



Analýza využití softwarově definovaného rádia v leteckých aplikacích.

Návrh korelátoru a jeho implementace

Autor: Petr BOJDA

Adresa: Morávka 575, okr. Frýdek-Místek

Dne: 10. září 2017

Verze dokumentu: 1.0

1 Úvod

Hlavní cíl zprávy Zpráva vzniká na podnět výzkumného týmu Katedry Leteckých elektrotechnických systémů k podpoře výzkumného úkolu v oblasti terestriálních navigačních systémů s rozprostřeným spektrem. Cílem je nalézt vhodnou strukturu a parametry signálu tak, aby co nejlépe vyhověl požadavkům na celkový systém. V této práci je rozebrán způsob realizace korelátoru a jeho implementace do systému číslicového zpracování signálu přijímače k za účelem získání autokorelačního maxima, resp. měření časového zpoždění.

Použité prostředky Zpráva vznikla jako komentovaný popis a dokumentace návrhu digitálního rádiového systému. Zahrnuje základní kroky návrhu - simulaci, testování (verifikaci) a nakonec i finální implementaci. Cílovou platformou je SDR, tudíž lze předpokládat, že funkce rádiového systému bude popsána, definována a implementována formou bloků software. Veškeré softwarové nástroje a prostředky použité při zpracování této zprávy jsou v kategorii "Open-Source Software License", zejména BSD licence v případě simulačních nástrojů a GNU GPL v případě implementačních nástrojů. Textová část je editována v systému Latex. Jednotlivé práce byly odvedeny na počítačích s operačním systémem Linux, distribuce Ubuntu. Nebyly použity žádné softwarové nástroje podléhající komerční licenci.

K simulaci je využit systém knihoven v jazyce Python, zejména pak knihovny NumPy, Matplotlib a SciPy. Implementace korelátorů byla provedena s využitím systému knihoven Accelerate verze 2.3.1 a Accelerate Cudalib verze 2.0 nainstalované v rámci balíku Anaconda 4.3.24. Tyto nástroje umožnily paralelizaci výpočtů korelačních funkcí v GPU procesorech NVidia a CPU procesorech Intel.

S využitím funkcí těchto knihoven byly vytvořeny soubory funkcí vypočítávajících korelační koeficient v časové i kmitočtové oblasti. Jak bylo uvedeno výše, všechny simulace jsou psány v programovacím jazyce Python.

2 Vyžití korelátoru v navigačních systémech

Tato práce se zabývá obecným digitálním rádiovým systémem. V leteckých aplikacích [Složení systému](#) jsou rádiové systémy použity ve třech základních oblastech: komunikační, navigační a přehledové. Podle toho, jaký je primární účel systému, mění se i jeho složení a nastavení parametrů. Společná je jejich základní struktura:

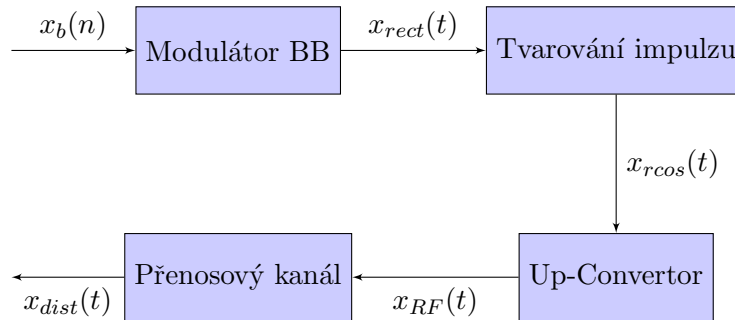
- Vysílač
- Přenosový kanál
- Přijímač

V dalším budou analyzovány tyto tři komponenty oprostěné od funkcí specifických pro dané oblasti použití. To znamená, že nebude zahrnuto například různé kódování přenášené bitové posloupnosti (zdrojové či kanálové) a podobně. Vstupem vysílače je logický signál. Ten je modulován některou z mnoha digitálních modulací na nosný harmonický signál a anténou vyslán do prostoru. Přenosový kanál reprezentuje souhrnně vliv prostředí (přenosového média) na přenášený signál. Jako příklad je možno jmenovat: přidaný šum, útlum, vícecestné šíření, dopplerův posun, interference, omezení šířky pásma přenášeného signálu a jiné. Zkreslený signál je poté detekován a zpracován pomocí přijímače.

2.1 Vysílač

Struktura vysílače, jak bude v následném textu chápána a analyzována je na obr. 1. [Popis struktury vysílače](#)
Vstupní signál $x_b(n)$ je bitovou posloupností nesoucí požadovanou informaci. Blok *Modulátor BB* přemění – moduluje vstupní bitovou posloupnost na komplexní signál v základním pásmu (angl. *baseband*). Signál v základním pásmu $x_{rect}(t)$ pomocí aktuální úrovně své reálné a imaginární složky precizně definuje amplitudu a fázi nosné složky budoucího výstupního signálu vysílače $x_{RF}(t)$. To je již signál s harmonickou nosnou, na kterou vybranou digitální modulací namodulována bitová posloupnost $x_b(n)$.

SDR (ať už simulované, nebo reálně využívané v této zprávě) používá k transformaci [Baseband modulace – signál v základním pásmu](#) (konverzi) ze základního kmitočtového pásma do pásma nosného kmitočtu (angl. *passband*) obvod kmitočtové přeměny (tzv. směšovač) *Up-Convertor* typu kvadraturní modulátor.



Obrázek 1: Základní struktura digitálního rádiového vysílače.

Tento typ obvodu vyžaduje komplexní vstupní signál v základním pásmu. Komplexní signál v základním pásmu je tvořen reálnou (soufázní, angl. *In-Phase*) a imaginární (kvadrurní, angl. *Quadrature*) složkou¹. Blok *Modulátor BB* tvaruje ze vstupní posloupnosti $x_b(n)$ komplexní $x_{rect}(t)$ signál, který je charakterizovaný svou tzv. bitovou (symbolovou) rychlostí (angl. *bitrate*)² a amplitudou. „*Modulátor BB*“ je již navržen konkrétně pro daný typ výsledné digitální modulace signálu na výstupu vysílače $x_{RF}(t)$. Přicházející bity vstupní posloupnosti $x_b(n)$ jsou nejprve seskupovány podle typu výsledné modulace, viz.:

- dvoustavová: 1 symbol signálu $x_{rect}(t)$ odpovídá 1 bitu posloupnosti $x_b(n)$,
- čtyřstavová: 1 symbol signálu $x_{rect}(t)$ odpovídá 2 bitům posloupnosti $x_b(n)$,
- osmistavová: 1 symbol signálu $x_{rect}(t)$ odpovídá 3 bitům posloupnosti $x_b(n)$, ...

Poté jsou seskupené bity tvořící *m-tice* (angl. *tuples*) mapovány na patřičnou reprezentaci pomocí komplexního signálu $x_{rect}(t)$. Pro každou kombinaci bitů symbolu je definována i kombinace úrovní³ soufázní a kvadrurní složky signálu $x_{rect}(t)$. Jedním ze základních parametrů digitálních rádiových systémů je i požadovaná přenosová *bitová* (angl. *bitrate*) a z ní odvozená *symbolová* (angl. *symbol-rate*) rychlost. Symbolová rychlost definuje dobu trvání jednoho symbolu (impulzu) T_S .

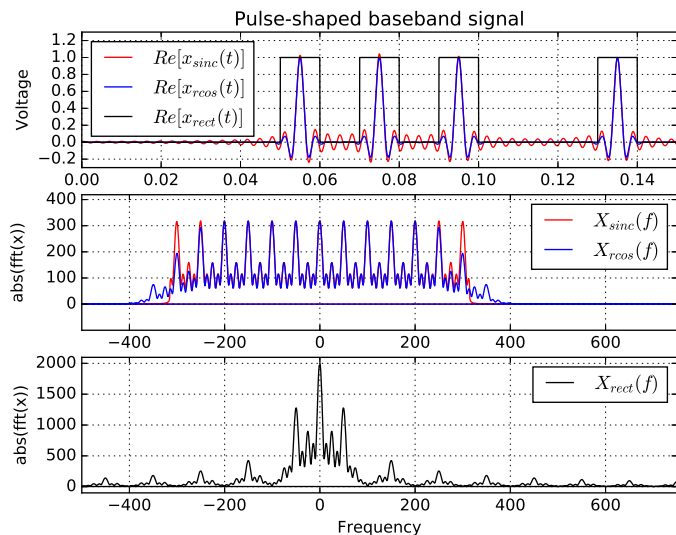
Tvarování pulzu

Za blokem *Modulátoru BB* jsou přechody mezi jednotlivými symboly signálu $x_{rect}(t)$ ⁴

¹Soufázní složka signálu bývá v technické praxi často označována jako signál I, kvadrurní pak jako Q.

²V případě *m-stavové* modulace je poměr bitové rychlosti k rychlosti symbolové dán počtem bitů přenesených v jednom symbolu.

³kvantizační úroveň může být dána přímo napětím, nebo vyjádřena jen číselně. To závisí na aktuální podobě obvodu Up-konvertoru. Někdy je vyžadován vstupní signál v základním pásmu, tzv. I a Q v analogové podobě. Moderní integrované obvody provádějící konverzi však často obsahují na vstupu dva DA převodníky a signál v základním pásmu je na ně přiváděn ve formě diskretních I a Q vzorků.



Obrázek 2: Příklad tvarování impulsu signálu v základním pásmu. Kvůli jednoduchosti je znázorněná pouze jedna (uvažujme například, že reálná - symfázní) složka komplexního signálu.

ideálně strmé, viz obr. 2. Signál v základním pásmu (*baseband* signál) má v tuto chvíli podobu ideálních obdélníkových impulsů. Obdélníkový průběh je v reálných podmínkách nevýhodný. Skutečné přenosové kanály značně omezují kmitočtové spektrum přenášených signálů a potlačují některé kmitočty. Zjednodušeně lze jejich chování modelovat filtry typu dolní nebo pásmová propust⁵. Pro soudobé systémy pracující s vysokými bitovými (symbolovými) rychlostmi by nepredikovatelné zkreslení impulsu jednotlivých symbolů představovalo vážný problém při detekci a následném zpracování signálu v přijímači. Proto blok *Tvarování impulsu* kontrolovaně omezí pásmo signálu $x_{rect}(t)$ tím, že tvaruje jednotlivé impulsy symbolů podle vhodných matematických funkcí. Teoreticky je jich používáno více (*gauss*, *sinc*, ...), prakticky jsou používány funkce *raised-cosine* (ve zkratce *rcos*). V podstatě se jedná o "utlumenou" funkci *sinc* (angl. *damped sinc*). Výstupem bloku "Tvarování impulsu" je tedy signál $x_{rcos}(t)$, jehož kmitočtové spektrum je na rozdíl od signálu $x_{rect}(t)$ ohraničené a definované už v době návrhu systému. Signál $x_{rect}(t)$ s ideálně obdélníkovým průběhem, signál $x_{rcos}(t)$, kde je impuls tvarován funkcí *raised-cosine* a nakonec i signál

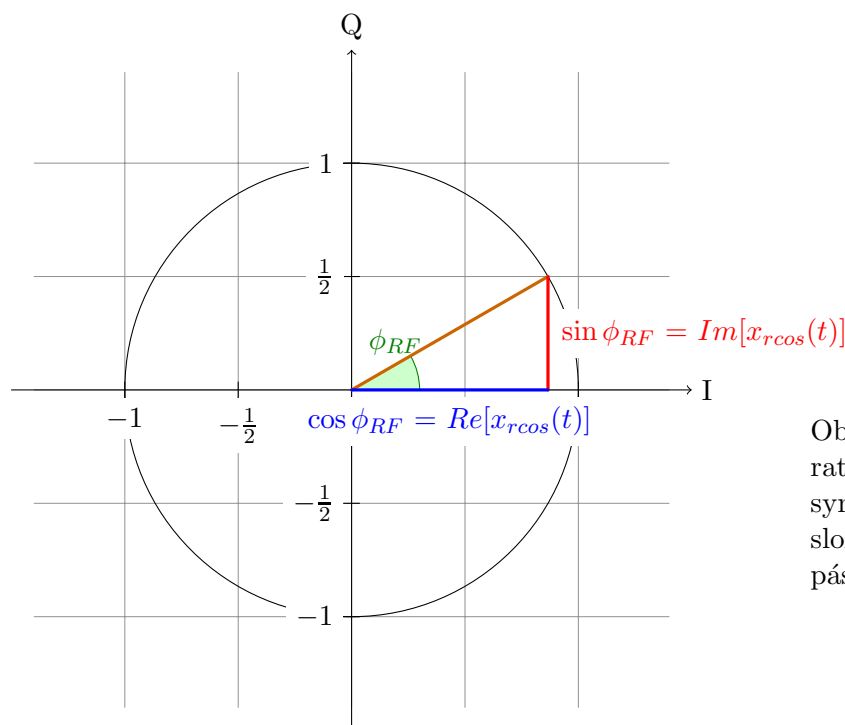
⁴U komplexního signálu v základním pásmu se už nehovoří o "bitech", ale o "symbolech". Přestože v tuto chvíli pojem "symbolu" ještě není precizně definován, je kvůli správnosti výkladu použit. Nicméně, v případě dvoustavových modulací obsahuje symbol jeden datový bit. Proto je možné setkat se někdy v technické literatuře s pojmem "bit" i v případě *baseband* signálu.

⁵V angl. psané literatuře jsou označovány jako "*bandlimited channels*"

$x_{\text{sinc}}(t)$ s impulzem ve tvaru funkce sinc spolu se svými kmitočtovými spektry jsou prezentovány na obrázku 2.

Up-Convertor,
Transformace do
pásmu VF

Komplexní signál $x_{\text{rcos}}(t)$ je připraven k transformaci (ke konverzi) do kmitočtového pásma vhodného pro daný systém. Většina SDR dnes využívá obvod *kvadrurní modulátor*. Tento typ modulátoru (často pojmenovávaného anglicky *Up-Convertor*) umožňuje obrovskou flexibilitu celého rádia. Je jedním z hlavních stavebních prvků SDR a příčinou "softwarového" přístupu k tvarování signálu rádia. Jde o to, že nezměněná hardwarová struktura může vygenerovat signál $x_{\text{RF}}(t)$ s takřka libovolnou digitální modulací. Záleží jen na tom, jak bude vypadat komplexní signál v základním pásmu $x_{\text{rcos}}(t)$. Ten, pomocí své reálné složky $\text{Re}[x_{\text{rcos}}(t)]$ a imaginární složky $\text{Im}[x_{\text{rcos}}(t)]$ přesně určuje fázi $\phi_{\text{RF}}(t)$ a amplitudu výsledného signálu $x_{\text{RF}}(t)$. Obrázek 3 demonstruje tento princip. Zde uvažujeme pouze digitální modulace s konstantní amplitudou nosného signálu. U nich je manipulováno jen s fází $\phi_{\text{RF}}(t)$ ⁶ nosného harmonického signálu.



Obrázek 3: Princip kvadrurní modulace. Význam symfázní a kvadrurní složky signálu základního pásma.

⁶Skupina modulací manipulujících s fází (v angl. *PSK*, čili *phase shift keying*)

Díky nízkým kmitočtům signálů základního pásma je možné k jejich tvarování využít procesorů (často obsahujících podpůrné struktury pro číslicové zpracování signálů) a tím pádem lze jejich strukturu i parametry určovat softwarem v procesoru. Přeneseně tedy software procesoru tvaruje (definuje) vysokofrekvenční signál na výstupu vysílače $x_{RF}(t)$.

3 Teorie korelace

Jednou z důležitých rolí korelační funkce při zpracování signálů je měření časových zpoždění mezi přijímanými signály a jejich kopiemi. Toto je pravděpodobně nejčtenější užití korelátorů v rádio-navigačních systémech.

V tomto textu bude rozebrána autokorelační funkce ve smyslu, blíže např. [1]:

$$\begin{aligned} c_{11}(\tau) &= \int_{-\infty}^{\infty} x_1(t) \cdot x_1^*(t - \tau) dt, \\ &= \int_{-\infty}^{\infty} x_1(t + \tau) \cdot x_1^*(t) dt, \end{aligned} \quad (1)$$

kdy je korelován signál se svou vlastní kopií a dále vzájemná korelační funkce podle:

$$\begin{aligned} c_{12}(\tau) &= \int_{-\infty}^{\infty} x_1(t) \cdot x_2^*(t - \tau) dt, \\ &= \int_{-\infty}^{\infty} x_1(t + \tau) \cdot x_2^*(t) dt. \end{aligned} \quad (2)$$

Signály $x_1(t)$ a $x_2(t)$ jsou dva vstupy korelátoru, jejichž indexy 1 a 2 označují dva rozdílné přijímače. Symbol $*$ značí signál komplexně sdružený (z ang. *complex conjugate*). Symbol τ představuje vzájemné časové zpoždění vstupů.

3.1 Popis signálu

VF signály přijaté rádiem”

$$\begin{aligned} s_1(t) &= h(t) \cdot e^{j\phi_{m1}} \cdot e^{j2\pi f_c t} \\ s_2(t) &= h(t) \cdot e^{j\phi_{m2}} \cdot e^{j2\pi f_c t} \end{aligned} \quad (3)$$

Zde $h(t)$ značí obálku impulzu vysílaného, resp. přijatého symbolu. V navigačních systémech je často použito obdélníkového impulzu (GPS). V případě, že chceme lépe využít dostupné kmitočtové spektrum, volíme tvar impulzu $h(t)$ např. raised-cosine, [2].

Při použití některého z typů vécestavové fázové modulace je každý ze symbolů charakterizován fází ϕ_m :

$$\begin{aligned}\phi_{m1} &= 2\pi \frac{m_1 - 1}{M}, \\ \phi_{m2} &= 2\pi \frac{m_2 - 1}{M},\end{aligned}\tag{4}$$

kde m představuje m -tý symbol z množiny M dostupných. Například u dvoustavové BPSK modulace se M rovná 2 a m nabývá hodnot 1 nebo 2. Potom ϕ_m bude nabývat hodnot 0° or 180° pro různá m .

What follows in a signal processing path is an optimal quadrature receiver as defined in [3] with a baseband output

$$\begin{aligned}x_1^I(t) &= h(t) \cdot e^{j\phi_{m1}}, \\ x_1^Q(t) &= h(t) \cdot e^{j\phi_{m1}} \cdot e^{j\frac{\pi}{2}}.\end{aligned}\tag{5}$$

$$\begin{aligned}x_2^I(t) &= h(t) \cdot e^{j\phi_{m2}}, \\ x_2^Q(t) &= h(t) \cdot e^{j\phi_{m2}} \cdot e^{j\frac{\pi}{2}}.\end{aligned}\tag{6}$$

For a further correlation processing

$$\begin{aligned}x_1(t) &= \sqrt{[x_1^I(t)]^2 + [x_1^Q(t)]^2} = h(t) \cdot e^{j\phi_{m1}}, \\ x_2(t) &= \sqrt{[x_2^I(t)]^2 + [x_2^Q(t)]^2} = h(t) \cdot e^{j\phi_{m2}}.\end{aligned}\tag{7}$$

3.2 Korelační funkce

Theoretical signal lasts from $-\infty$ to $+\infty$ and integration limits in Eq. (1, 2) are set accordingly. When a real-world implementation of the function needs to be designed an integration window is set in a more practical way. Let's denote T a length of the integration interval and t_0 a start of the integration:

$$\begin{aligned}
c_{11}(\tau) &= \int_{t_0}^{t_0+T} x_1(t) \cdot x_1^*(t - \tau) dt, \\
&= \int_{t_0}^{t_0+T} x_1(t + \tau) \cdot x_1^*(t) dt
\end{aligned} \tag{8}$$

and cross-correlation function as:

$$\begin{aligned}
c_{12}(\tau) &= \int_{t_0}^{t_0+T} x_1(t) \cdot x_2^*(t - \tau) dt, \\
&= \int_{t_0}^{t_0+T} x_1(t + \tau) \cdot x_2^*(t) dt.
\end{aligned} \tag{9}$$

An auto-correlation coefficient c_{11} as well as a cross-correlation coefficient c_{12} are computed for particular cross-signal time delay τ between input signals x_1 and x_2 . An auto-correlation and cross-correlation functions respectively are then functions of appropriate coefficients with respect to τ .

The question is how to set the integration limit T for a particular signal and expected delays τ between input signals. Obviously, in order to avoid ambiguity, condition

$$T > \tau \tag{10}$$

has to be satisfied. A good practice is to keep $T = 0.1\tau$.

Signals $s_1(t)$ and $s_2(t)$ are modulated by binary codes and/or digital data, depends on the system. As an example might serve a CDMA (Code Division Multiple Access) system or a DS-SS (Direct Sequence – Spread Spectrum), reference [4]. The code repeats periodically, data if any is modulated above the code.

The period of the code T_c needs to be the same as a T . In this case the correlator integrates over an entire code. It is important in case the code was designed with a purpose. Mostly in CDMA systems the code design provides low or none peak in cross-correlation functions when two different codes are correlated one against the other. On the contrary, a significant peak in an autocorrelation function is required to lock the system properly and to resolve the time-reference which is of interest. More details are given in [4]. If T is longer than T_c the autocorrelation function experiences more than one single peak due to the code periodicity and thus the time delay is measured with ambiguity.

4 Generátor rádiového signálu

V digitálním rádiovém systému slouží blok "Up-Converter" k transformaci připraveného [up-convertor](#) signálu v základním pásmu (baseband) do kmitočtového pásma vhodného k rádiovému přenosu. řekněme, že daný krok lze nazvat i "modulací nosné".

V tomto projektu bude vytvořen Up-convertor pro účely simulace. Ve vlastní hardwarové implementaci slouží k přeměně samotné rádio, čili fyzicky RF-frontend.

4.1 Realizace Up-convertoru - Python

Let's start with a time-domain first to derive the differential equation from a *free-body* [Rozbor](#) *diagram*. We will use the second Newton's law, setting the mass $M = 0$.

4.2 Použití bloku USRP v GNU Radio

We are required to derive both natural and step response.

5 Simulace - popis vytvořených funkcí

Veškeré simulace byly vytvořeny v jazyce Python, variantě 3.5.2. s využitím nástrojů knihoven Matplotlib 1.5.3, NumPy 1.11.1 a SciPy 0.18.1, vše instalováno v rámci balíku Anaconda 4.2.0. Simulace byly napsány, vyzkoušeny a provozovány na počítači s operačním systémem Linux Ubuntu 16.04.

Vytvořené funkce simulací jsou sdruženy do několika samostatných souborů s koncovkou `.py` a umístěny v podadresářích uvnitř samostatného adresáře `numpy`. Členění do podadresářů je podle logické a funkční příslušnosti. Dále hlavní adresář `numpy` obsahuje řadu příkladů použití simulačních funkcí.

Funkce ze souborů ve vnořených adresářích lze využít až poté, co jsou do skriptu hlavní úrovně importovány⁷.

5.1 Generátory signálu

Funkce zde popsané jsou umístěny v podadresáři `siggens`. Patří k základním prostředkům modulací – generují signál požadovaného tvaru a parametrů a to jak v podobě základní (jediný, neopakující se impuls), tak i ve variantě opakujících se impulsů obsahujících kód.

5.1.1 Jednoduché impulsy

Funkce, které byly vytvořeny proto, aby generovaly jediný a neopakující se impuls jsou sdruženy v souboru `one_pulse.py`. Mohou sloužit jednak k prezentaci impulsu daného tvaru a jeho vlastností, zároveň jsou ale základem pro funkce, které spojují jednotlivé impulsy do jejich posloupností.

Funkce `rect_p` - obdélníkový impuls Funkce, viz. listing 1, generuje jednoduchý obdélníkový impuls. Vstupními parametry jsou t_{START} a t_{END} , tedy okamžik začátku a konce impulsu v rámci časové osy t . Výstupem funkce je vektor p vzorků impulsu odpovídajících vzorkovacím okamžikům definovaných vstupním vektorem t .

Listing 1: Generátor jednorázového obdélníkového impulsu

```
1 | def rect_p( t, t_start, t_end ):
```

⁷Toto je pochopitelně dokumentovanou vlastností jazyka Python. Zde je tento fakt zmíněn pouze jako připomínka.

```

2 | p = ((t>t_start) & (t<t_end)) *1
3 | return (p)

```

Funkce sinc_p - sinc(x) impuls Funkce, viz. listing 2, generuje jednoduchý sinc impuls. Vstupními parametry jsou t_0 a T_B , tedy střed impulsu v rámci časové osy t a šířka hlavního laloku T_B . Výstupem funkce je vektor p vzorků impulsu odpovídajících vzorkovacím okmžikům definovaných vstupním vektorem t .

Listing 2: Generátor jednorázového impulsu tvaru $\frac{\sin(x)}{x}$

```

1 | def sinc_p( t,t0,Tb ):
2 |     p = np.sinc(np.pi*(t - t0)/Tb)
3 |     return (p)

```

Funkce rcos_p - Raised-cosine impuls Funkce, viz. listing 3, generuje jednoduchý raised-cosine impuls. Vstupními parametry jsou t_0 a T_B , tedy střed impulsu v rámci časové osy t a šířka hlavního laloku T_B . Výstupem funkce je vektor p vzorků impulsu odpovídajících vzorkovacím okmžikům definovaných vstupním vektorem t .

Listing 3: Generátor jednorázového impulsu tvaru raised-cosine

```

1 | def rcos_p( t,t0,Tb,alpha ):
2 |     beta = alpha / Tb
3 |     damp = np.cos(np.pi*beta*(t - t0)) / (1 - 4*(beta*(t - t0))**2)
4 |     p = np.sinc(np.pi*(t - t0)/Tb) * damp
5 |     return (p)

```

5.1.2 Posloupnosti impulsů

Funkce, které byly vytvořeny proto, aby generovaly posloupnost impulsů různého tvaru jsou sdruženy v souboru `train_pulse.py`. Jsou napsány tak, aby jednak generovaly impuls požadovaného tvaru několikrát opakovaně v průběhu časové osy dané vektorem t . Dále mohou zahrnout i kódování - impuls je nebo není na daném časovém úseku přítomen, podle toho, je-li příslušný bit kódové posloupnosti roven logické jedničce nebo nule.

Funkce rect_tr - obdélníkový impuls Funkce, viz. listing 4, generuje posloupnost obdélníkových impulsů. Vstupními parametry jsou délka impulsu t_p , délka mezery mezi

impulzy t_s , a zpoždění celé posloupnosti – mezera mezi začátkem časové osy t a začátkem prvního impulsu. Dále je vstupem vektor logických hodnot $code$. Ten určuje jednak kódovou posloupnost – tzn. přítomnost nebo nepřítomnost daného impulsu v rámci celé řady impulsů. Dále počet bitů posloupnosti $code$ určuje počet impulsů výsledného signálu.

Výstupem funkce je vektor x vzorků impulsů odpovídajících vzorkovacím okmžikům definovaných vstupním vektorem t .

Listing 4: Generátor posloupnosti obdélníkových impulsů

```

1 def rect_tr( t,tp,ts,td,code ):
2     n = np.size(code)
3     x=np.zeros(np.shape(t)) > 1
4     for i1 in range (0,n):
5         p = pulse.rect_p( t,td + i1*(tp+ts),td + tp + i1*(tp+ts) )
6         x = (x | p & code[i1]) * 1
7     return (x)

```

Funkce sinc_tr - $\text{sinc}(x)$ impuls Funkce, viz. listing 5, generuje posloupnost impulsů $\text{sinc}(x)$. Vstupními parametry jsou délka impulsu t_p , délka mezery mezi impulsy t_s , a zpoždění celé posloupnosti – mezera mezi začátkem časové osy t a středem prvního impulsu. Dále je vstupem vektor logických hodnot $code$. Ten určuje jednak kódovou posloupnost – tzn. přítomnost nebo nepřítomnost daného impulsu v rámci celé řady impulsů. Dále počet bitů posloupnosti $code$ určuje počet impulsů výsledného signálu.

Parametr pw je specifický pro tvar impulsu sinc . Určuje délku hlavního laloku funkce $\text{sinc}(x)$ uvnitř celého pulzu t_p . Jestliže parametr t_p můžeme nastavit podle bitové rychlosti $bitrate$, potom samotný $\text{sinc}(x)$ impuls může být v rámci jednoho bitu kratší - užší.

Výstupem funkce je vektor x vzorků impulsů odpovídajících vzorkovacím okmžikům definovaných vstupním vektorem t .

Listing 5: Generátor posloupnosti impulsů $\frac{\sin(x)}{x}$

```

1 def sinc_tr( t,ts,td,code,pw ):
2
3     #pw = 2e-3 # pulse width in ms
4     n = np.size(code)
5     x=np.zeros(np.shape(t))
6     for i1 in range (0,n):
7         p = pulse.sinc_p( t,td + i1*(ts),pw )
8         x = (x + p * code[i1])
9     return (x)

```

Funkce `rcos_tr` - Raised-cosine impuls Funkce, viz. listing 6, generuje posloupnost impulsů raised-cosine. Vstupními parametry jsou délka hlavního laloku impulsu t_p , délka mezery mezi impulsy t_s , a zpoždění celé posloupnosti – mezera mezi začátkem časové osy t a středem prvního impulsu. Dále je vstupem vektor logických hodnot `code`. Ten určuje jednak kódovou posloupnost – tzn. přítomnost nebo nepřítomnost daného impulsu v rámci celé řady impulsů. Dále počet bitů posloupnosti `code` určuje počet impulsů výsledného signálu.

Parametr pw je jako v případě *sinc* impulsu specifický pro tvar impulsu raised-cosine. Určuje délku hlavního laloku funkce *sinc*(x) uvnitř celého pulzu t_p . Jestliže parametr t_p můžeme nastavit podle bitové rychlosti *bitrate*, potom samotný *sinc*(x) impuls může být v rámci jednoho bitu kratší - užší.

Parametr α udává tzv. roll-off faktor funkce raised-cosine.

Výstupem funkce je vektor x vzorků impulsů odpovídajících vzorkovacím okmžíkům definovaných vstupním vektorem t .

Listing 6: Generátor posloupnosti raised-cosine impulsů

```

1 def rcos_tr( t,ts,td,code,pw,alpha ):
2
3     #pw = 2e-3 # pulse width in ms
4     #alpha = .5 # roll-off factor
5     n = np.size(code)
6     x=np.zeros(np.shape(t))
7     for i1 in range (0,n):
8         p = pulse.rcos_p( t,td + i1*(ts),pw,alpha )
9         x = (x + p * code[i1])
10    return (x)

```

Důležité: Parametr t_d znamená u posloupnosti obdélníkových impulsů zpoždění posloupnosti vzhledem k náběžné hraně prvního pulzu, u posloupností impulsů tvaru *sinc*(x) a raised-cosine vzhledem je ke středu hlavního laloku. To je potřeba respektovat, pokud se posloupnosti umisťují na časové ose společně.

5.1.3 Generátory pseudonáhodných posloupností

Funkce, které byly vytvořeny proto, aby generovaly posloupnosti bitů — generátory kódových a pseudonáhodných posloupností — jsou sdruženy v souboru `PRN_bitstreams.py`. Tyto

funkce jsou vhodné jednak k testování digitálních rádiových systémů a potom k rozproštění kmitočtového spektra v systémech typu DSSS (z angl. *Direct Sequence Spread Spectrum*).

tri_seq: Tří-bitový SSRG. Funkce, viz. listing 7, generuje bitovou posloupnost pomocí posuvného registru se zavedenými několika zpětnými vazbami SSRG⁸. V tomto konkrétním případě je použit generátor ve Fibonacciho tvaru s tří-bitovým registrem. Prvním vstupem funkce je parametr `init_reg` logický vektor (3 bity), který určuje počáteční stav registru před startem procesu generování posloupnosti. Druhým vstupním parametrem je opět tříbitový logický vektor `fb_reg`, který definuje zpětné vazby posuvného registru. Logická "0" v tomto vektoru znamená, že zpětná vazba není z daného bitu posuvného registru odvadana. Naopak logická "1" indikuje přítomnost zpětné vazby z příslušného bitu registru.

Výstupem funkce je 10-ti bitový vektor logických hodnot - vygenerovaná bitová posloupnost.

Listing 7: Generátor jednoduché posloupnosti. 3-bitový SSRG, generována je posloupnost 10-ti bitů.

```

1  # 10-bits sequence outputting 3-bits SSRG in a Fibonacci form.
2  def tri_seq(init_reg,fb_reg):
3      # Output register
4      x=np.zeros([10])
5      # Shift register
6      shft_reg = init_reg
7      # Feedback registers - bit '1' means -> FB is connected
8      # defined as an input argument fb_reg
9      for i1 in range (0,9):
10         in1 = int(np.dot(shft_reg,fb_reg)%2)
11         x[i1] = shft_reg[2]
12         shft_reg = np.roll(shft_reg,1)
13         shft_reg[0] = in1
14
15     return (x.astype(bool))

```

gold_seq: Generátor Goldova kódu. Funkce, viz. listing 8, generuje Goldovu pseudonáhodnou bitovou posloupnost, tzv. Goldův kód. Jsou použity dva 10-bitové posuvné registry se zavedenými vlastními několika zpětnými vazbami a se dvěma vazbami *vzájemnými*. Vzájemné vazby zabezpečují navázání registrů a spojení dvou nezávislých posloupností do

⁸z angl. *Simple Shift Register Generator*

jedné výsledné. Rozmístění vlastních zpětných vazeb jednotlivých registrů, stejně jako jejich počáteční stavy, jsou v souladu se standardem družicového navigačního systému GPS tak, aby generované posloupnosti odpovídaly posloupnostem použitým v tomto systému.

Prvními dvěma vstupními parametry jsou umístění *vzájemných* vazeb x_1 a x_2 . Jedná se o celá čísla v rozsahu 0–9, která definují číslo bitu posuvného registru, z něž je vazba odvedena⁹. Třetím vstupním parametrem je opět celé číslo N_{codes} , které určuje počet period kódu vygenerovaných na výstupu¹⁰.

Výstupem funkce je vektor logických hodnot o délce $(N_{\text{codes}} \cdot 1023)$ – vygenerovaných N_{codes} period Goldovy posloupnosti.

Listing 8: Generátor Goldovy posloupnosti.

```

1  # Gold sequence generator - two 10-bits SSRG
2  def gold_seq(x1,x2,N_codes):
3      # Output register
4      x=np.zeros([N_codes*1023])
5      # Maximum-length sequence generators:
6      # Shift registers
7      shft_reg_1 = np.array ([1,1,1,1,1,1,1,1,1,1])
8      shft_reg_2 = np.array ([1,1,1,1,1,1,1,1,1,1])
9      # Feedback registers - bit '1' means -> FB is connected
10     fbck_reg_1 = np.array ([0,0,1,0,0,0,0,0,0,1])
11     fbck_reg_2 = np.array ([0,1,1,0,0,1,0,1,1,1])
12
13     for i1 in range (0,N_codes*1023):
14         g1 = shft_reg_1 [9]
15         g2 = ( shft_reg_1[x1] + shft_reg_2[x2] ) % 2
16         in1 = np.dot(shft_reg_1,fbck_reg_1) % 2
17         in2 = np.dot(shft_reg_2,fbck_reg_2) % 2
18         x[i1] = ( g1 + g2 ) % 2
19         shft_reg_1 = np.roll(shft_reg_1,1)
20         shft_reg_1[0] = in1
21         shft_reg_2 = np.roll(shft_reg_2,1)
22         shft_reg_2[0] = in2
23
24     return (x.astype(bool))

```

⁹Jednotlivé kombinace x_1 a x_2 produkují unikátní posloupnost. Standard popisující systém GPS precizně definuje konkrétní kombinace pro kódy petřící jednotlivým družicím a jiným typům vysílačů v systému (např. družice SBAS, pseudolity a jiné)

¹⁰Jedna perioda Goldovy posloupnosti obsahuje 1023 bitů.

5.2 Modulátory

V této části jsou popisovány funkce uskutečňující modulace signálu — tvarování komplexního signálu v základním pásmu a jeho transformaci do vyšších kmitočtů - konverzi do tzv., *passbandu*. Funkce jsou umístěny v podadresáři `modulators`.

5.2.1 Tvarovače signálu v základním pásmu

V základním pásmu — *basebandu* — je potřeba vytvořit komplexní signál, složený ze symfázní (I) a kvadraturní (Q) složky. Aktuální úrovně těchto dvou složek definují okamžitou fázi a amplitudu signálu v *passbandu* — tedy na výstupu *Up-Convertoru*, potažmo celého vysílače.

Zde jsou popisovány funkce, vytvořené k tvarování signálu základního pásma. Jsou sdruženy v souboru `constellation_mappers.py`. Na základě vstupní bitové posloupnosti – `data` – je tvarován výstupní komplexní signál $I + jQ$ jednak podle požadované výsledné digitální modulace a také podle požadovaného tvaru impulzu.

`rect_bpsk_map`: BPSK modulace s obdélníkovým impulzem. Funkce tvaruje komplexní signál základního pásma pro modulaci BPSK s obdélníkovým impulzem. Vstupem je časová osa `t`, bitová posloupnost (vektor logických hodnot) `data`, která bude namodulována, požadovaná bitová rychlost `b_rate` a volitelně parametry výstupního signálu:

- `tp` je hodnota t_p , šířka pulzu (bitu). Pokud není tento parametr explicitně definován při volání funkce, je přednastaveno $t_p = \frac{1}{b_rate}$.
- `td` je hodnota t_d , zpoždění posloupnosti od počátku časové osy. Pokud není tento parametr explicitně definován při volání funkce, je přednastaveno $t_d = 0$.
- `ts` je hodnota t_s , mezera mezi impulzy. Pokud není tento parametr explicitně definován při volání funkce, je přednastaveno $t_s = 0$.

V případě BPSK je kvadraturní signál konstantní, roven nule; symfázní signál nabývá hodnoty 1 při aktuální hodnotě `data` rovné logické 1 a hodnoty -1, pokud je daný bit vektoru `data` roven logické 0.

rect_qpsk_map: QPSK modulace s obdélníkovým impulzem. Funkce tvaruje komplexní signál základního pásma pro modulaci QPSK s obdélníkovým impulzem. Vstupem je časová osa `t`, bitová posloupnost (vektor logických hodnot) `data`, která bude namodulována, požadovaná symbolová rychlost `s_rate` a volitelně parametry výstupního signálu:

- `tp` je hodnota t_p , šířka pulzu (symbolu). Pokud není tento parametr explicitně definován při volání funkce, je přednastaveno $t_p = \frac{1}{s_rate}$.
- `td` je hodnota t_d , zpoždění posloupnosti od počátku časové osy. Pokud není tento parametr explicitně definován při volání funkce, je přednastaveno $t_d = 0$.
- `ts` je hodnota t_s , mezera mezi impulzy. Pokud není tento parametr explicitně definován při volání funkce, je přednastaveno $t_s = 0$.

Modulace QPSK přenáší v jednom symbolu dva datové bity. Proto je datová posloupnost nejprve rozdělena na dvojice (data jsou přeskupeny do matice `data_2s` o dvou sloupcích). Pokud počet bitů v původním vstupním vektoru `data` není sudý, je doplněna nula. *Baseband* signál modulace QPSK je následně tvoren tak, že symfázní signál nabývá hodnoty 1, resp. -1 pro aktuální hodnoty logické 1, resp. 0 prvního bitu aktuální dvojice vstupních bitů (rozuměj aktuálního řádku matice `data_2s`). Obdobně je tvarován i kvadraturní signál, ten ovšem vychází z druhého bitu daného řádku matice `data_2s`.

sinc_bpsk_map: BPSK modulace s impulzem tvaru $\text{sinc}(x)$. Funkce tvaruje komplexní signál základního pásma pro modulaci BPSK s obdélníkovým impulzem. Vstupem je časová osa `t`, bitová posloupnost (vektor logických hodnot) `data`, která bude namodulována, požadovaná bitová rychlost `b_rate` a volitelně parametry výstupního signálu:

- `tp` je hodnota t_p , šířka pulzu (bitu). Pokud není tento parametr explicitně definován při volání funkce, je přednastaveno $t_p = \frac{1}{b_rate}$.
- `td` je hodnota t_d , zpoždění posloupnosti od počátku časové osy. Pokud není tento parametr explicitně definován při volání funkce, je přednastaveno $t_d = 0$.
- `ts` je hodnota t_s , mezera mezi impulzy. Pokud není tento parametr explicitně definován při volání funkce, je přednastaveno $t_s = 0$.

V případě BPSK je kvadrurní signál konstantní, roven nule; symfázní signál nabývá hodnoty 1 při aktuální hodnotě `data` rovné logické 1 a hodnoty -1, pokud je daný bit vektoru `data` roven logické 0.

sinc_qpsk_map: QPSK modulace s obdélníkovým impulzem. Funkce tvaruje komplexní signál základního pásma pro modulaci QPSK s obdélníkovým impulzem. Vstupem je časová osa `t`, bitová posloupnost (vektor logických hodnot) `data`, která bude namodulována, požadovaná symbolová rychlost `s_rate` a volitelně parametry výstupního signálu:

- `tp` je hodnota t_p , šířka pulzu (symbolu). Pokud není tento parametr explicitně definován při volání funkce, je přednastaveno $t_p = \frac{1}{s_rate}$.
- `td` je hodnota t_d , zpoždění posloupnosti od počátku časové osy. Pokud není tento parametr explicitně definován při volání funkce, je přednastaveno $t_d = 0$.
- `ts` je hodnota t_s , mezera mezi impulzy. Pokud není tento parametr explicitně definován při volání funkce, je přednastaveno $t_s = 0$.

Modulace QPSK přenáší v jednom symbolu dva datové bity. Proto je datová posloupnost nejprve rozdělena na dvojice (data jsou přeskupeny do matice `data_2s` o dvou sloupcích). Pokud počet bitů v původním vstupním vektoru `data` není sudý, je doplněna nula. *Baseband* signál modulace QPSK je následně tvoren tak, že symfázní signál nabývá hodnoty 1, resp. -1 pro aktuální hodnoty logické 1, resp. 0 prvního bitu aktuální dvojice vstupních bitů (rozuměj aktuálního řádku matice `data_2s`). Obdobně je tvarován i kvadrurní signál, ten ovšem vychází z druhého bitu daného řádku matice `data_2s`.

5.2.2 Up – Converter

5.3 Pomocné funkce

5.4 Filtry

5.4.1 Tvarovače impulzu

Reference

- [1] B. Lathi and Z. Ding, *Modern Digital and Analog Communication Systems*, ser. Oxford series in electrical and computer engineering. Oxford University Press, 2009. [Online]. Available: <https://books.google.cz/books?id=dltNPwAACAAJ>
- [2] J. Bajer, R. Schreiber, and P. Bojda, “Correlation analysis of the faster-than-nyquist signal,” in *2017 International Conference on Military Technologies (ICMT)*, May 2017, pp. 602–607.
- [3] S. Haykin, *Communication systems*. Wiley, 2001. [Online]. Available: <https://books.google.cz/books?id=ieZSAAAAMAAJ>
- [4] J. Holmes, *Spread Spectrum Systems for GNSS and Wireless Communications*, ser. GNSS technology and applications series. Artech House, 2007, no. v. 45. [Online]. Available: <https://books.google.cz/books?id=-AUfAQAAIAAJ>