

Signali i sistemi

Domaći 2

Branislav Đumić
Br. indeksa: 0267/2020

Parametri

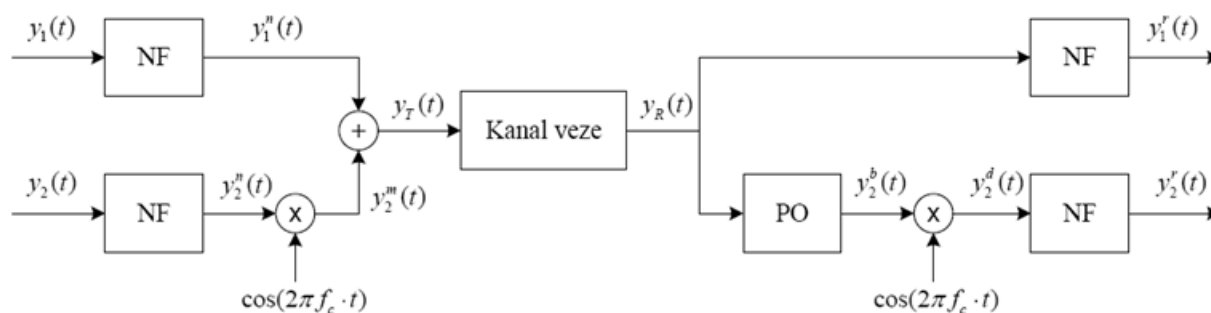
$P = 3, Q = 0$

Sadržaj

1	Filtri FDM sistema za paralelni prenos	3
2	Određivanje frekvencijskih spektara signala $y_1(t)$ i $y_2(t)$	5
2.1	Frekvencijski spektar signala $y_1(t)$	5
2.2	Frekvencijski spektar signala $y_2(t)$	8
3	Izbor graničnih učestanosti filtera i učestanosti nosioca f_c	10
4	Amplitudske karakteristike signala sistema	12
4.1	Signal $y_1^n(t)$	12
4.2	Signal $y_2^n(t)$	13
4.3	Signal $y_2^m(t)$	14
4.4	Signal $y_T(t)$	16
4.5	Signal $y_R(t)$	17
4.6	Signal $y_2^b(t)$	18
4.7	Signal $y_2^d(t)$	19
4.8	Signal $y_1^r(t)$	20
4.9	Signal $y_2^r(t)$	21
5	Rekonstruisani signali	22
6	Kodovi za generisanje grafika	22

Zadatak 1. Furijeova transformacija

Razmatra se FDM (Frequency Division Multiplex) sistem za nezavisan paralelni prenos dva signala kroz zajednički kanal veze, opisan blok-dijagramom na slici:



Niskopropusni filtri su označeni sa NF, propusnik opsega sa PO. Signal $y_1(t)$ prenosi se u osnovnom opsegu učestanosti, a signal $y_2(t)$ se pomera u opseg narednog frekvencijskog kanala, tj. amplitudski se modulira nosiocem učestanosti f_c . Kanal veze se može modelovati niskopropusnim filtrom propusnog opsega f_k . Pretpostaviti da su svi filtri idealni.

- Ukratko opisati funkciju svih filtara datih u blok dijagramu.
- Predložiti i obrazložiti izbor učestanosti nosioca f_c pri amplitudskoj modulaciji drugog signala, kao i zbor graničnih učestanosti svih filtara.
- Analitički odrediti spektre i skicirati amplitudske karakteristike originalnih signala $y_1(t)$ i $y_2(t)$.
- Skicirati amplitudske karakteristike spektar asledećih signala:
 - amplitudski modulisanog signala $y_2^m(t)$,
 - transmisionog signala $y_T(t)$ (na ulazu u kanal veze) i prijemnog signala $y_R(t)$ (na izlazu iz kanala veze),
 - filtracijom dobijenog signala $y_2^b(t)$ i njemu odgovarajućeg signala dobijenog demodulacijom $y_2^d(t)$,
 - finalno rekonstruisanih signala $y_1^r(t)$ i $y_2^r(t)$.
- Obrazložiti kako očekujemo da izgleda vremenski oblik finalno rekonstruisanih signala $y_1^r(t)$ i $y_2^r(t)$.

Rešenje

U daljem tekstu FDM sistem za paralelni prenos iz zadatka nazivaćemo Sistem. Sistem se sastoji od više filtara, sabirača i modulatora. Funkcije filtara će biti opisane u narednoj sekciji. Sistem sadrži dva modulatora kojima se moduliše i demoduliše jedan od prenošenih signala. Sabirač služi za kombinovanje prenošenih signala, kako bi se oni mogli prenositi istim kanalom.

1 Filtri FDM sistema za paralelni prenos

Prvo ćemo imenovati različite filtre u Sistemu. Usvajamo notaciju za obeležavanje filtara:

$$X[TF] \Big| \omega_x : x(t) \rightarrow y(t).$$

Pri ovoj notaciji X je ime funkcije frekvencijskog odziva filtra $X(j\omega)$, TF je tip filtra, ω_x je granična učestanost filtra, dok su signali $x(t)$ i $y(t)$ ulazni, odnosno izlazni, signal filtra. Pri ovoj notaciji filteri koje imamo u Sistemu su:

$$\begin{aligned} H_{n_1} [NF] & \Big| \omega_{u1} : y_1(t) \rightarrow y_1^n(t) \\ H_{n_2} [NF] & \Big| \omega_{u2} : y_2(t) \rightarrow y_2^n(t) \\ H_{kv} [NF] & \Big| \omega_{kv} : y_T(t) \rightarrow y_R(t) \\ H_{Rb_1} [NF] & \Big| \omega_{i1} : y_R(t) \rightarrow y_1^r(t) \\ H_{Rb_2} [PO] & \Big| \omega_{o2} : y_R(t) \rightarrow y_2^b(t) \\ H_{d_2r_2} [NF] & \Big| \omega_{i2} : y_2^d(t) \rightarrow y_2^r(t) \end{aligned}$$

H_{n_1} je ulazni filter Sistema. Njegova uloga je uklanjanje slabije izraženih učestanosti iz spektra ulaznog signala $y_1(t)$. Ovime se omogućava prenošenje signala $y_1(t)$ i $y_2(t)$ bez njihovog mešanja, tako što će sekanalom veze prenositi samo korisne, odnosno najizraženije, učestanosti.

H_{n_2} je, takođe, ulazni filter Sistema. Vršiti istu funkciju kao i filter H_{n_1} , ali nad signalom $y_2(t)$.

H_{kv} je filter kojim modeluje kanal veze. Prenošeni signal $y_T(t)$ prenosi informacije na nižim učestanostima, tako da je potrebno sprečiti smenje na ovim učestanostima. Na višim učestanostima nema značajnih informacija. Kanal veze se treba modelovati tako da što manje utiše na prenošeni signal. Zbog ovoga ga, kanal veze modelujemo filtrom niskih učestanosti.

H_{Rb_1} prihvata prenešeni signal i iz njegovog spektra izdvaja učestanosti koje nose korisne o prvobitnom signalu $y_1(t)$. Pod korisnim inforacijam smatraju se učestanosti prvobitno izdvojene ulaznim filtrima.

H_{Rb_2} takođe prihvata prenešeni signal, ali iz njega izdvaja spektar učestanosti koji predstavlja signal $y_2(t)$. Pošto je pre prenošenja signal bio modulisan, njegov spektar se nalazi levo i desno u odnosu na spektar ulaznog signala $y_2(t)$. Zbog ovoga je ovaj filter propusnik opsega; potrebno je prigušiti spektar učestanosti signala $y_1(t)$, kao i učestanosti viših od onih na kojima se prenosi signal $y_2(t)$.

$H_{d_2r_2}$ je filter koji uklanja duplikate spektra početnog signala, tj. one učestanosti koje su se pojavile u spektru usled postupaka modulacije i demodulacije.

2 Određivanje frekvencijskih spektara signala $y_1(t)$ i $y_2(t)$

Na osnovu zadatih parametara mogu se odrediti analitički oblici signala $y_1(t)$ i $y_2(t)$:

$$y_1(t) = \sin(2000\pi t)e^{-100t}u(t) \quad (1)$$

$$y_2(t) = [u(t) - u(t - 0.0005)] - [u(t - 0.0005) - u(t - 0.001)] \quad (2)$$

Frekvencijski spektri ulaznih signala, $Y_1(j\omega)$ i $Y_2(j\omega)$ jednaki su njihovim Furijeovim transformacijama:

$$Y_1(j\omega) = \mathcal{F}\{y_1(t)\} \quad (3)$$

$$Y_2(j\omega) = \mathcal{F}\{y_2(t)\} \quad (4)$$

2.1 Frekvencijski spektar signala $y_1(t)$

Kako bi lakše odredili frekvencijski spektar signala $y_1(t)$, napisaćemo ga kao proizvod signala $y_{11}(t)$ i $y_{12}(t)$:

$$y_1(t) = y_{11}(t) \cdot y_{12}(t) \quad (5)$$

$$y_{11}(t) = \sin(2000\pi t) \quad (6)$$

$$y_{12}(t) = e^{-100t}u(t) \quad (7)$$

Odredimo sada spektar signala $Y_1(j\omega)$. Prema osobinama Furijeove transformacije imamo:

$$Y_1(j\omega) = \mathcal{F}\{y_1(t)\} = \mathcal{F}\{y_{11}(t) \cdot y_{12}(t)\} = \frac{1}{2\pi} \mathcal{F}\{y_{11}(t)\} * \mathcal{F}\{y_{12}(t)\} \quad (8)$$

Odredimo sada spektar signala $y_{11}(t)$:

$$Y_{11}(j\omega) = \mathcal{F}\{y_{11}(t)\} = \mathcal{F}\{\sin(2000\pi t)\} = \mathcal{F}\left\{\frac{e^{j2000\pi t} - e^{-j2000\pi t}}{2j}\right\}$$

$$Y_{11}(j\omega) = \frac{1}{2j} \left(\mathcal{F}\{e^{j2000\pi t}\} - \mathcal{F}\{e^{-j2000\pi t}\} \right)$$

$$Y_{11}(j\omega) = \frac{1}{2j} \left(2\pi\delta(\omega - 2000\pi) - 2\pi\delta(\omega + 2000\pi) \right)$$

Konačan izraz za spektar signala $y_{11}(t)$ je:

$$Y_{11}(j\omega) = \frac{\pi}{j} \left(\delta(\omega - 2000\pi) - \delta(\omega + 2000\pi) \right) \quad (9)$$

Izračunajmo spektar signala $y_{12}(t)$:

$$\begin{aligned}
 Y_{12}(j\omega) &= \mathcal{F}\{y_{12}(t)\} = \int_{-\infty}^{+\infty} e^{-100t} u(t) e^{-j\omega t} dt = \int_0^{+\infty} e^{-100t} e^{-j\omega t} dt \\
 Y_{12}(j\omega) &= \int_0^{+\infty} e^{-(100+j\omega)t} dt = -\frac{1}{100+j\omega} \left[e^{-100t} e^{-j\omega t} \right] \Big|_0^{+\infty} \\
 Y_{12}(j\omega) &= -\frac{1}{100+j\omega} \left[\lim_{\tau \rightarrow +\infty} (e^{-100\tau} e^{-j\omega\tau}) - \lim_{\tau \rightarrow +\infty} (e^{-100\tau} e^{-j\omega\tau}) \right] \Big|_0^{+\infty} \\
 Y_{12}(j\omega) &= -\frac{1}{100+j\omega} \left[\lim_{\tau \rightarrow +\infty} (e^{-100\tau}) \overset{0}{\lim_{\tau \rightarrow +\infty} (e^{-j\omega\tau})} - 1 \right]
 \end{aligned}$$

Konačan izraz za spektar signala $y_{12}(t)$:

$$Y_{12}(j\omega) = \frac{1}{100+j\omega} \quad (10)$$

Iz izraza (8), (9) i (10) imamo za spektar signala $y_1(t)$, $Y_1(j\omega)$:

$$\begin{aligned}
 Y_1(j\omega) &= \frac{1}{2\pi} Y_{11}(j\omega) * Y_{11}(j\omega) \\
 Y_1(j\omega) &= \frac{1}{2\pi} \int_{-\infty}^{+\infty} \frac{\pi}{j} \left(\delta(\lambda - 2000\pi) - \delta(\lambda + 2000\pi) \right) \frac{1}{100+j(\omega-\lambda)} d\lambda \\
 Y_1(j\omega) &= \frac{1}{2j} \left(\int_{-\infty}^{+\infty} \frac{\delta(\lambda - 2000\pi)}{100+j(\omega-\lambda)} d\lambda - \int_{-\infty}^{+\infty} \frac{\delta(\lambda + 2000\pi)}{100+j(\omega-\lambda)} d\lambda \right)
 \end{aligned}$$

Konačno se za frekvencijski spektar $Y_1(j\omega)$ dobija izraz:

$$Y_1(j\omega) = \frac{1}{2j} \left(\frac{1}{100+j(\omega-2000\pi)} - \frac{1}{100+j(\omega+2000\pi)} \right) \quad (11)$$

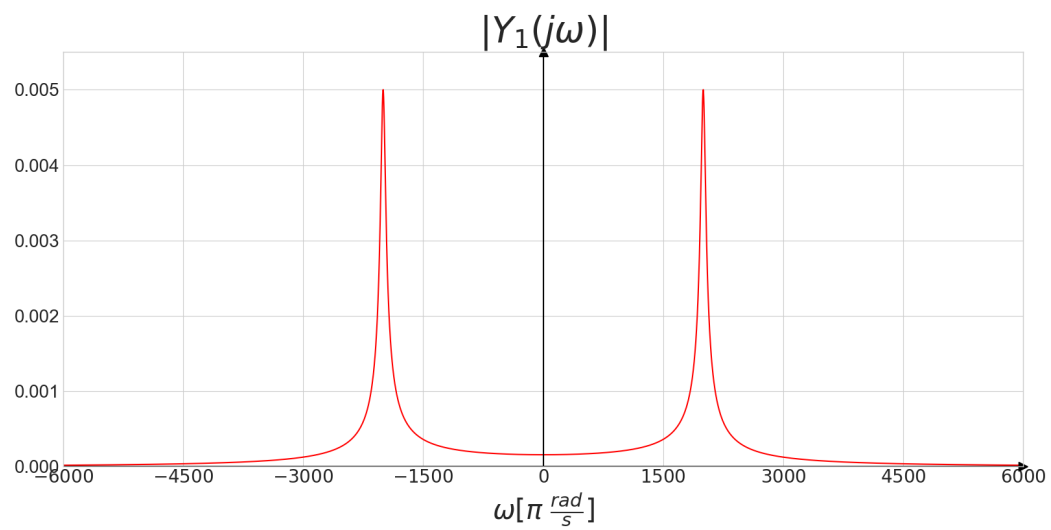
Sređivanjem izraza (11) dobijamo:

$$Y_1(j\omega) = \frac{2000\pi}{100^2 - \omega^2 + 2000^2\pi^2 + j200\omega}$$

Oдавde možemo odrediti amplitudsku frekvencijsku karakteristiku signala $y_1(t)$:

$$|Y_1(j\omega)| = \frac{2000\pi}{\sqrt{(100^2 - \omega^2 + 2000^2\pi^2)^2 + 200^2\omega^2}} \quad (12)$$

Kada se nacrtaju grafik amplitudne frekvencijske karakteristike dobijamo sledeću sliku:



Slika 1: Amplitudski spektar $Y_1(j\omega)$

2.2 Frekvencijski spektar signala $y_2(t)$

Kako bi lakše odredili frekvencijski spektar signala $y_2(t)$, napisaćemo ga kao razliku signala $y_{21}(t)$ i $y_{22}(t)$:

$$y_2(t) = y_{21}(t) - y_{22}(t) \quad (13)$$

$$y_{21}(t) = u(t) - u(t - 0.0005) \quad (14)$$

$$y_{22}(t) = u(t - 0.0005) - u(t - 0.001) \quad (15)$$

Za signale $y_{21}(t)$ i $y_{22}(t)$ važi:

$$y_{22}(t) = y_{21}(t - 0.0005)$$

Iz osobina furijeove transformacije važe sledeće relacije:

$$Y_2(j\omega) = \mathcal{F}\{y_2(t)\} = \mathcal{F}\{y_{21}(t) - y_{22}(t)\} = \mathcal{F}\{y_{21}(t)\} - \mathcal{F}\{y_{22}(t)\} \quad (16)$$

$$Y_2(j\omega) = Y_{21}(j\omega) - Y_{22}(j\omega) \quad (17)$$

$$Y_{22}(j\omega) = \mathcal{F}\{y_1(t - 0.0005)\} = e^{-j\omega 0.0005} Y_{21}(j\omega) \quad (18)$$

Izračunajmo spektar signala $y_{21}(t)$:

$$\begin{aligned} Y_{21}(j\omega) &= \int_{-\infty}^{+\infty} (u(t) - u(t - 0.0005)) e^{-j\omega t} dt = \int_0^{0.0005} e^{-j\omega t} dt \\ Y_{21}(j\omega) &= -\frac{1}{j\omega} \cdot e^{-j\omega t} \Big|_0^{0.0005} = \frac{1 - e^{-j\omega 0.0005}}{j\omega} \end{aligned} \quad (19)$$

Iz jednačina (16), (18) i (19) možemo sračunati konačan izraz za spektar $Y_2(j\omega)$:

$$\begin{aligned} Y_2(j\omega) &= Y_{21}(j\omega) - e^{-j\omega 0.0005} Y_{21}(j\omega) \\ Y_2(j\omega) &= \frac{1}{j\omega} \left(1 - e^{-j\omega 0.0005} \right)^2 \\ Y_2(j\omega) &= \frac{1}{j\omega} \left(e^{-j\omega 0.00025} \left(e^{j\omega 0.00025} - e^{-j\omega 0.00025} \right) \right)^2 \\ Y_2(j\omega) &= \frac{1}{j\omega} e^{-j\omega 0.0005} \left(2j \cdot \sin(0.00025\omega) \right)^2 \\ Y_2(j\omega) &= -\frac{4}{j\omega} e^{-j\omega 0.0005} \cdot \sin^2(0.00025\omega) \cdot \frac{0.00025^2 \omega^2}{0.00025^2 \omega^2} \end{aligned}$$

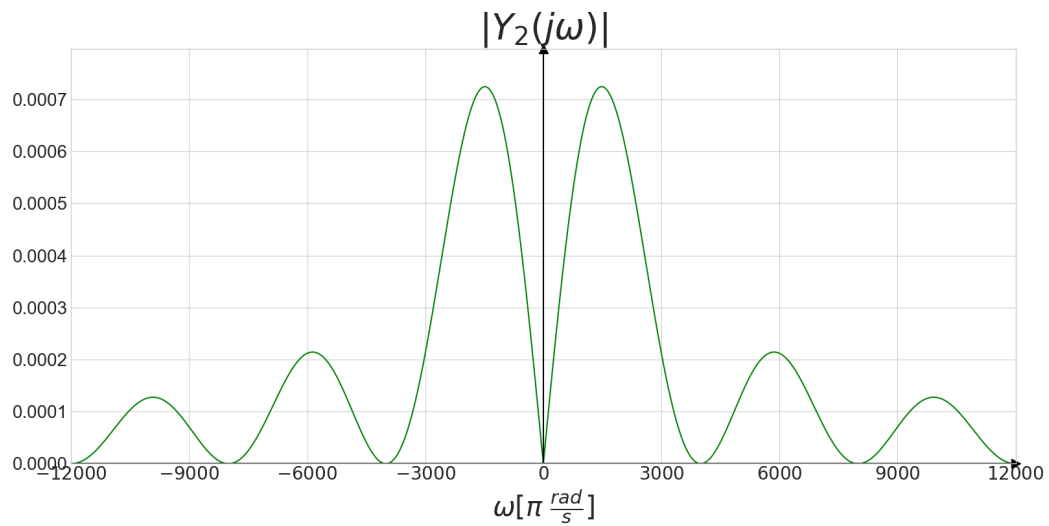
Konačno za frekvencijski spektar $Y_2(j\omega)$ dobijamo:

$$Y_2(j\omega) = 2.5 \cdot 10^{-7} j\omega e^{-j\omega 0.0005} \cdot \text{sinc}^2(0.00025\omega) \quad (20)$$

Iz jednačine (20) možemo odrediti amplitudsku frekvencijsku karakteristiku signala $y_2(t)$:

$$|Y_2(j\omega)| = 2.5 \cdot 10^{-7} |\omega| \operatorname{sinc}^2(0.00025\omega) \quad (21)$$

Kada se nacрта grafik amplitudske frekvencijske karakteristike dobijmao sledeću sliku:



Slika 2: Amplitudski spektar $Y_2(j\omega)$

3 Izbor graničnih učestanosti filtara i učestanosti nosioca f_c

Izaberimo i obrazložimo učestanosti svih filtara u Sistemu i granične učestanosti nosioca. Na osnovu usvojenih graničnih učestanosti možemo odrediti izraze za amplitudске frekvencijske karakteristike svih filtara sistema.

Na osnovu slike 1. vidimo da je najznačajniju komponentu signala $y_1(t)$ moguće predstaviti učestanostima do $4000\pi \frac{rad}{s}$, tako da za učestanost filtra H_{n1} biramo $\omega_{u1} = 4000\pi \frac{rad}{s}$.

$$|H_{n1}(j\omega)| = \begin{cases} 1 & |\omega| < \omega_{u1} \\ 0 & |\omega| > \omega_{u1} \end{cases} \quad (22)$$

Na osnovu slike 2. vidimo da je najveći deo spektra signala $y_2(t)$ obuhvaćen učestanostima manjim od $\omega_g = 8000\pi \frac{rad}{s}$, tako da za učestanost filtra H_{n2} biramo $\omega_{u2} = 8000\pi \frac{rad}{s}$. Čak i ovim izborom, značajan deo učestanosti biva izostavljen, ali projektovanje filtara sa većim propusnim opsegom bi bilo nepraktično.

$$|H_{n2}(j\omega)| = \begin{cases} 1 & |\omega| < \omega_{u2} \\ 0 & |\omega| > \omega_{u2} \end{cases} \quad (23)$$

Učestanost nosioca f_c treba da odaberemo tako da ne dođe do mešanja signala $y_1^n(t)$ i $y_2^m(t)$ pri njihovom sabiranju. Kako bi to postigli, modulacijom signala $y_2^n(t)$ potrebno je pomeriti njegov spektar za $\omega_c = \omega_{u1} + \omega_{u2} = 12000\pi \frac{rad}{s}$. Kako je $f_c = \frac{\omega_c}{2\pi}$, dobijamo da je učestanost nosioca $f_c = 6000Hz$.

Granična učestanost filtra H_{Rb1} treba da bude jednaka učestanosti filtra H_{n1} , pošto oni treba da izoluju isti opseg učestanosti. Dakle, $\omega_{i1} = 4000\pi \frac{rad}{s}$.

$$|H_{Rb1}(j\omega)| = \begin{cases} 1 & |\omega| < \omega_{i1} \\ 0 & |\omega| > \omega_{i1} \end{cases} \quad (24)$$

Filtar H_{Rb2} ima dve granične učestanosti, s obzirom da je on filter propusnik opsega. Ovaj filter bi trebalo da izoluje modulisan signal $y_2^m(t)$ tj. razdvoji ga od signala $y_1^n(t)$. Donja granična učestanost, ω_{o2d} jednaka je gornjoj graničnoj učestanosti filtra H_{n1} , jer na njoj prestaju učestanosti kojima se prenosi signal $y_1^n(t)$. Dakle, imamo da je $\omega_{o2d} = 4000\pi \frac{rad}{s}$. Gornja granična učestanost ω_{o2g} jednaka je

najvišoj učestanosti na kojoj prenosimo informacije. Kako je signal koji izdvajamo nastao modulacijom, njegov imamo $\omega_{o2g} = \omega_{u2} + \omega_c = 20000\pi \frac{rad}{s}$.

$$|H_{Rb_2}(j\omega)| = \begin{cases} 0 & |\omega| < \omega_{o2d} \\ 1 & \omega_{o2d} < |\omega| < \omega_{o2g} \\ 0 & |\omega| > \omega_{o2g} \end{cases} \quad (25)$$

Granična učestanost filtra $H_{d_2r_2}$ treba da bude jednaka učestanosti filtra H_{n2} , pošto oni treba da izoluju isti opseg učestanosti. Dakle, $\omega_{i1} = 8000\pi \frac{rad}{s}$.

$$|H_{d_2r_2}(j\omega)| = \begin{cases} 1 & |\omega| < \omega_{i2} \\ 0 & |\omega| > \omega_{i2} \end{cases} \quad (26)$$

Granična učestanost filtra H_{kv} , kanala veze, mora biti dovoljno visoka da ne dođe do oštećenja signala $y_T(t)$ pri prenosu. Potrebno je obuhvatiti čitav spektar ovog signala. Imamo da je $\omega_{kv} = \omega_{u1} + 2 \cdot \omega_{u2} = 20000\pi \frac{rad}{s}$.

$$|H_{kv}(j\omega)| = \begin{cases} 1 & |\omega| < \omega_{kv} \\ 0 & |\omega| > \omega_{kv} \end{cases} \quad (27)$$

4 Amplitudske karakteristike signala sistema

4.1 Signal $y_1^n(t)$

Signal $y_1^n(t)$ nastaje propuštanjem signala $y_1(t)$ kroz filter H_{n1} . Usled filtracije, signal $y_1^n(t)$ sadrži samo učestanosti manje od $\omega_{u1} = 4000\pi \frac{rad}{s}$. Znamo da je izlazni signal filtra H_{n1} , $y_1^n(t)$, jednak konvoluciji ulaznog signala $y_1(t)$ i impulsnog odziva filtra $h_{n1}(t)$, odnosno

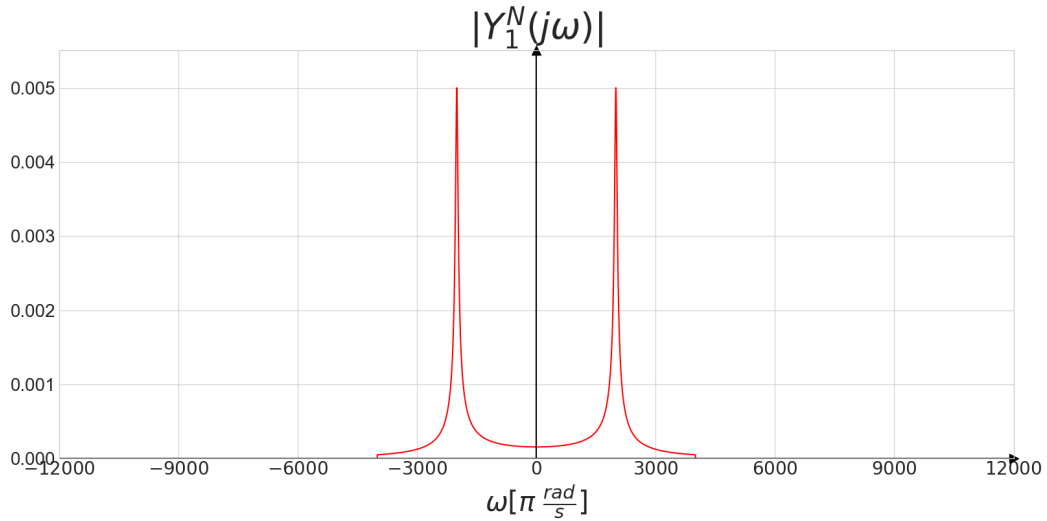
$$y_1^n(t) = y_1(t) * h_{n1}(t). \quad (28)$$

Prema osobinama Furijeove transformacije, u frekvencijskom domenu imamo:

$$Y_1^N(j\omega) = Y_1(j\omega) \cdot H_{n1}(j\omega) \quad (29)$$

$$|Y_1^N(j\omega)| = |Y_1(j\omega)| \cdot |H_{n1}(j\omega)|$$
$$|Y_1^N(j\omega)| = \begin{cases} |Y_1(j\omega)| & |\omega| < \omega_{u1} \\ 0 & |\omega| > \omega_{u1} \end{cases} \quad (30)$$

Na donjoj slici možemo videti amplitudski spektar signala $y_1^n(t)$.



Slika 3: Amplitudski spektar $|Y_1^N(j\omega)|$

4.2 Signal $y_2^n(t)$

Signal $y_2^n(t)$ nastaje propuštanjem signala $y_2(t)$ kroz filtar H_{n2} . Usled filtracije, signal $y_2^n(t)$ sadrži samo učestanosti manje od $\omega_{u2} = 8000\pi \frac{rad}{s}$. Znamo da je izlazni signal filtra H_{n2} , $y_2^n(t)$, jednak konvoluciji ulaznog signala $y_2(t)$ i impulsnog odziva filtra $h_{n2}(t)$, odnosno

$$y_2^n(t) = y_2(t) * h_{n2}(t). \quad (31)$$

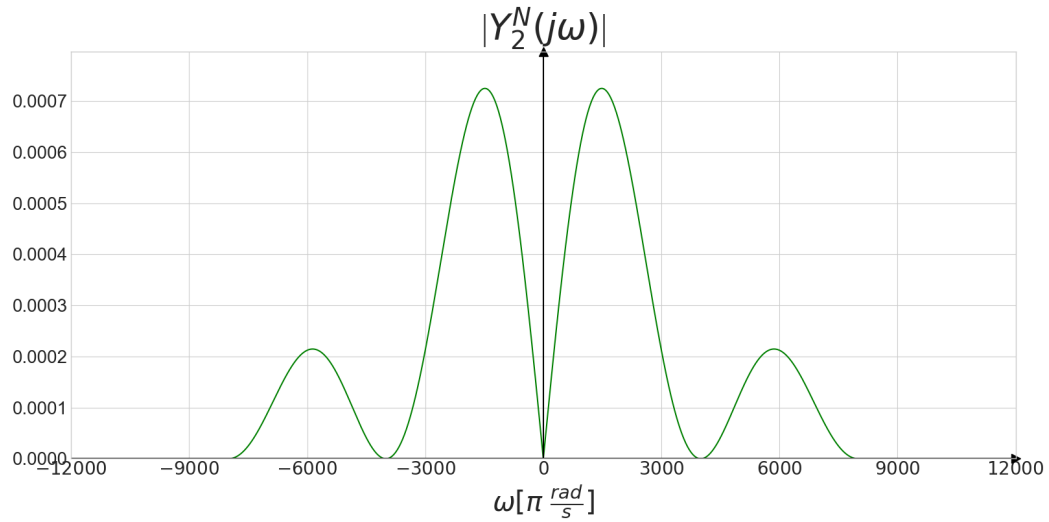
Prema osobinama Furijeove transformacije, u frekvencijskom domenu imamo:

$$Y_2^N(j\omega) = Y_2(j\omega) \cdot H_{n2}(j\omega) \quad (32)$$

$$|Y_2^N(j\omega)| = |Y_2(j\omega)| \cdot |H_{n2}(j\omega)|$$

$$|Y_2^N(j\omega)| = \begin{cases} |Y_2(j\omega)| & |\omega| < \omega_{u2} \\ 0 & |\omega| > \omega_{u2} \end{cases} \quad (33)$$

Na donjoj slici možemo videti amplitudski spektar signala $y_2^n(t)$.



Slika 4: Amplitudski spektar $|Y_2^N(j\omega)|$

4.3 Signal $y_2^m(t)$

Signal $y_2^m(t)$ nastaje modulisanjem signala $y_2^n(t)$ sinusoidom $\cos(2\pi f_c \cdot t)$. Dakle u vremenskom domenu imamo:

$$y_2^m(t) = y_2^n(t) \cos(2\pi f_c \cdot t). \quad (34)$$

Prelaskom u frekvencijski domen dobijamo sledeće:

$$Y_2^M(j\omega) = \frac{1}{2\pi} Y_2^N(j\omega) * \mathcal{F}\{\cos(2\pi f_c \cdot t)\} \quad (35)$$

Odredimo frekvencijski spektar signala $y_2^m(t)$.

$$\begin{aligned} Y_2^M(j\omega) &= \frac{1}{2\pi} Y_2^N(j\omega) * \mathcal{F}\left\{\frac{e^{j2\pi f_c t} + e^{-j2\pi f_c t}}{2}\right\} \\ Y_2^M(j\omega) &= \frac{1}{4\pi} Y_2^N(j\omega) * \left(\mathcal{F}\{e^{j2\pi f_c t}\} + \mathcal{F}\{e^{-j2\pi f_c t}\}\right) \\ Y_2^M(j\omega) &= \frac{1}{4\pi} Y_2^N(j\omega) * \left(2\pi\delta(\omega - 2\pi f_c) + 2\pi\delta(\omega + 2\pi f_c)\right) \\ Y_2^M(j\omega) &= \frac{1}{2} Y_2^N(j\omega) * \left(\delta(\omega - 2\pi f_c) + \delta(\omega + 2\pi f_c)\right) \\ Y_2^M(j\omega) &= \frac{1}{2} \int_{-\infty}^{+\infty} Y_2^N(j\lambda) \left(\delta(\omega - \lambda - 2\pi f_c) + \delta(\omega - \lambda + 2\pi f_c)\right) d\lambda \\ Y_2^M(j\omega) &= \frac{1}{2} \left[Y_2^N(j\lambda) \Big|_{\lambda=\omega-2\pi f_c} + Y_2^N(j\lambda) \Big|_{\lambda=\omega+2\pi f_c} \right] \end{aligned}$$

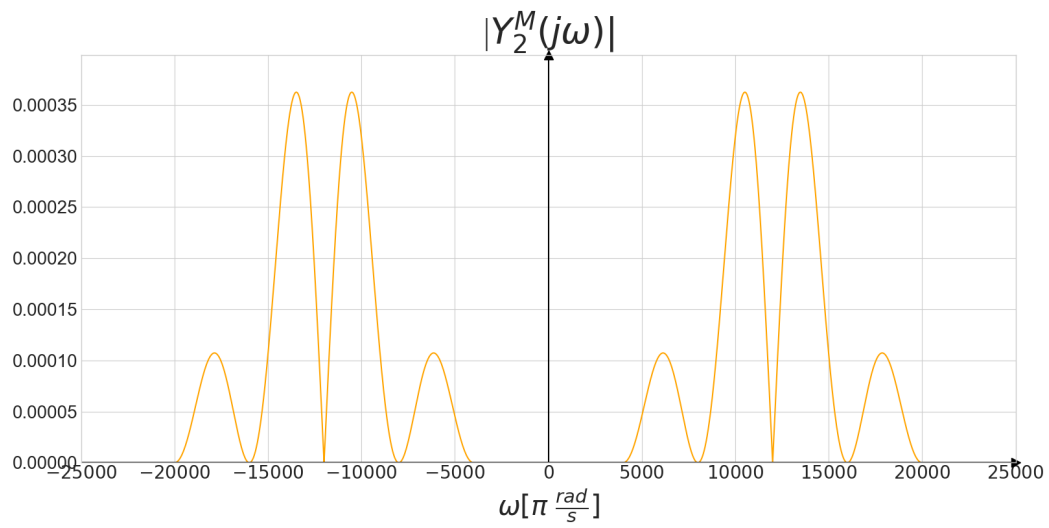
Konačno se za frekvencijski spektar dobija:

$$Y_2^M(j\omega) = \frac{1}{2} \left[Y_2^N(j(\omega - 2\pi f_c)) + Y_2^N(j(\omega + 2\pi f_c)) \right] \quad (36)$$

Sada možemo odrediti amplitudski spektar signala $y_2^m(t)$:

$$|Y_2^M(j\omega)| = \begin{cases} 0 & \omega \in (-\infty, -20000\pi) \\ \frac{1}{2} |Y_2^N(j(\omega + 2\pi f_c))| & \omega \in (-20000\pi, -4000\pi) \\ 0 & \omega \in (-4000\pi, 4000\pi) \\ \frac{1}{2} |Y_2^N(j(\omega - 2\pi f_c))| & \omega \in (4000\pi, 20000\pi) \\ 0 & \omega \in (20000\pi, +\infty) \end{cases} \quad (37)$$

Na sledećoj slici možemo videti amplitudski spektar signala $y_2^m(t)$:



Slika 5: Amplitudski spektar $|Y_2^M(j\omega)|$

4.4 Signal $y_T(t)$

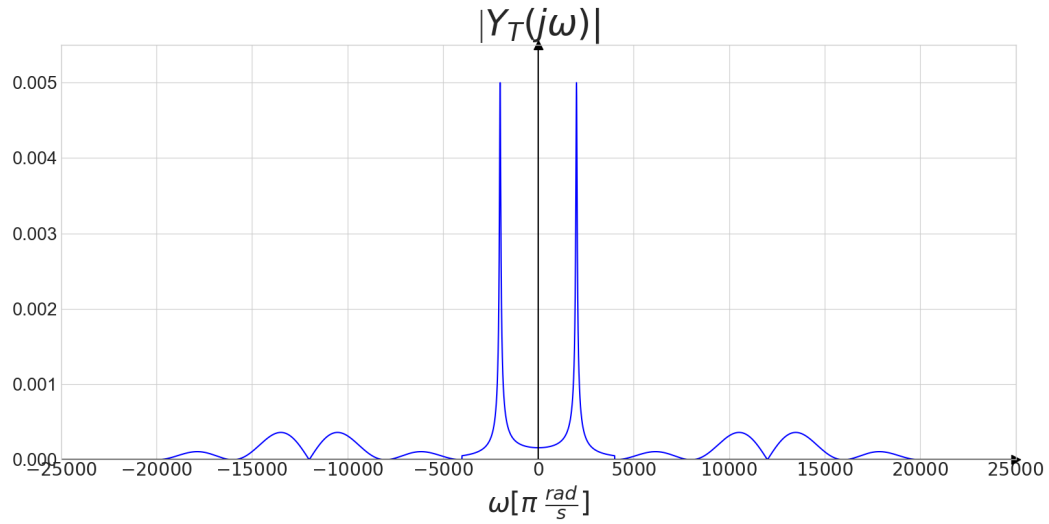
Signal $y_T(t)$ je signal koji se prenosi putem kanala veze. Nastaje sabiranjem signala $y_1^n(t)$ i $y_2^m(t)$. Dakle imamo:

$$y_T(t) = y_1^n(t) + y_2^m(t) \quad (38)$$

$$Y_T(j\omega) = Y_1^N(j\omega) + Y_2^M(j\omega) \quad (39)$$

$$|Y_T(j\omega)| = \begin{cases} 0 & \omega \in (-\infty, -20000\pi) \\ \frac{1}{2}|Y_2^N(j(\omega + 2\pi f_c))| & \omega \in (-20000\pi, -4000\pi) \\ |Y_1^N(j\omega)| & \omega \in (-4000\pi, 4000\pi) \\ \frac{1}{2}|Y_2^N(j(\omega + 2\pi f_c))| & \omega \in (4000\pi, 20000\pi) \\ 0 & \omega \in (20000\pi, +\infty) \end{cases} \quad (40)$$

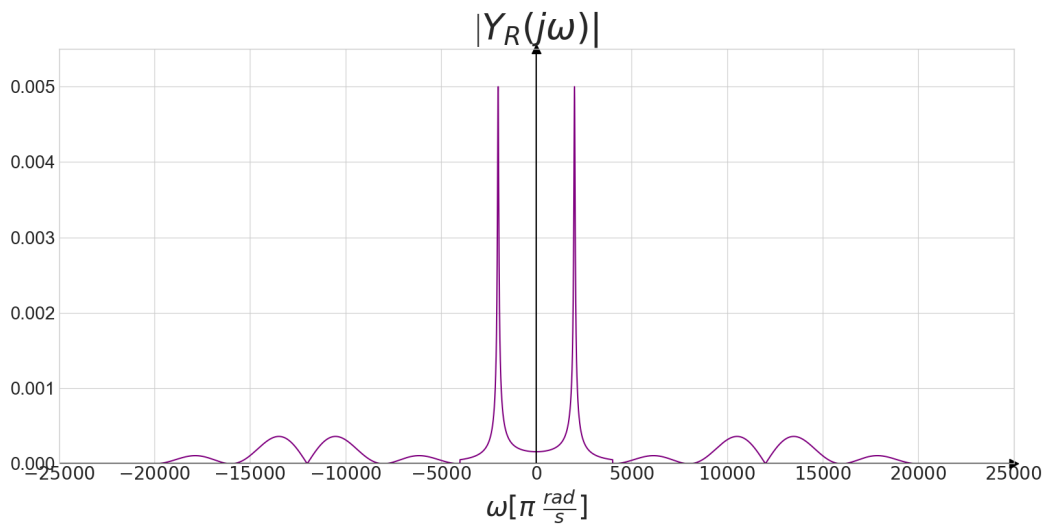
Na sledećoj slici vidimo amplitudski spektar signala $y_T(t)$:



Slika 6: Amplitudski spektar $|Y_T(j\omega)|$

4.5 Signal $y_R(t)$

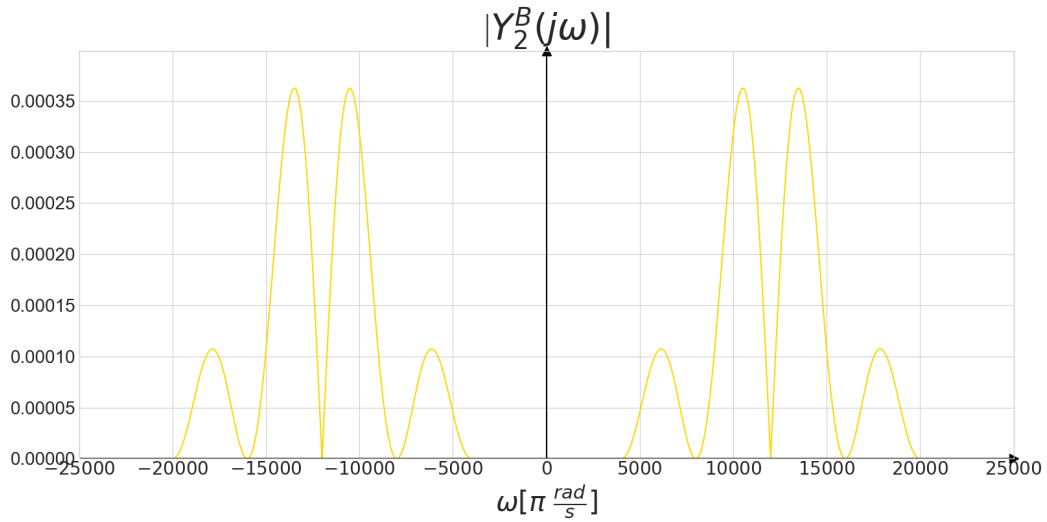
Signal $y_R(t)$ je signal koji dolazi na drugi kraj kanala veze. Pošto je ω_{kv} odabrano tako da ne dolazi do gubitka značajnih učestanosti, očekuje se da će signali $y_R(t)$ imati isti amplitudski spektar kao i signal $y_T(t)$. Na sledećoj slici vidimo ovaj spektar:



Slika 7: Amplitudski spektar $|Y_R(j\omega)|$

4.6 Signal $y_2^b(t)$

Signal $y_2^b(t)$ dobija se izvdajanjem opsega učestanosti na kojima se prenosi modulisani signal $y_2^m(t)$, odnosno $\omega_{o2d} < |\omega| < \omega_{o2g}$, filtrom H_{Rb_2} , iz signala $y_R(t)$. Kako su filtri idealni, ovako filtriran signal biće jednak signalu $y_2^m(t)$. Ovim filtriranjem izvdajaju se učestanosti na kojima se prenosio signal $y_1^n(t)$. Na donjoj slici može se videti amplitudski spektar signala:



Slika 8: Amplitudski spektar $|Y_2^B(j\omega)|$

4.7 Signal $y_2^d(t)$

Demodulacijom signala $y_2^b(t)$ sinusoidom $\cos(2\pi f_c \cdot t)$, istom kao i pri modulaciji, dobijamo signal $y_2^d(t)$. U vremenskom domenu imamo:

$$y_2^d(t) = y_2^b(t) \cdot \cos(2\pi f_c \cdot t) \quad (41)$$

Prelaskom u frekvencijski domen dobijamo:

$$Y_2^D(j\omega) = \frac{1}{2\pi} Y_2^D(j\omega) * \mathcal{F}\{\cos(2\pi f_c t)\}. \quad (42)$$

Sličnim postupkom kao i kod signala $y_2^m(t)$, dobijamo izraz za spektar signala $y_2^d(t)$:

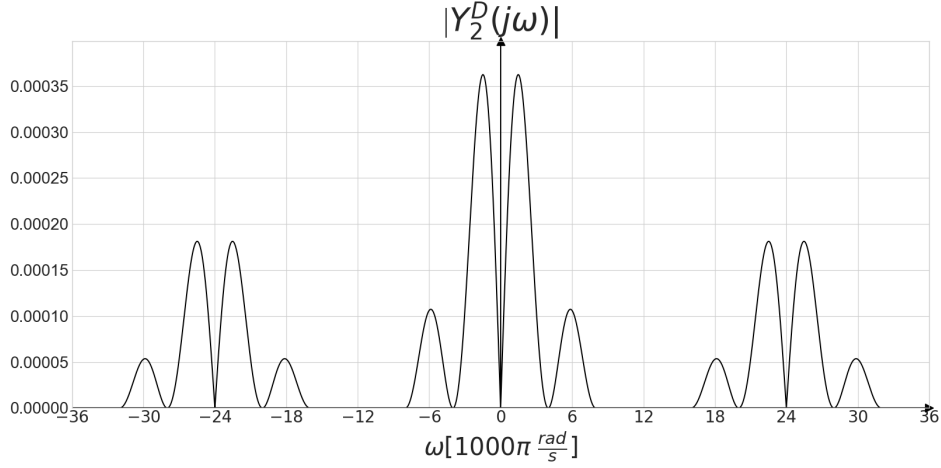
$$Y_2^D(j\omega) = Y_2^N(j\omega) \frac{1}{2} \left[Y_2^N(j(\omega - 4\pi f_c)) + Y_2^N(j(\omega + 4\pi f_c)) \right] \quad (43)$$

Izraz za amplitudski spektar je odavde:

$$|Y_2^D(j\omega)| = \begin{cases} 0 & \omega \in (-\infty, -32000\pi) \\ \frac{1}{4}|Y_2^N(j(\omega + 4\pi f_c))| & \omega \in (-32000\pi, -16000\pi) \\ 0 & \omega \in (-16000\pi, -8000\pi) \\ \frac{1}{2}|Y_2^N(j\omega)| & \omega \in (-8000\pi, 8000\pi) \\ 0 & \omega \in (8000\pi, 16000\pi) \\ \frac{1}{4}|Y_2^N(j(\omega - 4\pi f_c))| & \omega \in (16000\pi, 32000\pi) \\ 0 & \omega \in (32000\pi, +\infty) \end{cases} \quad (44)$$

Iz formule vidimo da pri drugoj modulaciji, odnosno demodulaciji, u spektru vidimo 3 skalirane kopije spektra početnog signala. Kopija centrirana u $\omega = 0 \frac{\text{rad}}{\text{s}}$ ima amplitudu koja jeduplo manja od amplitude početnog signala, i ovo je deo spektra koji ćemo izdvojiti kao izlazni signal $y_2^r(t)$. U spektru takođe postoje još dve skapirane kopije spektra, sa amplitudom koja je četiri puta manja od početne, na učestanostima $\omega = \pm 24000\pi \frac{\text{rad}}{\text{s}}$. Ove dve kopije ćemo filtrom Hd_2r_2 ukloniti iz konačnog signala.

Na slici ispod može se videti amplitudski spektar signala $y_2^d(t)$.



Slika 9: Amplitudski spektar $|Y_2^D(j\omega)|$

4.8 Signal $y_1^r(t)$

Signal $y_1^r(t)$ dobija se izdvajanjem opsega učestanosti $|\omega| < \omega_{i1}$ filtrom H_{Rb_1} iz signala $y_R(t)$. Ovaj opseg predstavlja spektar signala $y_1^n(t)$ koji je prenošen kao deo signala $y_T(t)$. S obzirom na to da su filtri bili idealni, očekujemo da je signal $y_1^r(t)$ identičan kao i signal $y_1^n(t)$. S obzirom na to da je iz prvobitnog signala $y_1(t)$ vršeno filtriranje viših učestanosti filtrom H_{n1} , ove učestanosti nisu prisutne u spektru signala $y_1^r(t)$, što znači da se signali $y_1(t)$ i $y_1^r(t)$ razlikuju. Može se izvesti formula za vremenski oblik signala $y_1^r(t)$:

$$y_1^r(t) = \frac{1}{2\pi} \int_{-\omega_{i1}}^{\omega_{i1}} Y_1^N(j\omega) e^{j\omega t} d\omega, \quad (45)$$

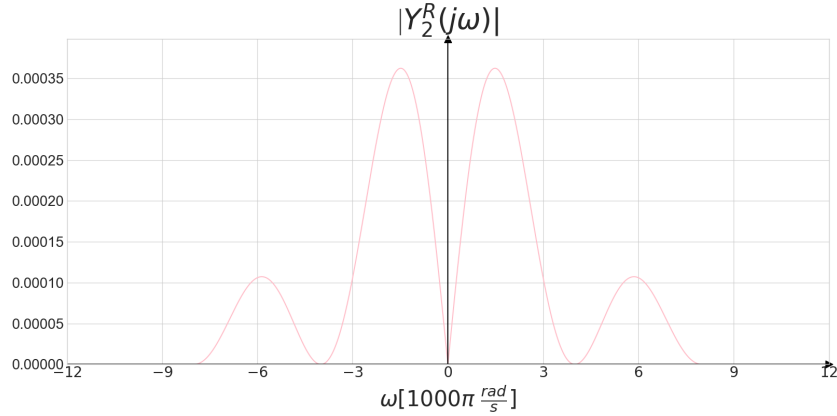
ali kako se integral ne može analitički odrediti, vremenski oblik signala nije prikazan.

4.9 Signal $y_2^r(t)$

Signal $y_2^r(t)$ dobija se propuštanjem signala $y_2^d(t)$ kroz filter $H_{d_2r_2}$. Ovime se izdvajaju skapirane kopije spektra na učestanostima $\pm 4\pi f_c$, nastale modulacijom i demodulacijom. Za rezultujući spektar važi:

$$Y_2^R(j\omega) = \frac{1}{2}Y_2^N(j\omega)|Y_2^R(j\omega)| = \frac{1}{2}|Y_2^N(j\omega)|. \quad (46)$$

Iz prethodno izvedenih jednačina faznih spektara ostalih signala, vidimo da će amplituda prenošenog signala $y_2^r(t)$ biti dva puta manja usled procesa filtriranja, modulacije i demodulacije. Vremenski oblik može se pronaći sličnim postupkom kao i kod signala $y_2^r(t)$, ali je izostavljen jer ni ovaj integral nije analitički rešiv. Na sledećoj slici može se videti amplitudski spektar signala $y_2^r(t)$.



Slika 10: Amplitudski spektar $|Y_2^R(j\omega)|$

5 Rekonstruisani signali

Treba analizirati kako izgledaju signali $y_1^r(t)$ i $y_2^r(t)$ u odnosu na signale koji su početno transmitovani, $y_1(t)$ i $y_2(t)$.

Signal $y_1^r(t)$, u odnosu na signal $y_1(t)$, zadržava amplitudu, ali u njemu, usled filtracija koje su vršene u Sistemu, nedostaju visoke učestanosti. U spektru se ovo oslikava nultim vrednostima amplitude za visoke učestanosti, dok se u vremenu mogu očekivati izobličenja signala.

Signal $y_2^r(t)$ u odnosu na signal $y_2(t)$ ima duplo manju amplitudu. Ovo se dešava usled procesa modulacije i demodulacije signala u okviru Sistema. Kao i kod signala $y_1^r(t)$, i u ovom signalu nedostaju visoke učestanosti, i očekivana su izobličenja u odnosu na početni signal.

6 Kodovi za generisanje grafika

Za crtanje grafika korišćen je programski jezik **Python** i biblioteke *matplotlib* i *numpy*. Za repetativni deo koda napisane su dve klase koje olakšavaju crtanje pojedinačnih karakteristika. Postoje i dva fajla u kojima se nalaze sve funkcije koje su grafički predstavljanje.

Kod kalsa za crtanje funkcija po zadatim vrednostima:

```
1 import matplotlib.ticker as ticker
2 import numpy as np
3 from math import pi
4
5
6 class Graph:
7     def __init__(self, data, values=[], generator=None,
8                 xlabel="", title="",
9                 color="black", ticks_x=ticker.FuncFormatter(
10                     lambda x, pos: f'${(x / pi):g}$')):
11         self.data = data
12         if generator is None:
13             if values is None and len(values) != len(data):
14                 raise ValueError("You need to provide a
15 generator or a value list")
16             self.values = values
17         else:
18             try:
19                 self.values = generator(data)
20             except ValueError:
21                 try:
22                     self.values = np.vectorize(generator)(
23 data)
```

```

20         except ValueError:
21             raise ValueError("Generator was not a
valid funciton")
22
23         self.xlabel = xlabel
24         self.title = title
25         self.color = color
26         self.ticks_x=ticks_x
27
28
29 class MyPlotter:
30     import matplotlib.pyplot as plt
31     font = {'fontname': 'Times New Roman', 'fontsize': 40}
32     style = 'seaborn-whitegrid'
33     plt.style.use(style)
34
35     graphs = []
36
37     lx = 0
38     hx = 0
39     ly = 0
40     hy = 0
41
42     def __init__(self, plot_count=1, resolution=4):
43         self.resolution = resolution
44         self.plot_count = plot_count
45         self.figsize = (self.resolution*4, 2*resolution*self.
plot_count)
46         self.fig, ax = self.plt.subplots(self.plot_count,
47                                         figsize=self.figsize,
48                                         constrained_layout=True)
49
50         self.ax = ax if plot_count > 1 else [ax]
51
52     def add_graph(self, graph):
53         self.graphs.append(graph)
54
55     def prepare(self, divisions=1500*pi):
56         self.lx = min(map(lambda g: min(g.data), self.graphs)
57 )
58         self.hx = max(map(lambda g: max(g.data), self.graphs)
59 )
60         self.ly = min(map(lambda g: min(g.values), self.
graphs))
61         self.hy = max(map(lambda g: max(g.values), self.
graphs))
62         x_ticks1 = np.arange(self.lx, 0, divisions)
63         x_ticks2 = np.arange(0, self.hx, divisions)
64         x_ticks_all = np.concatenate((x_ticks1, x_ticks2),

```



```

axis=None)
63     for i in range(self.plot_count):
64         self.plt.figure(i+1)
65         graph = self.graphs[i]
66         self.ax[i].xaxis.set_major_formatter(graph.
ticks_x)
67         self.ax[i].set(xlim=(self.lx, self.hx),
68                        ylim=(self.ly, self.hy*1.1),
69                        xticks=x_ticks_all)
70         self.ax[i].plot(graph.data, graph.values,
linewidth=1.5,
71                        label=graph.data, color=graph.
color)
72         self.ax[i].set_title(graph.title, fontdict=self.
font)
73         self.ax[i].xaxis.set_tick_params(labelsize=20)
74         self.ax[i].yaxis.set_tick_params(labelsize=20)
75         self.ax[i].axhline(0, color="black")
76         self.ax[i].axvline(0, color="black")
77         self.ax[i].set_xlabel(graph.xlabel, fontdict=self
.font, fontsize=30)
78         self.ax[i].plot(1, 0, ">k", ms=10, transform=self
.ax[i].get_yaxis_transform(), clip_on=False)
79         self.ax[i].plot(0, 1, "^k", ms=10, transform=self
.ax[i].get_xaxis_transform(), clip_on=False)
80
81     def render(self):
82         self.plt.show()

```

Kod 1: drawer.py

Kod pomocnih funkcija:

```

1 from cmath import exp, pi, sin
2
3 j = complex(0, 1)
4
5
6 def ddf(t):
7     epsilon = 10
8     return 1 if abs(t) < epsilon else 0
9
10 # def ddf(t):
11 #     return 1 if t == 0 else 0
12
13
14 def u(t):
15     return 1 if t > 0 else 0
16
17

```

```

18 def sinc(t):
19     return 1 if t == 0 else sin(t) / t
20
21
22 def y1(t):
23     return sin(2000 * pi * t) * exp(-100 * t) * u(t)
24
25
26 def y2(t):
27     return u(t) - 2 * u(t - 5e-4) + u(t - 1e-3)
28
29
30 # def Y1(w):
31 #     return 2000 * pi / (100 ** 2 + 200 * j * w - w ** 2 +
32 #         (2000 * pi) ** 2)
33
34 def Y1(w):
35     return 1/2/j*(1/(100+j*(w-2000*pi))-1/(100+j*(w+2000*pi))
36 )
37
38 def Y2(w):
39     return 4 * 0.00025 ** 2 * w * sinc(0.00025 * w) ** 2
40
41
42 def makeLPF(w):
43     return lambda x: 1 if abs(x) < w else 0
44
45
46 def makeBPF(w1, w2):
47     return lambda x: 1 if w1 < abs(x) < w2 else 0
48
49
50 def modulate(w0):
51     return lambda w: pi*(ddf(w-w0)+ddf(w+w0))

```

Kod 2: functions.py

Kod pomocnih funkcija - signala:

```

1 from functions import *
2 import numpy as np
3
4 fc = 6000
5 wc = 2*pi*fc
6 Hn1 = makeLPF(4000*pi)
7 Hn2 = makeLPF(8000*pi)
8 Hkv = makeLPF(20000*pi)
9 HRb1 = makeLPF(4000*pi)
10 HRb2 = makeBPF(4000*pi, 8000*pi)

```

```

11 Hd2r2 = makeLPF(8000*pi)
12
13 y1 = np.vectorize(y1)
14 y2 = np.vectorize(y2)
15 Y1 = np.vectorize(Y1)
16 Y2 = np.vectorize(Y2)
17 Y1N = np.vectorize(lambda w: Y1(w)*Hn1(w))
18 Y2N = np.vectorize(lambda w: Y2(w)*Hn2(w))
19 Y2M = np.vectorize(lambda w: (Y2N(w-wc)+Y2N(w+wc))/2)
20 YT = np.vectorize(lambda w: Y1N(w) + Y2M(w))
21 YR = YT
22 Y2B = Y2M
23 def Y2D(w):
24     return 1/4*Y2N(w-2*wc) + 1/2*Y2N(w) + 1/4*Y2N(w+2*wc)
25 sinc = np.vectorize(sinc)
26
27
28 def modulateVEC(w0):
29     return np.vectorize(modulate(w0))
30
31
32 def integrateInFreq(f,data,low,high,step=0.0001):
33     result = np.array([])
34     t = low
35     progressx = 1
36     while t <= high:
37         if t/high*100 >= progressx:
38             print(f"{progressx}% done")
39             progressx += 1
40         s = 0
41         for w in data:
42             s += f(w,t)
43         result = np.append(result, s*(data[-1]-data[0])/len(
data))
44         t += step
45     return result

```

Kod 3: signals.py

Kod za crtanje amplitudske karakteristike $Y_1(j\omega)$:

```

1 import numpy as np
2 import functions as sig
3 from math import pi
4 from drawer import Graph, MyPlotter
5
6 w_range, dt = 6000, 1
7 data = np.arange(-w_range * pi, w_range * pi + dt, dt)
8
9

```

```

10 g1 = Graph(data, None, lambda x: abs(sig.Y1(x)),
11           xlabel=r"$\omega [ \pi \ \frac{rad}{s} ]$",
12           title=r"$|Y_1(j\omega)|$", color="red")
13
14 plotter = MyPlotter(plot_count=1)
15 plotter.add_graph(g1)
16 plotter.prepare()
17 plotter.render()

```

Kod 4: AmpY1.py

Kod za crtanje amplitudske karakteristike $Y_2(j\omega)$:

```

1 import numpy as np
2 import functions as sig
3 from math import pi
4 from drawer import Graph, MyPlotter
5
6 w_range, dt = 12000, 1
7 data = np.arange(-w_range * pi, w_range * pi + dt, dt)
8
9
10 g1 = Graph(data, None, lambda x: abs(sig.Y1(x)), xlabel=r"$\omega [ \pi \ \frac{rad}{s} ]$",
11           title=r"$Y_1(j\omega)$", color="red")
12 g2 = Graph(data, None, lambda x: abs(sig.Y2(x)), xlabel=r"$\omega [ \pi \ \frac{rad}{s} ]$",
13           title=r"$Y_2(j\omega)$", color="green")
14
15 plotter = MyPlotter(plot_count=1)
16 plotter.add_graph(g2)
17 plotter.add_graph(g1)
18 plotter.prepare(3000*pi)
19 plotter.render()

```

Kod 5: AmpY2.py

Kod za crtanje amplitudske karakteristike $Y_1^N(j\omega)$:

```

1 import numpy as np
2 import signals as sig
3 from math import pi
4 from drawer import Graph, MyPlotter
5
6 w_range, dt = 12000, 1
7 data = np.arange(-w_range * pi, w_range * pi + dt, dt)
8
9
10 g1 = Graph(data, None, lambda x: np.abs(sig.Y1N(x)), xlabel=r"$\omega [ \pi \ \frac{rad}{s} ]$",
11           title=r"$|Y_1^N(j\omega)|$", color="red")

```

```

12
13 plotter = MyPlotter(plot_count=1)
14 plotter.add_graph(g1)
15 plotter.prepare(3000*pi)
16 plotter.render()

```

Kod 6: AmpY1N.py

Kod za crtanje amplitudske karakteristike $Y_2^N(j\omega)$:

```

1 import numpy as np
2 import signals as sig
3 from math import pi
4 from drawer import Graph, MyPlotter
5
6 w_range, dt = 12000, 1
7 data = np.arange(-w_range * pi, w_range * pi + dt, dt)
8
9
10 g2 = Graph(data, None, lambda x: abs(sig.Y2N(x)), xlabel=r"$\omega [ \pi \ \frac{rad}{s} ]$",
11           title=r"|Y_2^N(j\omega)|", color="green")
12
13 plotter = MyPlotter(plot_count=1)
14 plotter.add_graph(g2)
15 plotter.prepare(3000*pi)
16 plotter.render()

```

Kod 7: AmpY2N.py

Kod za crtanje amplitudske karakteristike $Y_2^M(j\omega)$:

```

1 import numpy as np
2 import signals as sig
3 from cmath import pi, exp
4 from drawer import Graph, MyPlotter
5
6 w_range, dt = 25000, 10
7 data = np.arange(-w_range * pi, w_range * pi + dt, dt)
8
9 g1 = Graph(data, generator=sig.Y2M,
10           xlabel=r"$\omega [ \pi \ \frac{rad}{s} ]$",
11           title=r"|Y_2^M(j\omega)|", color="orange")
12
13 plotter = MyPlotter(plot_count=1)
14 plotter.add_graph(g1)
15 plotter.prepare(5000*pi)
16 plotter.render()

```

Kod 8: AmpY2M.py

Kod za crtanje amplitudske karakteristike $Y_T(j\omega)$:

```

1 import numpy
2 import numpy as np
3 import signals as sig
4 from cmath import pi, exp
5 from drawer import Graph, MyPlotter
6
7 w_range, dt = 25000, 10
8 data = np.arange(-w_range * pi, w_range * pi + dt, dt)
9
10 g1 = Graph(data, generator=lambda x: abs(sig.YT(x)),
11            xlabel=r"$\omega$ [ \pi \ \frac{rad}{s} ] $",
12            title=r"|$Y_T(j\omega)$|$", color="blue")
13
14 plotter = MyPlotter(plot_count=1)
15 plotter.add_graph(g1)
16 plotter.prepare(5000*pi)
17 plotter.render()

```

Kod 9: AmpYT.py

Kod za crtanje amplitudske karakteristike $Y_R(j\omega)$:

```

1 import numpy
2 import numpy as np
3 import signals as sig
4 from cmath import pi, exp
5 from drawer import Graph, MyPlotter
6
7 w_range, dt = 25000, 10
8 data = np.arange(-w_range * pi, w_range * pi + dt, dt)
9
10 g1 = Graph(data, generator=lambda w: abs(sig.YR(w)),
11            xlabel=r"$\omega$ [ \pi \ \frac{rad}{s} ] $",
12            title=r"|$Y_R(j\omega)$|$", color="purple")
13
14 plotter = MyPlotter(plot_count=1)
15 plotter.add_graph(g1)
16 plotter.prepare(5000*pi)
17 plotter.render()

```

Kod 10: AmpYR.py

Kod za crtanje amplitudske karakteristike $Y_2^B(j\omega)$:

```

1 import numpy as np
2 from cmath import exp, pi
3 import matplotlib.ticker as ticker
4 import signals as sig
5 from drawer import Graph, MyPlotter
6
7 j = complex(0, 1)

```

```

8
9 w_range, dw = 25000, 10
10 data = np.arange(-w_range*pi, w_range*pi + dw, dw)
11
12 g1 = Graph(data, generator=lambda t: abs(sig.Y2B(t)),
13           xlabel=r"$\omega$ [ \pi \ \frac{rad}{s} ] $",
14           title=r"|$Y_2^B(j\omega)$|$", color="gold")
15
16 plotter = MyPlotter(plot_count=1)
17 plotter.add_graph(g1)
18 plotter.prepare(5000*pi)
19 plotter.render()

```

Kod 11: AmpY2B.py

Kod za crtanje amplitudske karakteristike $Y_2^D(j\omega)$:

```

1 import numpy as np
2 from cmath import exp, pi
3 import matplotlib.ticker as ticker
4 import signals as sig
5 from drawer import Graph, MyPlotter
6
7 j = complex(0, 1)
8
9 w_range, dw = 36000, 10
10 data = np.arange(-w_range*pi, w_range*pi + dw, dw)
11
12 g1 = Graph(data, generator=lambda t: abs(sig.Y2D(t)),
13           xlabel=r"$\omega$ [ 1000\pi \ \frac{rad}{s} ] $",
14           title=r"|$Y_2^D(j\omega)$|$", color="gold",
15           ticks_x=ticker.FuncFormatter(lambda x, pos: f'${(x
16           /1000/ pi):g}$'))
17
18 plotter = MyPlotter(plot_count=1)
19 plotter.add_graph(g1)
20 plotter.prepare(6000*pi)
21 plotter.render()

```

Kod 12: AmpY2D.py

Kod za crtanje amplitudske karakteristike $Y_2^R(j\omega)$:

```

1 import numpy as np
2 from cmath import exp, pi
3 import matplotlib.ticker as ticker
4 import signals as sig
5 from drawer import Graph, MyPlotter
6
7 j = complex(0, 1)
8

```

```

9 w_range, dw = 12000, 10
10 data = np.arange(-w_range*pi, w_range*pi + dw, dw)
11
12 g1 = Graph(data, generator=lambda w: abs(sig.Y2D(w))*sig.
    Hd2r2(w),
13             xlabel=r"$\omega [ 1000\pi \ \frac{\text{rad}}{\text{s}} ]$",
14             title=r"|$Y_2^R(j\omega)|$", color="pink",
15             ticks_x=ticker.FuncFormatter(lambda x, pos: f'${(x
    /1000/ pi):g}$'))
16
17 plotter = MyPlotter(plot_count=1)
18 plotter.add_graph(g1)
19 plotter.prepare(3000*pi)
20 plotter.render()

```

Kod 13: AmpY2R.py

Lista Kodova

1	drawer.py	22
2	functions.py	24
3	signals.py	25
4	AmpY1.py	26
5	AmpY2.py	27
6	AmpY1N.py	27
7	AmpY2N.py	28
8	AmpY2M.py	28
9	AmpYT.py	29
10	AmpYR.py	29
11	AmpY2B.py	29
12	AmpY2D.py	30
13	AmpY2R.py	30