika zati bat.

Hasteko, irudikatu seinale osoa eta markatu laginak:

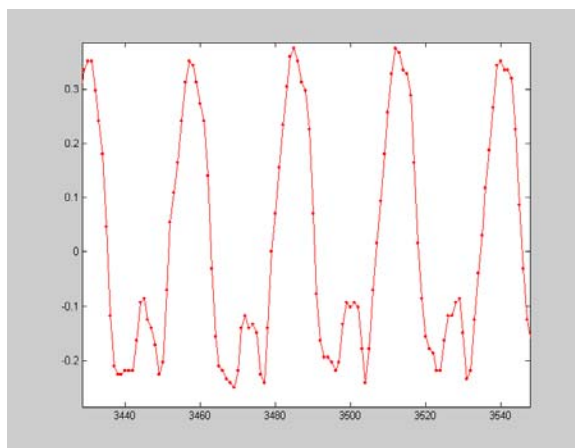
```
>> [melo fs] = wavread('melodia.wav');
>> plot(melo, 'r.-')
>> wavplay(melo, fs)
```

Seinalearen anplitudea ikusita, argi eta garbi ageri da zein unetan sakatzen diren pianoaren teklak. *Zoom*-a erabiliz, seinalearen zati txikiagoak azter ditzakezu. Horrela, laginak kontatuz, nota bakoitzari dagokion periodoa eta maiztasuna kalkula ditzakezu.

### >> ARIKETA

Lehenbiziko notaren zati bat aztertzen baduzu (ikus ondoko irudia), 27 lagin dituela konturatu zara.

Lagintze-maiztasuna  $fs = 11.025$  Hz denez, **zein da nota horren oinarrizko maiztasuna? Ondorioz, zein nota izan daiteke?**

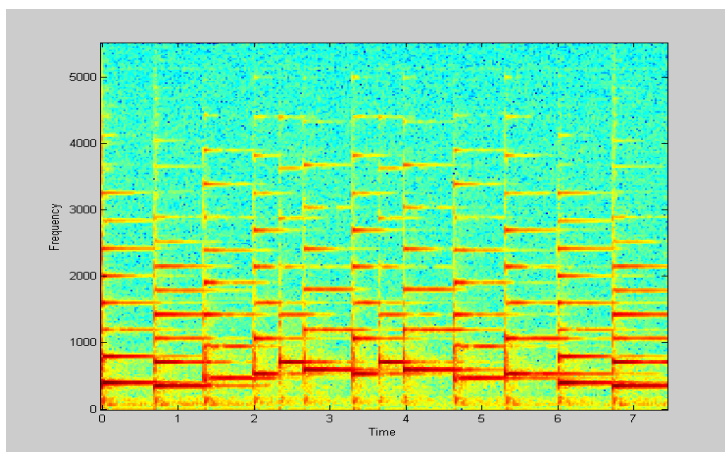


**Errepika ezazu prozesua bigarren notarentzat.**

Musika zati baten noten maiztasuna aztertzeko beste modu bat da haren **espektrograma** lortzea eta analizatzea. Espektrograma lortzeko, exekuta ezazu:

```
>> specgram(melo, 512, fs, 512, 212)
```

Adibide honetan, MATLABek seinalea zatitu egiten du (zati bakoitzean, 512 puntu) eta zati bakoitzean maiztasunei buruzko informazioa kalkulatu du, ingeniartzan maiz erabiltzen den funtzioa bat aplikatuz: Fourieren transformatua. Funtzio horren eragina nolabait gizakiok "koklean" egiten dugun maiztasun-deskonposaketaren antzekoa da.



Funtzioa konplexua da, eta ezin dugun orain azaldu, baina nahikoa da onartzea adierazpen grafikoan seinalearen maiztasun-konposaketaren aldaketak (ardatz bertikala) denboran zehar (ardatz horizontala) ikusten direla. Marra gorriek edo gorrien ingurukoek adierazten dute maiztasun jakin baten sinusoidalaren kontribuzioa handia dela zati horretan. Zentzu horretan, adierazpen grafiko honetako beheko aldean dauden marrek noten oinarriko maiztasunaren aldaketa denboran zehar adierazten dute.

Hori guztia kontuan harturik, eta *zoom*-a erabiliz, identifika itzazu musika zati honen 13 notak (maiztasuna eta iraupena). Osatu ezazu honako taula hau:

	Oinarriko maiztasun: (Hz)	Iraupena (ms)
1	400	0.7
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		



Azkenik, **tono** funtzioa erabil dezakegu musika horren sintesia artifizialki egiteko, seinale sinusoidalak konbinatuz. Horretarako, nota bakoitzari dagokion seinalea sortu beharko duzue, adibidez:

```
>> N1 = tono(400,0.7);  
>> ...
```

13 notak sortuta eta gero, kateatu eta entzunarazi:

```
>> wavplay([N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13],22050)
```

Doinu bera sortu al duzu?

## **OHARRAK**

