Rádióamatőr tanfolyamot segítő jegyzet
Szabó Áron HA1FLX, Mérő László HA1CNT, Bazsó Márton HA7BM,
Baráth László HA7DN
Retzler András HA7ILM szoftverrádiós vevője alapján

Szoftverrádió gyakorlat

Tartalom

1. Bevezető

A gyakorlat célja a szoftverrádió lehetőségeinek megismerése, és bemutatása egy C nyelven megírt FM demodulátoron keresztül.

2. Előkészületek

A gyakorlathoz Linux operációs rendszerre lesz szükség.

Szükséges csomagok:

- netcat-bsd (a sima verzió nem támogatja a -4 kapcsolót)
- rtl-sdr *(saját Realtek alapú szoftverrádiós eszköz esetén)*
- tcc
- sox
- aplay
- git *(A kezdőkészlet letöltéséhez)*
- *ízlés szerinti szövegszerkesztő program*

Fontos, hogy működjön az internetelérés (vagy saját hardver esetén annak a drivere), illetve legyen működő hangkártya kimenet, és hangszóró/fülhallgató.

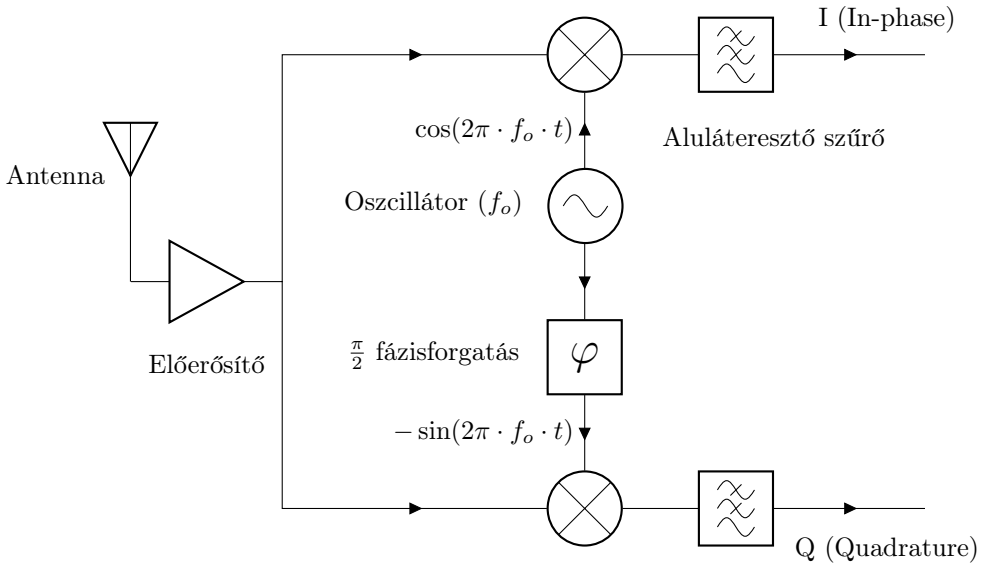
3. Elméleti összefoglaló

3.1. IQ demodulátor

Az IQ demodulátor egy szoftverrádiós eszközöben gyakran használt demodulátor. Ilyen hardver található a gyakorlaton használt RTL-SDR USB stickben is. Az IQ demodulátorban a bejövő jelet két azonos frekvenciájú, de 90°-kal eltolt jellel keverjük le (3.1. ábra). A felső keverési termékekeket egy low-pass filterrel kiszűrjük, íg kapjuk az I (in phase) és a Q (quadrature) jeleket.

Ezen jeleknek a frekvenciája a helyi oszicillátor és a vett jel frekvenciájának különbsége lesz. Az I és Q jelet lehet külön-külön digitalizálni, ezen a ponton már nem szükséges nagyon gyors analóg-digitális átalakítás, hiszen alacsony frekvenciájú jelekkel dolgozunk. Az I és Q jelet együtt, mint $I + jQ$

komplex jelet értelmezzük. Ez már digitálisan feldolgozható, elvégezhető a demoduláció.



1. ábra. Az IQ demodulátor egyszerűsített blokkvázlata

3.2. Fazorok

Egy adott frekvenciás szinuszos jel egyértelműen megfeleltethető egy komplex számmal, a *fazorával*, amelynek fázisa és amplitúdója megegyezik a jel fázisával és amplitúdójával.

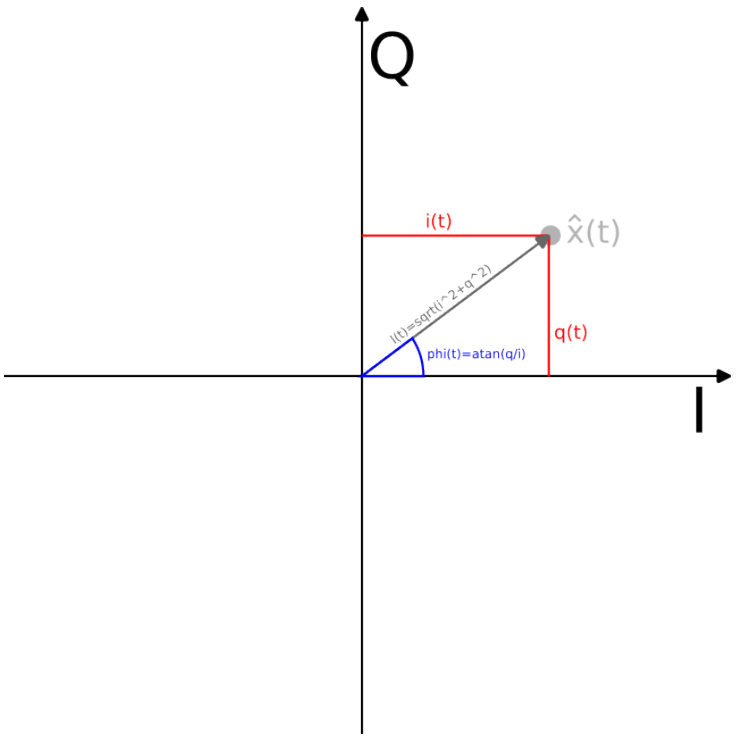
$$x(t) = |A| \cdot \cos\left(2\pi f \cdot t + \phi_0\right) \Longleftrightarrow \hat{X}$$

Ahol

- $\left|\hat{X}\right| = A$
- $\arg \hat{X} = \phi_0$

Megjegyzés: ha megengedjük hogy A negatív legyen, ami a gyakorlatban előfordulhat, picit komplexebb lesz a feltétel.

A 3 és a 4 ábrákon bemutatásra kerül néhány fazor-jel párosítás, amelyek amplitúdóban vagy fázisban különböznek. Látható, hogy két-két jel megegyezik a két ábra között - a 180°-os forgatás egyaránt felfogható negatív amplitúdóként is.



2. ábra. (Időfüggő)fazor Descartes és polárkoordinátás reprezentációi

A komplex szám felírható egyaránt polárkoordinátás és Descartes-alakban (2 ábra). Utóbbiban a valós részt I-vel, mint *in-phase*, a képzetes részt Q-val mint *quadrature* jelöljük.

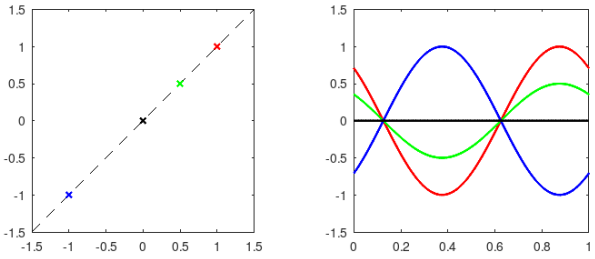
$$\begin{aligned} \hat{X} &= |A| \cdot \mathrm{e}^{j\phi_0} \\ \hat{X} &= I + jQ \\ I &= A\cos\phi_0 \\ Q &= A\sin\phi_0 \end{aligned}$$

Ha a jel paraméterei időben változnak (például modulációból adódóan), felírható a fazor-idő, illetve a komponensek $i(t), q(t), A(t)$ és $\phi(t)$ függvényei.

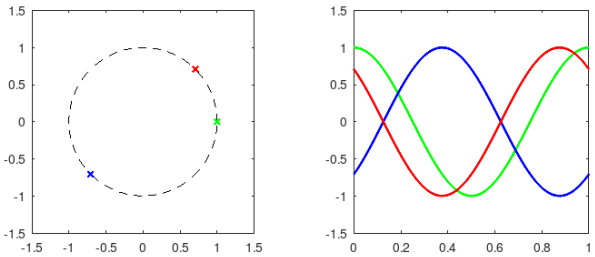
A fentebb vázolt IQ demodulátor valójában a jel fazorát méri meg és digitalizálja számunkra (mint I és Q jelek), így a mi feladatunk az információ visszaállítása a jel fazorából.

3.3. Komplex jelek és a frekvenciahiba hatása

Az így kapott komplex jelek szoftveres feldolgozása gyakran könnyebb, mint a csak valós számokból állóké. Az eredeti vett jel szinuszos komponensei itt



3. ábra. Azonos fázisú, de különböző amplitúdójú (amelyből egyik negatív) fazorok és hozzá tartozó jelek



4. ábra. Azonos amplitúdójú, de különböző fázisú fazorok és hozzá tartozó jelek

komplex szinuszoidként jelennek meg.

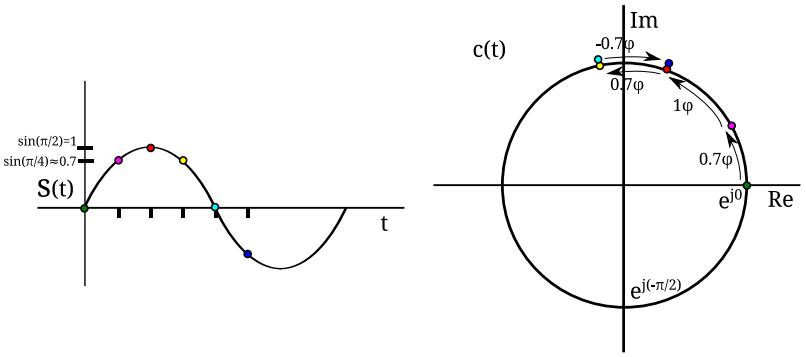
A komplex szinuszoidok a következő alakot veszik fel:

$$Ae^{j(\phi+2\pi ft)} = A(\cos(\phi + 2\pi ft) + j\sin(\phi + 2\pi ft))$$

Az IQ vektorokat ábrázolva ezeket A hosszúságú f frekvenciával az origó körül körbeforgó vektorokként fogjuk látni. A arányos a jel amplitudójával, f pedig a helyi oszcillátor és a vett komponens frekvenciája közötti eltérés.

Amennyiben a vizsgált komponens frekvenciája nagyobb volt, mint a helyi oszcillátoré, f pozitív lesz, tehát az IQ vektor az óramutató járásával ellentétesen forog. Ha a vizsgált komponens frekvenciája kisebb volt, mint a helyi oszcillátoré, f negatív lesz, tehát az IQ vektor az óramutató járásával megegyezően forog.

Ha az LO és a vett komponens frekvenciája megegyezik, $f = 0$ lesz, így a demodulált IQ vektor konstans. Ilyenkor ϕ az LO és a vett jel közötti fáziseltérés. (Általánosságban a $\phi + 2\pi ft$ kifejezés értelmezhető a két jel fáziseltéréseként, azonban ez időben változó, ha azok nem azonos frekvenciájúak.)



5. ábra. Komplex forgó vektor szinuszos jel esetén

3.4. Frekvencia moduláció

Frekvencia modulációnál egy vivő jel frekvenciájába kódoljuk az $x_m(t)$ moduláló jel által hordozott információt. Az $y(t)$ modulált jelünket a következő képpen számolhatjuk ki:

$$y(t) = A\cos(2\pi t(f_c + f_\Delta x(t)))$$

Itt A az amplitúdónk, f_c a középfrekvenciánk, ez a "vivő eredeti frekvenciája", illetve az állomás névleges frekvenciája (pl. 97.1 MHz). f_Δ a frekvencialöket, ± 1 közé normált moduláló jelnél az kimeneti jel energiájának nagy része $f_c \pm f_\Delta$ közé esik.

4. Feladatok

4.1. FM demoduláció

A feladathoz egy RTL-SDR hardvert fogunk felhasználni. Az USB stick a digitalizált IQ jelet egy bytestreamként küldi: egy 1 byte-os I majd egy 1 byte-os Q mintát küld felváltva. A 8 bites minták offszettel ábrázolják a negatív értékeket (így például a jel 0 értékét valójában a 127-es bináris érték jelenti). Ez az ábrázolási mód NEM egyezik meg az elterjedt kettes komplemens számábrázolással. A bytestreamet a demoduláló programunk a standard inputon fogja megkapni, majd a standard outra kell, hogy kiadja az abból kiszámolt hangmintát.

Segítségképpen a minta beolvasás, illetve kiírás már szerepel a kiinduló kódban. A fogadott IQ jel magasabb mintavételi rátával fog jönni, mint amit a hangkártya kezelni tud, azonban ezt egy külső programmal a demoduláció után fogjuk csökkenteni. Ezért minden fogadott IQ mintára pontosan egy hangmintát kell a programnak kiadni.

Lista 1. Kiinduló kód

```
#include <math.h>
#include <stdio.h>

int main()
{
  double i, q, s;
  for(;;) //vegtelen ciklus
  {
                // beolvassuk az I mintat, az offszetet levonjuk, hogy a 0 tenyleg 0 legyen
    i=((unsigned char)getchar()-127);
                // Q-val ugyan ez
    q=((unsigned char)getchar()-127); //bolvassuk

                // s-be kell kiszamolni a demodulalt hangot
                // a kiiras miatt s-t ugy kell skalazni hogy kb. -127 es 128 koze essen.
    // s = ??;

                // s-t visszaalakitjuk offszetese majd kiirjuk
    putchar((unsigned char)(s+127));
  }
}
```

A programot saját hardverrel a következő parancssorral lehet futtatni:

```
rtl_sdr -s 300000 -f 103300000 -g 20 - | tcc -lm -run minidemod-wfm.c |↩
    sox -t raw -r 300000 -e unsigned -b 8 -c 1 - -t raw - rate 48000 ↩
    sinc -15k | aplay -f U8 -c1 -r 48000 --buffer-size=200000
```

Ha nincs saját hardvered, használhatod a Kafu `rtl_mus` szerverét ezzel a paranccsal:

```
nc teto.sch.bme.hu 7373 -4 | tcc -lm -run minidemod-wfm.c | sox -t raw -↩
    r 300000 -e unsigned -b 8 -c 1 - -t raw - rate 48000 sinc -15k | ↩
    aplay -f U8 -c1 -r 48000 --buffer-size=200000
```

A gyakorlaton a parancsokat nem kell használni, azok előre meg vannak írva a `parancs.sh` fileban.

Mindkét parancssorban az egyes programokat a `|` (pipe) karakter választja el, ami azt jelzi a shellnek, hogy a balra lévő program standard outját adja át a jobbra lévő program standard injére.

Az `rtl_sdr` program az USB stickről olvas mintákat, amiket a standard outra ír. A -s kapcsoló a mintavételi rátát, a -f a helyi oszcillátor frekvenciáját, a -g pedig az erősítést állítja. Az `nc` (netcat) egy TCP kapcsolatot hoz létre a megadott szerverrel a megadott porton, és az onnan jövő adatot a standard outra írja. A megadott gépen és porton egy `rtl_mus` nevű program fut, ami a szerverbe dugott RTL-SDRből jövő mintákat fogja továbbítani.

A `tcc` lefordítja a C kódunkat, és egyből futtatja is az elkészült programot. A `sox` az újramintavételezést végzi el, a 240 kHz-s jelből 48 kHz-eset csinál. Az `aplay` pedig kijátsza a kapott mintákat a hangkártyán.

A gyakorlat kiinduló anyagai, illetve megoldásai elérhetőek a címen.