

Radionica: softverski definition radio

Dimitrije Erdeljan

Matematička* gimnazija

28. 04. 2021.

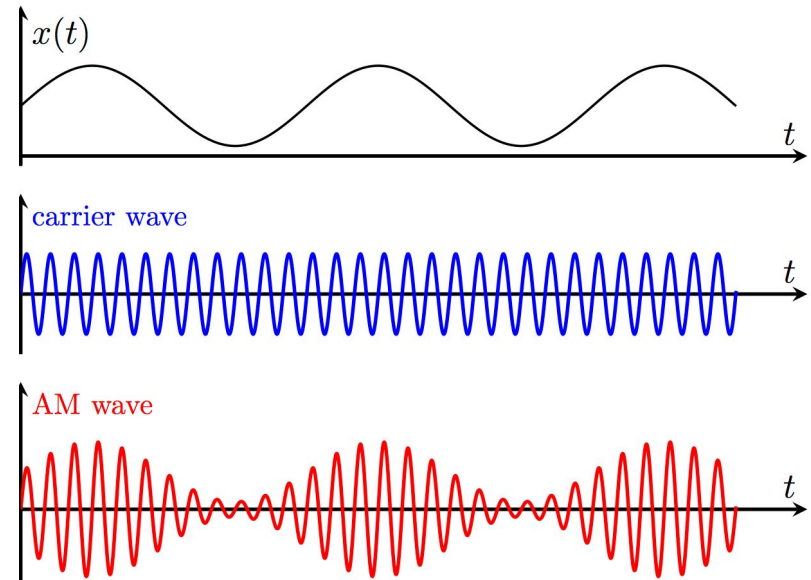
1. Uvod u softverski definisan radio
2. Radionica: teorija
3. Praktični detalji

- Tradicionalno: radio-prijemnik i obrada signala “u elektronici”
 - Novi dizajn za svaki protokol
 - Jednostavno i jeftino, ali nema fleksibilnosti
 - I dalje korisno za masovnu proizvodnju
- SDR: što manje obrade u hardveru; digitalizovan signal obrađuje računar
 - Fleksibilno, odlično za prototipe i istraživanje
 - Skuplji hardver od jednonamenskog

- Radio: komunikacija putem elektromagnetnih talasa
 - Promena napona na anteni predajnika → EM talas
 - EM talas → napon na anteni prijemnika
- Antena “vidi” zbir* svih signala koji stižu do nje – kako ih razdvojiti?
- Većina signala sadrži mali opseg frekvencija, tako da se možemo dogovoriti da se ne preklapaju
 - FM radio: 88-108 MHz, svaka stanica ima svoju frekvenciju
 - Mobilna telefonija: 806-947 MHz
 - ...

- Jedan od najjednostavnijih protokola
- Podaci koje emitujemo: $A(t)$
 - Za radio: amplituda zvuka
- **Amplitudna** modulacija: podatke prenosimo kao amplitudu nosećeg talasa – sinusoide $\cos(2\pi f t)$

$$s(t) = A(t) \cos(2\pi f t)$$



Petar Veličković: <https://github.com/PetarV-/TikZ/>

- Demodulacija: od $s(t) = A(t) \cos(2\pi f t)$ do $A(t)$
- Naivno rešenje: izračunamo $|s(t)|$ i eliminišemo visokofrekventnu komponentu filtriranjem
 - ... šta sa ostalim signalima?
- Prvo filter koji propušta samo frekvencije blizu f ?
 - ... radi, ali nam treba novi filter za svaku stanicu...

- Šta ako “pomerimo” noseću frekvenciju signala?
- Korisno je posmatrati kompleksne sinusoide umesto realnih:
 - $\cos(x) = \frac{1}{2} (e^x + e^{-x})$

$$s(t) = \frac{1}{2} A(t) (e^{2\pi i f t} + e^{-2\pi i f t})$$

- Pomnožimo ovo sa $e^{-2\pi i f t}$:

$$2 s(t) e^{-2\pi i f t} = A(t) + A(t) e^{-4\pi i f t}$$

- Sada možemo eliminisati drugi član filtrom koji propušta samo niske frekvencije (i ne zavisi od f!)



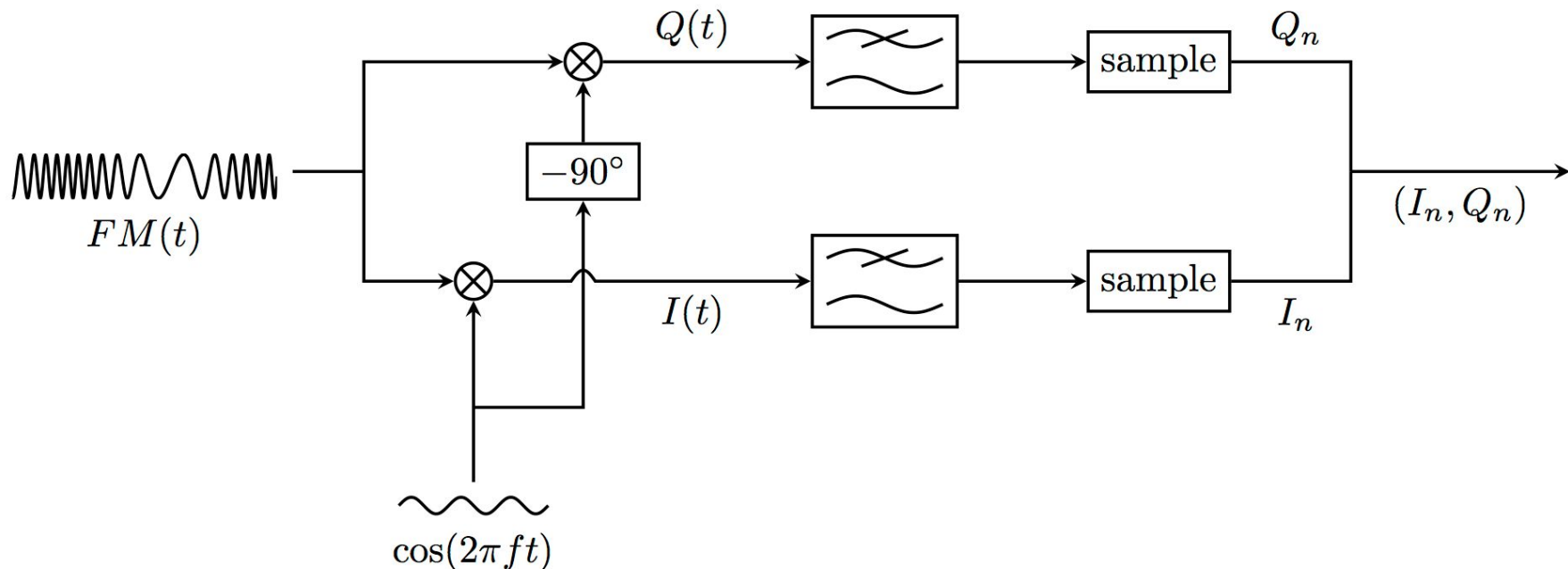
- Istu ideju možemo koristiti i drugde: množenje sa $e^{2\pi i f t}$ efektivno pomera sve frekvencije u signalu za f
- Ako nas interesuje signal koji se nalazi oko neke frekvencije f , možemo pomeriti za $-f$ i filtrom eliminisati ostalo
- SDR prijemnici ovo rade u hardveru

- Uzmemo antenu, izmerimo napon na njoj puno puta u sekundi i pošaljemo to računaru...?
- Koliko puta (“sampling rate”)?
 - Nyquist: barem $2f$, gde je f najveća frekvencija u signalu
 - Da bi imali razuman sampling rate: pomeranje i filtriranje (u hardveru) pre merenja!

Analogno u digitalno (2)

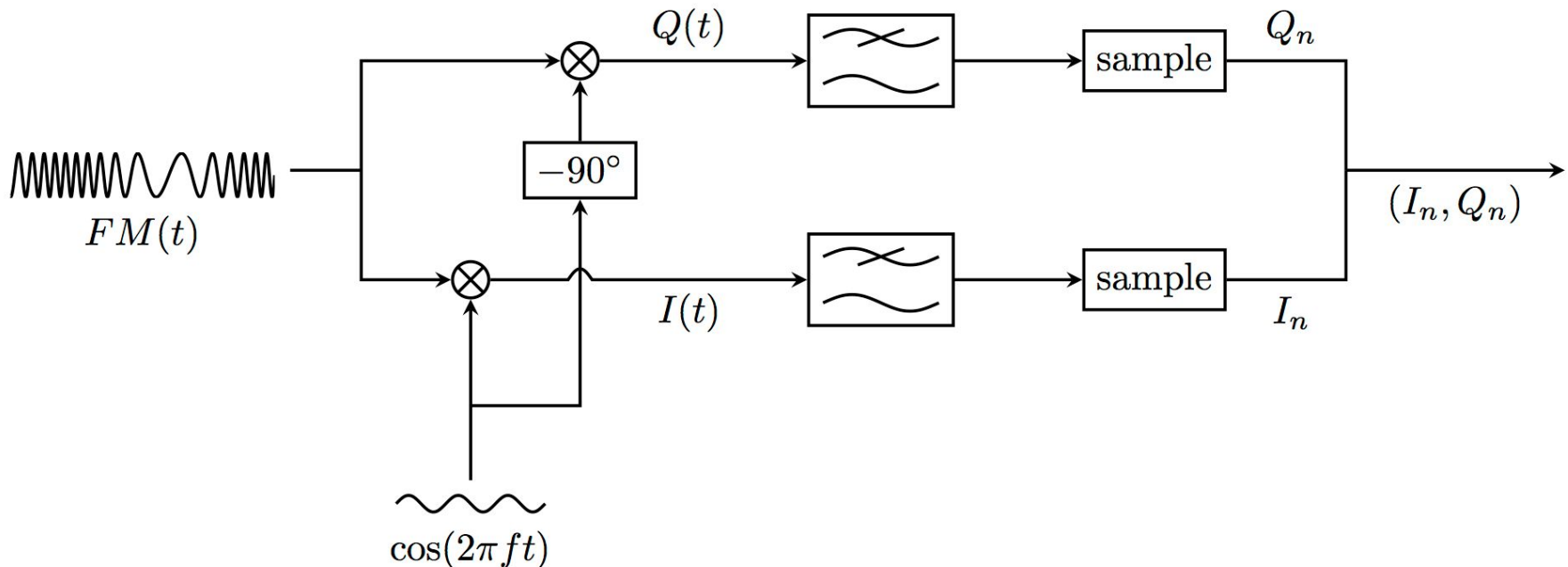


- Inženjersko pitanje: kako pomnožiti (realan) napon sa kompleksnim brojem?
- $e^{2\pi i f t} = \cos(2\pi f t) + i \sin(2\pi f t)$
- Dve žice: jedna za realan, a druga za imaginaran deo



Petar Veličković: <https://github.com/PetarV-/TikZ/>

- Prijemnik konfiguriramo sa frekvencijom f koja nas interesuje i sampling frekvencijom f_s
- Hardver: pomeri spektar za $-f$ i filtrira tako da je najveća frekvencija $< f_s/2$
- Hardver: izmeri napon f_s puta u sekundi i pošalje računaru
- Softver: sve ostalo!



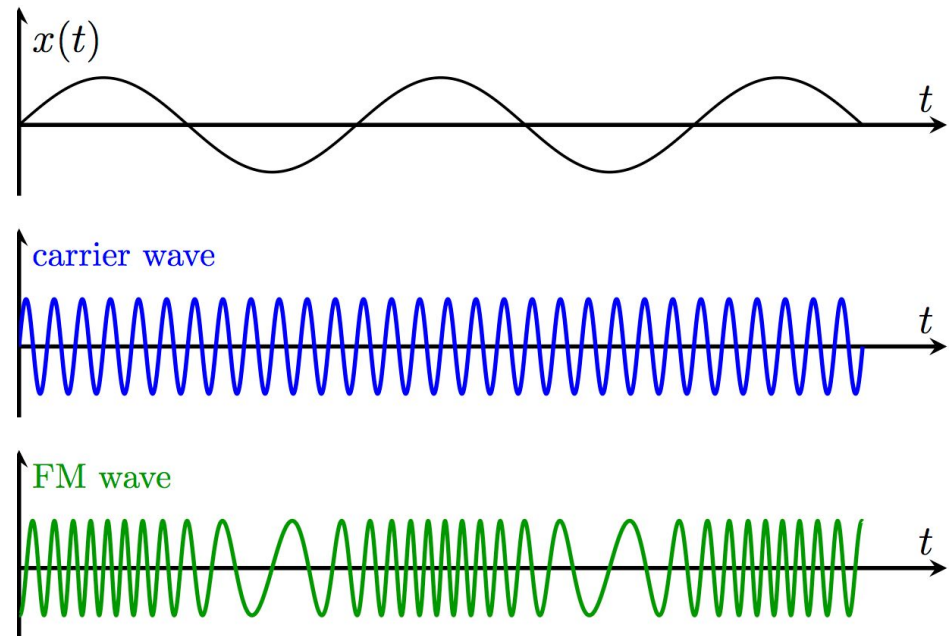
Petar Veličković: <https://github.com/PetarV-/TikZ/>

1. Uvod u softverski definisan radio
2. Radionica: teorija
3. Praktični detalji

- Implementiraćemo demodulator za FM radio: malo složeniji nego AM, ali u široj upotrebi
- Plan:
 - Kratak uvod u FM
 - Implementacija (ulaz / izlaz već rešeni)
 - Veštački signal
 - Malo realniji veštački signal
 - Pravi snimak
 - Radio uživo?

- Noseći talas $\cos(2\pi f t)$ kao i za AM
- **Frekventna** modulacija:
amplituda je stalna, podatke prenosimo promenom frekvencije
 - $f(t) = f + \Delta_f A(t)$

$$s(t) = \cos(2\pi f t + 2\pi \Delta_f \int A(t) dt)$$



Petar Veličković: <https://github.com/PetarV-/TikZ/>

- Početni signal:
 - $s(t) = \cos(2\pi i f t + 2\pi i \Delta_f \int A(t) dt)$
- Pomnožimo sa $e^{-2\pi i f t}$ da bi pomerili noseću frekvenciju na 0 i filtriramo:
 - $e^{2\pi i \Delta_f \int A(t) dt}$
- Uzmemo fazu (ugao rotacije):
 - $2\pi \Delta_f \int A(t) dt$
- Izračunamo izvod:
 - $2\pi \Delta_f A(t)$
- ... i filtriramo da bi smanjili šum

- Kako eliminisati visokofrekventne (“brze”) komponente signala?
- Naivno rešenje: uprosečimo nekoliko elemenata...
 - $S'_n = S_n + S_{n-1} + \dots + S_{n-k}$
 - Radi, ali će izobličiti i niske frekvencije
- Možemo odabrati koeficijente tako da su frekvencije ispod f_0 (skoro) nepromenjene, a one iznad skoro eliminisane:
 - $S'_n = b_0 S_n + b_1 S_{n-1} + \dots + b_m S_{n-m}$
- Izbor koeficijenata b_i : van opsega ove radionice :)
 - Matlab: `fir1(m, 2 * f0 / fs)`
 - Julia: `digitalfilter(Lowpass(f0/fs), FIRWindow(hanning(m+1)))`
- Nekoliko već dizajniranih filtera u materijalima

- Radio snimci sa kojima ćemo raditi sadrže 1M vrednosti u sekundi
- Najviša frekvencija koju ljudi čuju je ~20 kHz – nema potrebe za ovoliko vrednosti
- Možemo smanjiti broj odbiraka u signalu tako što sačuvamo svaki k-ti
 - Pažljivo: ovo je OK samo ako je najveća frekvencija u signalu $f_s/2k$
- Sampling frekvencije koje nisu f_s / k : moguće ali komplikovanije (interpolacija...)
 - Matlab, Julia: `resample()`

- Kompleksan broj → faza?
 - C++: `x.arg()`
 - Python: `cmath.phase(x)`
 - Matlab / Julia: `angle(x)`
- Izvod?
 - Dovoljno dobro numeričko rešenje: razlika susednih elemenata
 - Pažljivo sa granicom $2\pi \rightarrow 0$
 - Matlab: `unwrap()`

1. Uvod u softverski definisan radio
2. Radionica: teorija
3. Praktični detalji

- Fajlovi sa SDR snimcima: `input/ime-xyzM-1M.dat`
 - Centralna frekvencija: u imenu
 - Sampling rate: 1 MHz
- Format: niz kompleksnih brojeva, svaki predstavljen kao dva 32-bitna float-a (realan + imaginaran)
- Funkcija `read_input(filename)` učitava fajl i vraća niz kompleksnih brojeva

- Krajnji cilj: audio fajl sa radio prenosom
- Audio fajl je u suštini samo niz amplituda, sa relativno malim brojem vrednosti u sekundi (tradicionalno 44.1 KHz ili 48 KHz)
 - Većina formata očekuje vrednosti u $[-1, 1]$
- Funkcija `audio_write(filename, data, fs)` proizvodi .wav fajl
- Vaš jezik možda ima druge načine da ovo reši – real-time zvuk?
 - Linux: `./program | aplay -r 48000`

Real-time radio?



Ako uspete da demodulirate sve ulaze, javite se da probamo da povežemo vaš program na pravi SDR :)



Hvala na pažnji!

Pitanja?

(materijali na: <https://tinyurl.com/ni-sdr-21>)