



HASKFU

rádióamatőr klub

Rádióamatőr tanfolyamot segítő jegyzet, egyelőre kidolgozás alatt
2021 tavasz

Szabó Áron HA1FLX, Kiss Ádám HA8KDA,
Bazsó Márton HA7BM, Keresztes Botond

Modulációs gyakorlat

Tartalom

1. Bevezető	2
2. Előkészületek	2
3. Elméleti összefoglaló	2
3.1. Keverés szükségessége	2
3.1.1. Tükörfrekvencia	3
3.1.2. Elgondolkodtató megjegyzések	4
3.2. Keverés matematikája	4
3.3. Egyszerű modulációk konstellációs ábrája	5
4. Feladatok	7
4.1. Szinusz jel generálása, ábrázolása	7
4.2. Keverés	10
4.3. IQ moduláció	11
4.3.1. Fázismoduláció	11
4.4. Újramintavételezés	13

1. Bevezető

A gyakorlat célja bemutatni a különböző modulációs eljárások idő- és frekvenciartománybeli viselkedését, illetve matematikai leírásukat. Ezzel megalapozzuk a szoftverrádiózáshoz szükséges tudást.

2. Előkészületek

A gyakorlat Octave-ban és Matlab-ban is elvégezhető, a használt parancsok mindkettőben működőképeseek.

Octave telepítés:

- **GNU Octave telepítése**
<https://wiki.octave.org/Category:Installation>

- **Signal package telepítése**

A jelfeldolgozó könyvtárak a Matlab-ban a telepítéssel érkeznek, Octave esetén csomagként érhetőek el, és használat előtt be kell tölteni őket. A `signal` könyvtár a `control`-ra is támaszkodik, így a megfelelő sorrendben telepítsük őket.

Írjuk be az Octave parancssorába:

```
pkg install -forge control
```

```
pkg install -forge signal
```

Az Octave indulásakor ez nem töltődik be automatikusan, mielőtt jelfeldolgozási függvényeket használnánk, töltsük be a `signal` csomagot:

```
pkg load signal
```

A könyvtárat csomagkezelőből is telepítheted:

```
sudo apt-get install octave-signal #Ubuntu
```

```
sudo pacman -S octave-signal #Arch vagy Manjaro
```

```
sudo zypper install octave-forge-signal #RPM (OpenSUSE)
```

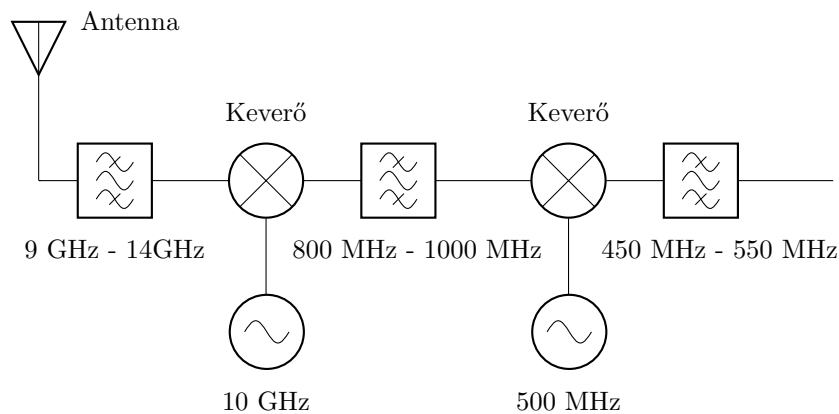
3. Elméleti összefoglaló

3.1. Keverés szükségessége

Tegyük fel, hogy van egy modulált jelünk valahol a rádiós spektrumban. (Pl. 540 kHz-en a Kossuth rádió vagy a 12 GHz-en egy prémium műhold csatorna kedvenc szolgáltatóknál) Ezt szeretnénk venni. A vett jel szempontjából zavaró információ az összes többi adás, illetve az egyéb

frekvenciákon mérhető zaj, ezeket szükséges kiszűrni a tökéletes vételhez. A probléma, hogy gyakorlatilag kivitelezhetetlen nagyon keskeny szűrő magas frekvenciákon. Például egy 1 MHz sávszélességű jelet kiszűrni GHz-es tartományban már lehetetlen. Az úgynevezett relatív sávszélesség (sávszélesség osztva sávközéppel) egy ezred. Sok gyakorlattal és minőségi alkatrészekből építkezve körülbelül egy tizedes relatív sávszélességig tudunk megvalósítani áramkört.

A problémát egy csellel oldjuk meg. Nagyjából kiszűrjük a jelet a saját sávjában, esetleg még erősítjük is (itt a sorrend lehet fordított!). Majd egy keverőt helyezünk el. A keverő frekvenciában eltolja a bemeneti jelet az ő oszcillátor bemenetének frekvenciájával mindkét irányba. A nem használt keverési terméket kiszűrjük. (Kellően nagy keverési frekvencia esetén ez messze fog kerülni a használttól, így könnyű hozzá szűrőt tervezni). Így a jelünk alacsonyabb frekvenciára került, ahol már ugyanakkora ügyességgel és anyagminőséggel keskenyebb sávú szűrőt tudunk építeni. Ezzel a módszerrel ki tudjuk szűrni a számunkra fontos jel körül a többi zavaró jelet.



1. ábra. Szuperheterodin struktúra

3.1.1. Tükörfrekvencia

$$\cos(\omega_1 t) \cdot \cos(\omega_2 t) = \frac{1}{2} \cos((\omega_1 + \omega_2) t) + \frac{1}{2} \cos((\omega_1 - \omega_2) t) \quad (1)$$

Létezik úgynevezett tükörfrekvencia. Ez az a sáv, amit a keverő ugyanarra a frekvenciára kever, mint a hasznos jelet. Vegyünk például egy 100 MHz-es adást, amit 10 MHz-re szeretnénk lekeverni. A modulációs tétel (1) értelmében ezt például úgy érhetjük el, hogy a 100 MHz-es adást megszorozzuk egy 90 MHz-es oszcillátorral (a másik lehetőség egy 110 MHz-es helyi

oszillátor jel lenne), így az 10 MHz-en megjelenik mint különbségi frekvencia. Viszont ekkor a 80 MHz-en lévő spektrumrész is bekeverődik 10 MHz-re, mint összegfrekvencia. Az így fellépő, de oda nem illő részt hívjuk tükörátlapolódásnak.

Ezt az eredeti, ide lapolódó spektrumrészt ki kell szűrni a keverés előtt, különben áthallás lesz két adó között. A keverő kimenetét középfrekvenciának szokták hívni, az ezt követő szűrőt KF szűrőnek. (Angolul intermediate freq, és IF filter).

3.1.2. Elgondolkodtató megjegyzések

- Ha hangolható a keverőnk oszillátora, akkor a lefedett sáv nem lehet nagyobb a KF szűrő sávközepének kétszeresénél.
- Ha vevőt építünk, akkor érdemes először felfele keverni, azaz a magasabb frekvenciájú keverési terméket használni, ott szűrni, majd onnan lekeverni, mert így lesz a legnagyobb az ún. tükørszelektivitásunk.
- Digitális jeleknél is fennáll a fenti gondolatmenet, létezik úgynevezett digitális KF is.

3.2. Keverés matematikája

Vizsgáljuk csupán egy szinuszjelet, ennél összetettebb jel felfogható több szinusz összegeként, a lentiek igazak erre az esetre is, csak több példányban értelmezve.

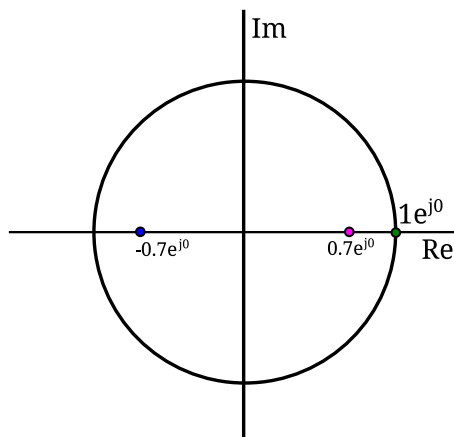
Vegyünk egy vektort, ami forog körbe körbe. Vegyük ennek a vektor csücskének az egyik tengelyre vett árnyékát. A tengelyen a szinusz függvény szerint fog mozogni a pont. Ha ezt a tengelyen mozgó pontot egy ceruzára cseréljük, és a papírt elkezdjük a mozgására merőlegesen lehúzni egyenletes sebességgel, akkor egy ideális szinuszt fogunk kapni. Bizonyítást lásd Euler-tétel.

A keverés során semmi mást nem teszünk, mint próbáljuk felvenni a sebességet a jellel. Azaz a vektor továbbra is forog úgy, ahogy forog, viszont a koordináta rendszert elkezdjük forgatni hozzá. Ha ugyan azon irányba, akkor különbségi keverést csinálunk, ha ellentétesbe, akkor összegzőt. (A valóságban persze a kettőt csak együtt tudjuk csinálni. Valós szinusszal keverünk, ami az Euler-tétel miatt két részre bontható, az egyik az egyik, a másik iránynak felel meg.) Ekkor elképzelhető, hogy a kapott szinuszunk lassul vagy gyorsul. Le tudjuk azonos sebesség esetén nullára is lassítani a látszólagos sebességét. Ekkor a szinusz és a koordináta-rendszerünk fázisától függően egy valamilyen álló fázisban látjuk a vektort.

A koordináta-rendszert, amiben ábrázoljuk a lelassított szinuszt, konstellációs diagramnak nevezzük. A vett jel (tfh. szinuszt) vektorának a végpontját pedig konstellációs pontnak. Szokás a különböző konstellációs ténnyedekhez vagy ennél is finomabb felbontásokhoz digitális szimbólumokat is rendelni, amiket később (miután rászinkronizált a vevő az adó bitkibocsájtási ütemére) bitfolyamként tud értelmezni egy következő fokozat.

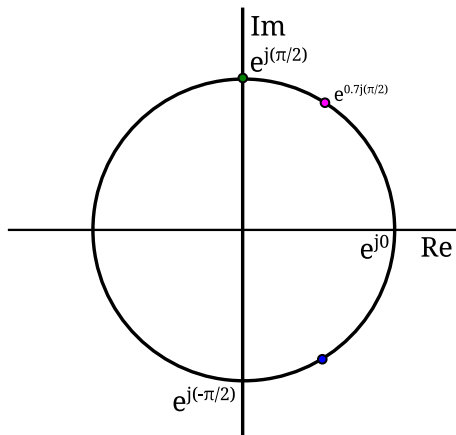
3.3. Egyszerű modulációk konstellációs ábrája

AM Amplitúdó modulált esetben a konstellációs pont szöge áll, azonban az amplitúdója változik. Amennyiben a szög folyamatosan nő enyhén, úgy frekvencia-hibánk van az adóhoz képest, az oszcillátorunk frekvenciája kicsit más.



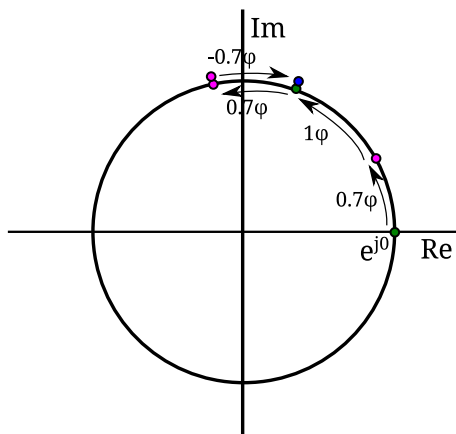
2. ábra. AM konstellációs ábrája

PM Fázis modulációnál az adó néha fázist vált a jelében, ekkor nálunk a lekevert/visszalassított szinusztban is lesz egy fázisugrás. Amennyiben az elvárt szimbólumok helyei itt is egy kör mentén vándorolnak, úgy frekvenciahibánk van.



3. ábra. PM konstellációs ábrája

FM Frekvenciamodulációnál nem szoktuk lekövetni pontosan a frekvenciaváltozását a jelnek. Azt nézzük, hogy épp milyen ütemben gyorsul el tőlünk, vagy lassul vissza hozzánk. Felírva szögfüggvényekkel, hogy épp mekkora szögsebessége van az FM jelnek (ehhez kell az előző vektor értéke is) számolható egy érték, amit megfelelő konstanssal (hangerő) megszorozva már demoduláltuk is a jelet. A frekvenciahiba itt egy konstans eltolást jelent majd a demodulált jelben. Hang esetén ezt nem halljuk ki.



4. ábra. FM konstellációs ábrája

4. Feladatok

4.1. Szinusz jel generálása, ábrázolása

Első feladatunk egy szinusz jel generálása lesz, melyet a $\sin()$ függvénnyel tehetünk meg. Fontos, hogy ez a függvény egy adott érték szinuszát adja vissza, nekünk viszont egy időtartománybeli függvényre van szükségünk.

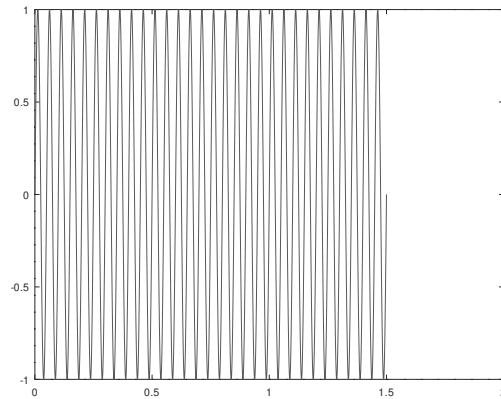
Létre kell hoznunk tehát egy időtengelyt, aminek pontjaiban kiszámoljuk a függvény értékét. Ebből a szempontból fontos meghatározni a mintavételi frekvenciát (F_s), hiszen ebből az értékből számolható, hogy két minta közt mennyi idő fog eltelni. Válasszuk a mintavételi frekvenciát $44\ 100\ Hz$ -re, ami a digitális audió jelfeldolgozásban használatos frekvencia, és a számítógép hangkártyáján könnyen lejátszható.

Ha a minta hosszát (l) másodpercben szeretnénk megadni, a minták számának (N) meghatározásához egyszerűen szorozzuk meg a másodperc értéket a mintavételi frekvenciával. A másodperc időtengelyt (t) úgy kaphatjuk meg, ha az $1:N$ sorozatot elemenként elosztjuk a mintavételi frekvenciával (a transzponálásra azért van szükség, hogy oszlopvektort kapjunk). Ez után a szinusz jel egyszerűen kiszámolható: mivel a szinusz egy periódusa 2π , a szinuszjel pillanatnyi értékét $2\pi \cdot f \cdot t$ -re kell választani, ahol f a szinusz frekvenciája.

Kód 1. Mintavételi frekvencia beállítása és szinuszjel előállítás

```
1 Fs = 44100; % mintaveteli freki
2 l = 1.5; % minta hossza masodpercben
3 N = l * Fs, % mintak szama
4 f = 440; % 20 Hz-es lesz a jel
5 t = ( 1:N )' ./ Fs; % Ido oszlopvektora
6 y = sin( 2 * pi * f * t );
7 plot( t, y );
8 xlim([t(1), t(end)]) %grafikon x-tengely hatarainak beallitasa
```

Ábrázoljuk a Plot függvénnyel időtartományban:



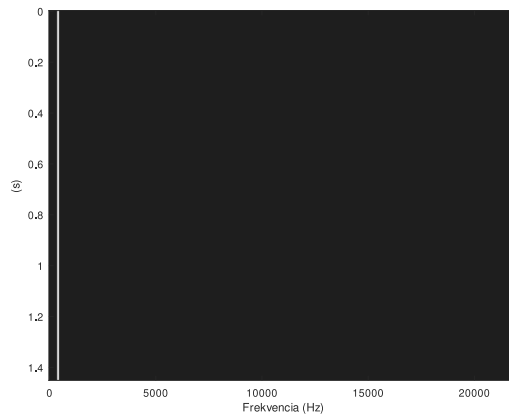
5. ábra. 20 Hz-es szinusz függvény ábrázolása időtartományban

A 20 Hz-es jelet könnyen ábrázoltuk, de a hallható tartomány magasabb frekvencián van. Változtassuk a jel frekvenciáját 440 Hz-re (zenei A hang), és játsszuk ki a számítógép hangkártyáján a jelet:

```
soundsc(y, Fs); %Hang lejatszasa Fs mintaveteli frekvenciával
```

Itt már több információt kapnánk a jelről, ha frekvenciatartományban, vízésés diagramon ábrázolnánk a jelet. A HA5KFU Octave gyakorlathoz csomagolt $wf(y, Fs)$ függvénnyel ezt a rádiózásban megszokott formában rajzolhatjuk ki.

```
wf(y, Fs); %Vizeses diagram kirajzolasa
```

6. ábra. 440 Hz-es szinusz függvény ábrázolása frekvenciatartományban

A vízésés diagram 0-tól a mintavételi frekvencia feléig ábrázolja a frekvenciakomponenseket, az ábrán láthatjuk a konstans szinusz jel képét 440 Hz-nél. Próbálkozzunk meg a jel manipulálásával, hallgassuk meg, ábrázoljuk frekvenciatartományban!

```
y = ( 1 - ( 1 / 1 ) * t ) .* sin( 2 * pi * f * t ); %Fokozatosan halkulo jel
```

```
y = sin( t .* ( f * t ) * 2 * pi ); %Valtozo frekvenciaju szinusz
```

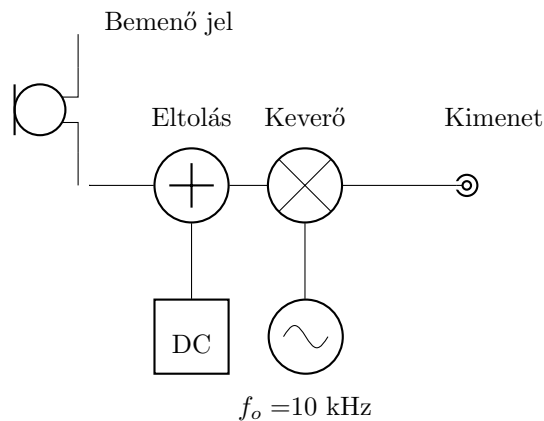
```
y = sin( 2 * pi * f * t ) + sin( 2 * pi * ( 1.5 * f ) * t ); %Tobb szinusz osszege
```

Megpróbálhatunk akár fehér zajt generálni, vagy hangfájlt betölteni.

```
y = rand( N, 1 ); %N elemu zaj
```

```
[y,Fs] = audioread( 'akkord.wav' ); N = length( y ); l=N/Fs; %WAV file beolvasasa
```

4.2. Keverés



7. ábra. A keverő blokkvázlata

Amikor egy jelet egy szinuszoszcillátor jelével keverünk (szorzunk), a jel megjelenik a frekvenciatartományban a két frekvencia összegeként és különbségeként. Ha pl a két jel egy ω_1 frekvenciájú koszinusz vivő, és egy ω_2 frekvenciájú koszinusz moduláló jel, a modulációs tétel alapján:

$$\cos(\omega_1 t) \cdot \cos(\omega_2 t) = \frac{1}{2} \cos(\omega_1 t + \omega_2 t) + \frac{1}{2} \cos(\omega_1 t - \omega_2 t) \quad (2)$$

Nézzük meg a gyakorlatban, próbáljuk meg az *akkord.wav* hangfílet keverni, például egy 10 kHz-es szinusz jellel!

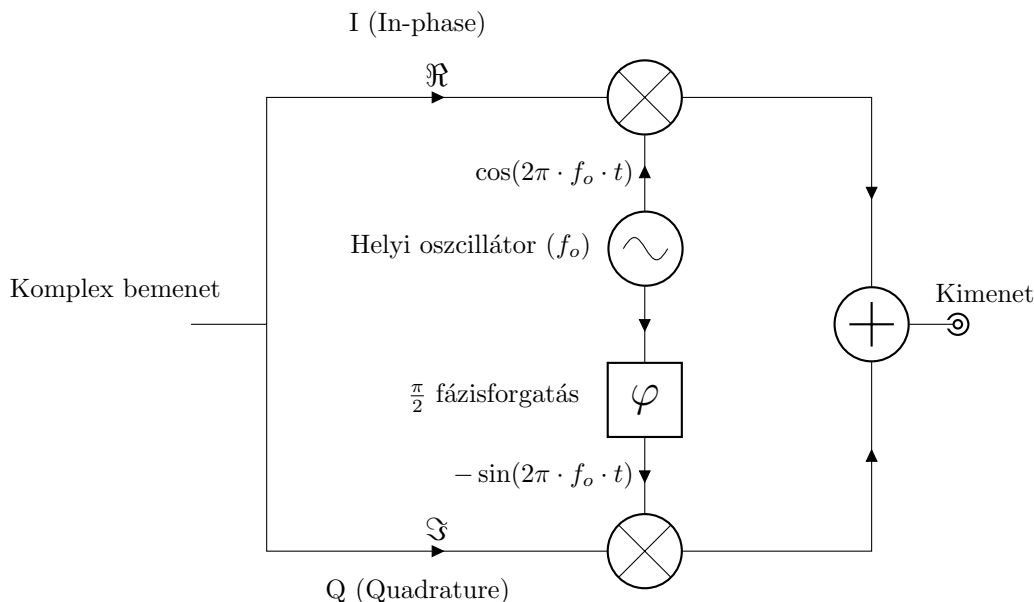
Kód 2. Keverés

```
1 [a,Fs] = audioread('akkord.wav'); N=length(a);
2 f = 10000; % 10 kHz-es lesz a vivo jel
3 t = ( 1:N )' ./ Fs;
4 b = sin( 2 * pi * f * t );
5
6 m = 0.75; % modulacios index
7 dc_offset = (1 + 1-m) * max(a);
8 a_dc = a + dc_offset; % dc eltolas
9
10 y = a_dc .* b; % keveres (a .* b)
11
12 wf(y, Fs);
```

Fontos, hogy eltoljuk az eredeti jelet úgy, hogy a moduláló jel ne menjen 0 szint alá, különben túlmoduláljuk a vivő jelet. A vivő és moduláló jel amplitúdójának kapcsolatát a modulációs mélységgel jellemezzük.

Vajon hogyan kapjuk vissza a kevert jeltől az eredeti információt?

4.3. IQ moduláció



8. ábra. Az IQ modulátor blokkvázlata

Az IQ modulátornál egy komplex jellel moduláljuk a vivő frekvenciát. Ezzel a módszerrel az alapsávi jelet egy komplex fazorként ábrázolhatjuk, így – kellő ügyességgel – tetszőleges moduláció létrehozható.

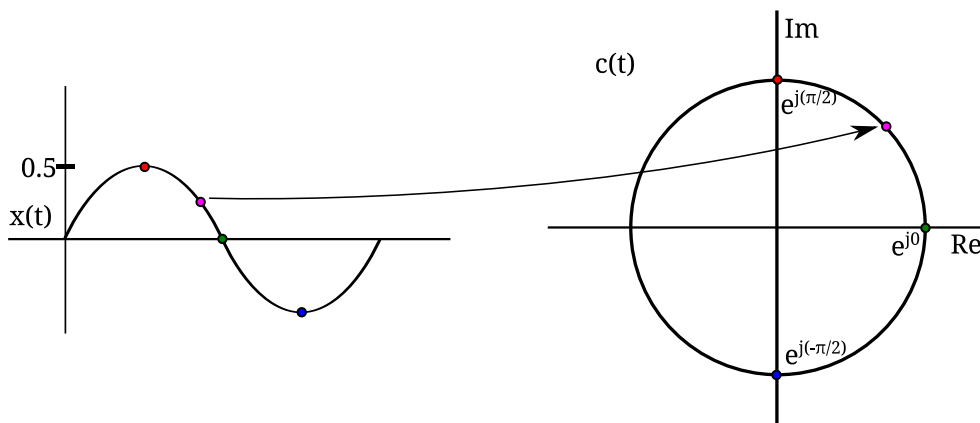
4.3.1. Fázismoduláció

Vegyük át újra hogy is néz ki egy szinusz.

$$y(t) = A \cdot \sin(2\pi \cdot f \cdot t + \phi) \quad (3)$$

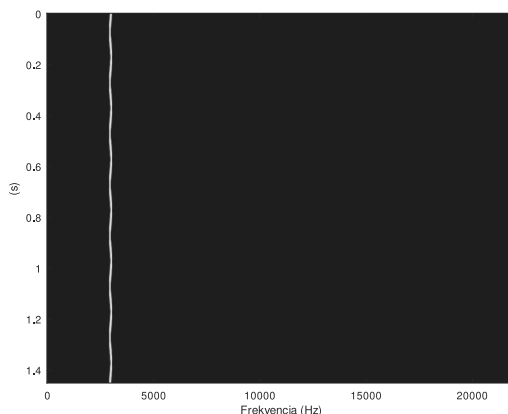
A képletben lévő amplitúdót amplitúdómodulációban, a frekvenciát frekvenciamodulációban használjuk, a fázist pedig fázismodulációban. Talán a fázismodulációt a legnehezebb megfogni élettapasztalatból, mivel a fülünkkel nem halljuk ki a jelek fázisát – furcsa is lenne ha egy méterrel távolabbról másnak hallanánk mások hangját. A fázis jelen esetben is nehezen értelmezhető, mivel a kiadott jel úgymond elveszik, nincs mihez hasonlítani első közelítésben majd vevő oldalon a fázist. A probléma feloldása az, hogy egyrészt eltárolhatjuk a jel mintáit majd vevőoldalon, illetve a fázist tudja majd a vevő a saját, azonos frekvencián futó oszcillátorához képest mérni. Különböző frekvenciákhoz nincs értelme fázist mérni általában, mert a különböző frekvencia miatt fázisban úgymint egymásnak egymástól.

Készítsünk fázismodulátort IQ moduláció segítségével! Legyen az átvinni kívánt (moduláló) jel $x(t)$ (ügyelve arra, hogy $|x(t)| < 1$)! Ekkor a $c(t)$ komplex vektor hossza $|c(t)| = 1$, fázisa pedig a modulált jel pillanatnyi értéke ($\arg(c(t)) = x(t) \cdot \pi$). Így a jel 1 értéke a π fázishoz kerül. A $c(t)$ jelet az Euler alakkal könnyen előállíthatjuk: $c(t) = e^{j \cdot \pi \cdot x(t)}$.



9. ábra. Fázisok előállítása a moduláló jeltől

Az IQ keveréshez ennek a komplex jelnek a valós részét az in-phase $\cos(2\pi \cdot f_o \cdot t)$, a képzetes részét a quadrature $-\sin(2\pi \cdot f_o \cdot t)$ oszcillátor jellel moduláljuk (szorozzuk), majd összeadjuk. Kész is a modulált jel!



10. ábra. Az előállított fázismodulált jel

Kód 3. Fázismoduláció IQ modulátorral

```
1 Fs = 44100; % mintaveteli freki
2 l = 1.5; % minta hossza masodpercben
3 N = l * Fs; % mintak szama
4 t = ( 1:N )' ./ Fs; % Ido oszlopvektora
5
6 f = 5; % eredeti jel frek
7 x = sin( 2 * pi * f * t ) .* 0.5; % ez lesz az atvinni kivant jel (1/2
    amplitudo)
8
9 c = ones(N,1); % N meretu vektor létrehozasa es feltoltese 1-essel
10 for i=1:N
11     c(i) = exp( j * pi * x(i) ); % fazismodulalt jel létrehozasa
12     % e^(2pi*j*x/2), mert nem szeretnenk a korbeforgassal atlapolodni
13 end
14
15 fo = 3000; % oszcillator frek
16 oi = cos( 2 * pi * fo * t ); % in-phase oszcillator
17 oq = -sin( 2 * pi * fo * t ); % quadrature oszcillator
18
19 y = oq .* imag(c) + oi .* real(c); % IQ keveres
20
21 figure(1);
22 plot(t, y);
23 figure(2);
24 wf(y, Fs);
```

Feladat: Ez alapján készítsünk el egy FM modulátort! Nagyon hasonlít az előzőhöz, csak itt a komplex vektor forgásának frekvenciája változik.

4.4. Újramintavételezés

Próbáljuk ki hangfileokkal! Ügyeljünk arra, hogy a FM sávszélessége nagy lesz, ezért nagyobb helyre lesz szükségünk a frekvenciatartományban. Ha megváltoztatjuk a mintavételi frekvenciát és a hangot újramintavételezzük ez alapján, ki tudjuk terjeszteni a frekvenciatartomány méretét.

Vigyázzunk, hogy n -szeres mintavételi frekvenciánál a minták száma is n -szeresére változik, és ez nagyon megnövelheti a számítási időt.

Az újramintavételezés a $resample(x, p, q)$ függvénnyel tehető meg, ami $\frac{p}{q}$ -szorosára változtatja a mintavételi frekvenciát (ehhez be kell tölteni a signal csomagot). Ne felejtsük a F_s változót is beállítani!

Kód 4. Újramintavételezés

```
1 [x,Fs] = audioread('voice.wav');
2 % Ujramintavetelezes -> nagyobb frekitartomany
3 Fs = Fs * 4;
4 x = resample(x, 4, 1);
5 N = length( x ); l = N / Fs;
```