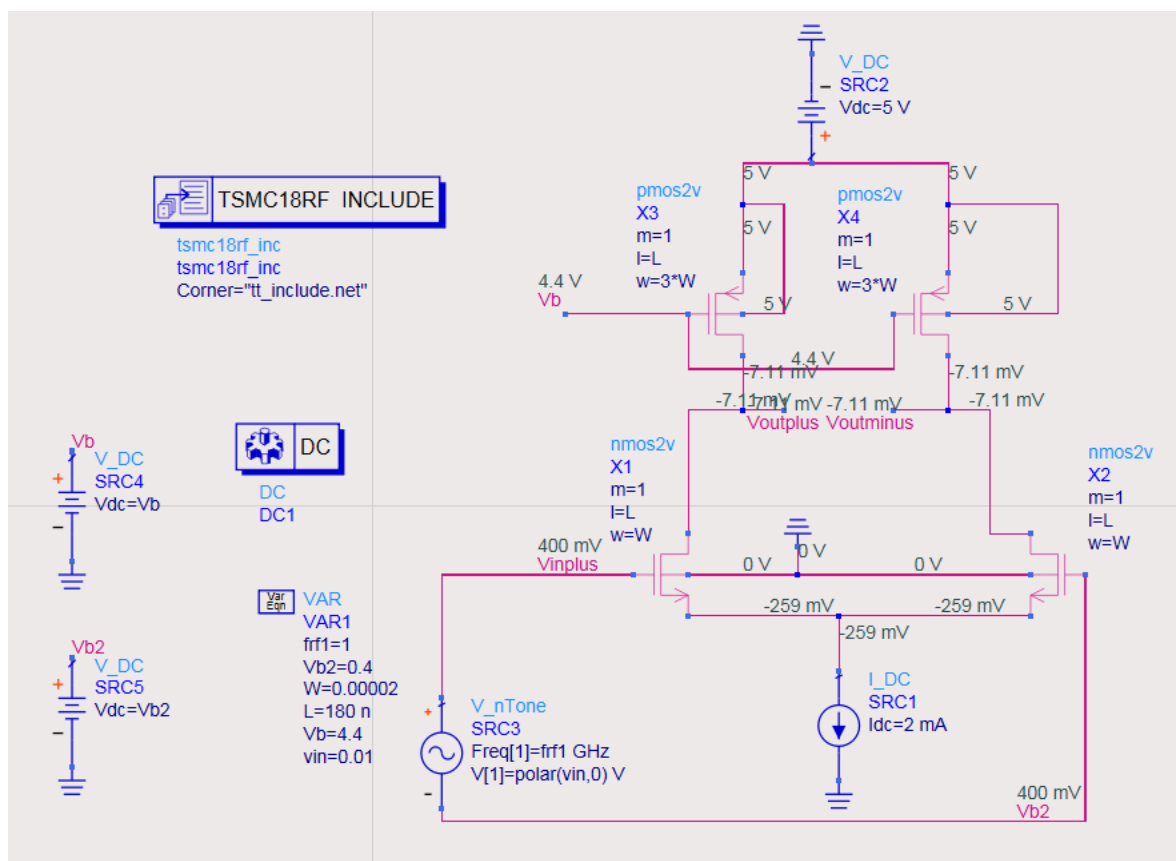


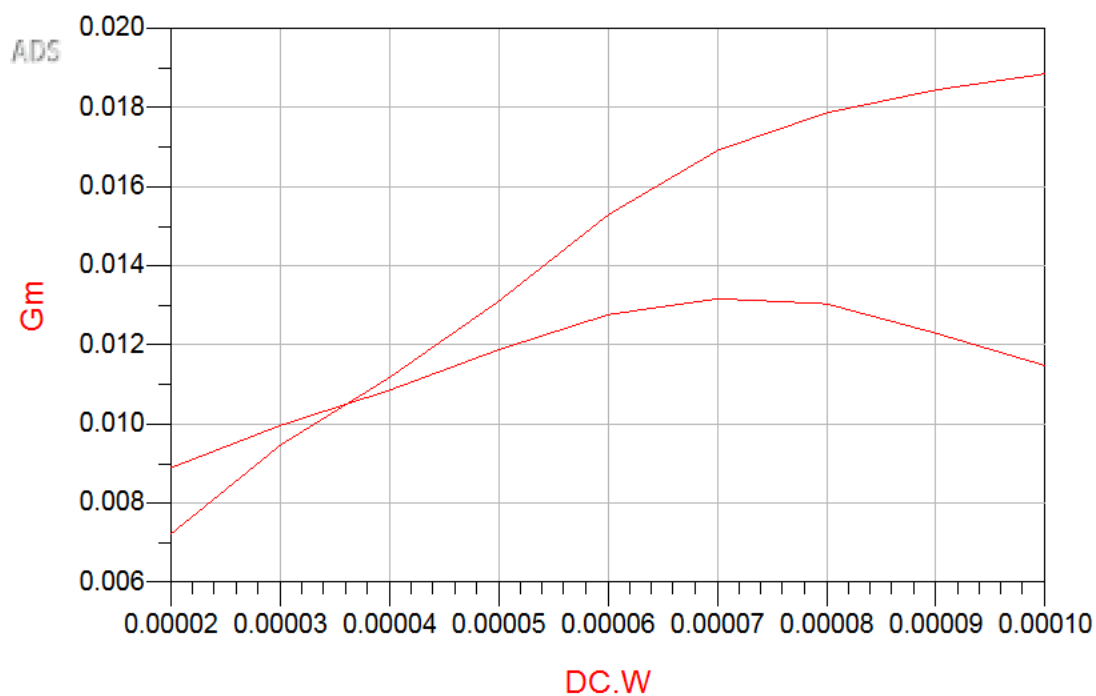
به نام خدا

مبین خطیب - ۹۹۱۰۶۱۱۴ - تمرین سوم (کامپیوتری اول) مدار های مخبراتی

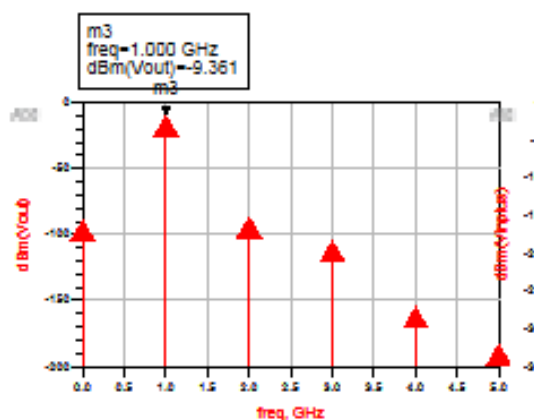
-۱



ابتدا با قرار دادن dc component سعی میکنیم تا بتوانیم نیاز های خواسته شده سوال را $G_m = 10 \text{ Ms}$ در حدود این مقدار رساندیم هرچهار ترانزیستور این G_m رامیت میکنند که این کار را با تغییر بایاس های ورودی ترانزیستور های nmos یعنی $V_{b2} = 0.2 \text{ V}$ و همچنین ترانزیستور های pmos در حدود $V_b = 4.38 \text{ V}$ تنظیم کرده ایم. با تنظیم W, I, V_b, V_{b2} مناسب تنظیم کنیم در اینجا G_m های ترانزیستور ها را بر حسب W رسم کرده ایم و در پایین میبینیم که $W = 36 \mu$ باشد مقدار G_m نیز برای ترانزیستور ها خوب tune شده است همچنین قابل ذکر است که نمودار های ترانزیستور ها دو به دو روی هم افتاده اند:

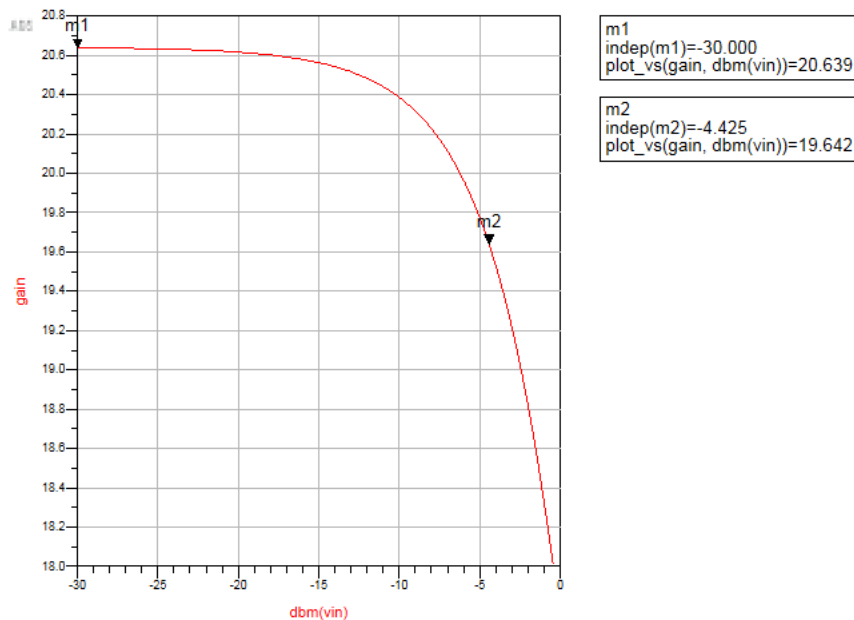


حال از تنظیم حالت روی DC خارج میشویم و به حالت TSMC18RF برمیگردیم برای بررسی P1db. با توجه به توضیحات ویدیو خواهیم داشت:



در فرکانس 1GHz که فرکانس ورودی ما میباشد $V_{out} = -9.361\text{dbm}$ میباشد.

حال با سوییپ کردن ورودی نمودار های مختلف و همچنین گین و در نتیجه آن P1db compression point را خواهیم دید:

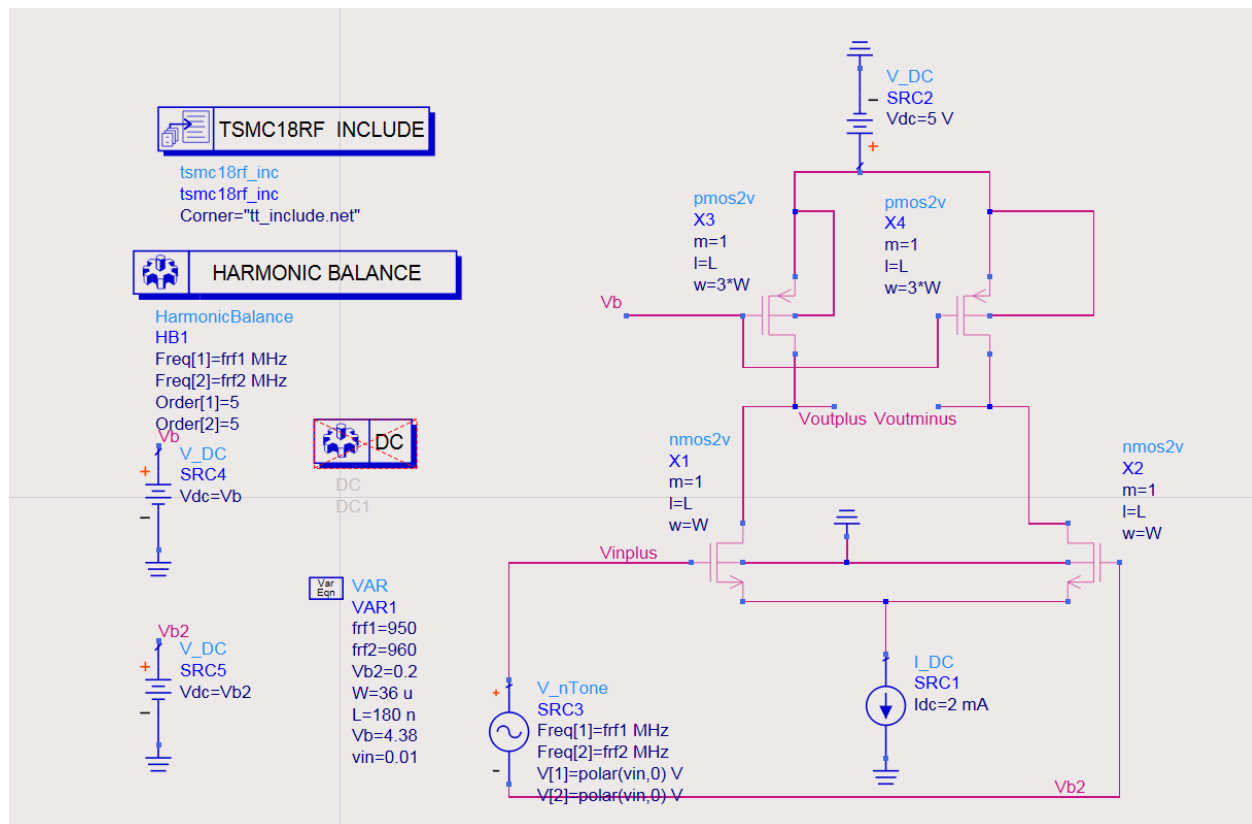


با توجه به نمودار بالا:

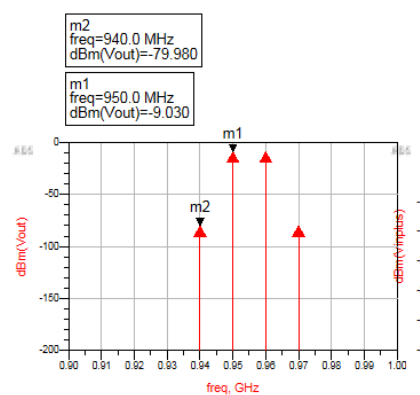
بهره دیفرانسیلی که در همان -30 dbm می باشد برابر است با : 20.639

P1db compression point = - 4.425 dbm

این نمودار ما به ازای $v_{in} = 0.01$ رسم شده است. همانطور که می بینیم در حدود 9db این مولفه اصلی تقویت شده است که باز گین آن را جداگانه در پایین خواهیم دید. اما همانطور که می بینیم در فرکانس های 2w,3w و.. هم مولفه های فرکانسی داریم:



در اینجا با توجه به توضیحات ویدیو ورودی ها و مقادیر خواسته شده را رسم کرده ایم:



IIP3:

تن های اصلی در وسط و تن های فرعی در کناره ها هستند که 10kHz با هم فاصله دارند در اینجا $vin = 0.01$ است. فاصله آن ها را هم با استفاده از m1,m2 مشخص کرده ایم.

Eqn $V_{in}=0.01$
 Eqn $P_{in} = dbm(V_{in})$
 Eqn $\Delta P = m1 - m2$
 Eqn $IIP3 = \Delta P/2 + P_{in}$

Eqn $V_{out} = V_{outplus} - V_{outminus}$

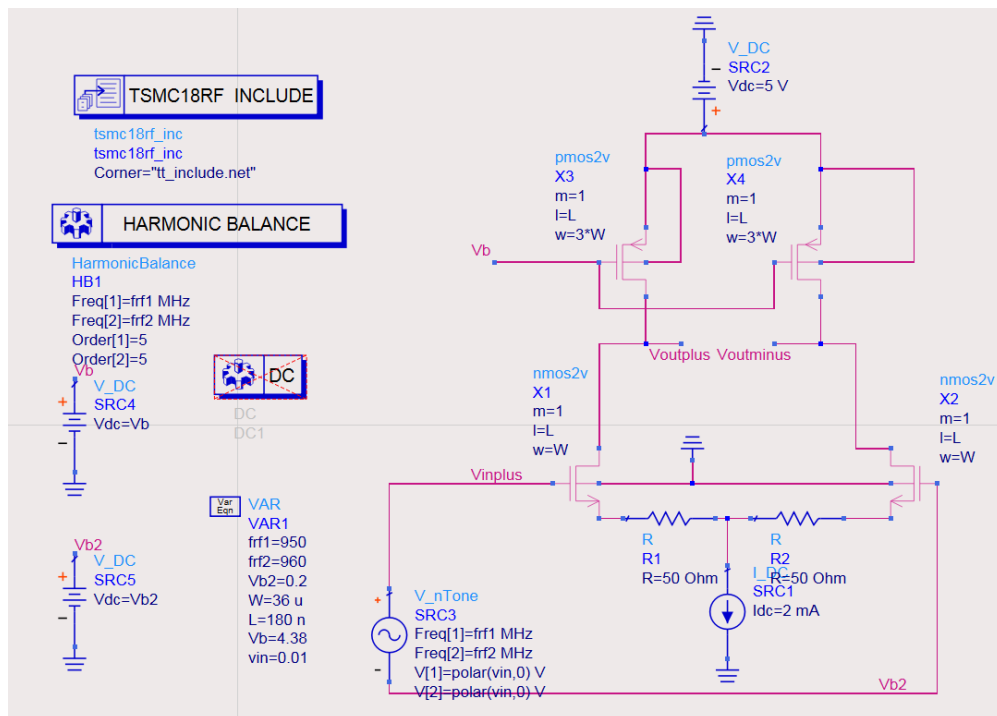
Eqn $gain = db(V_{out}[1]) - db(V_{inplus}[1])$

freq	IIP3
<invalid>	5.475

با تنظیم equation ها به صورت بالا میتوانیم ببینیم که مقدار $IIP3=5.475$ است که مقدار خوبی است.

(ب)

برای بهبود خطینگی میتوان از مقاومت در سورس ترانزیستور های nmos استفاده کرد چون مقداری از ولتاژ ورودی روی یک المان خطی می افتد.



P1db تغییر چندانی نداشت اما:

Eqn $V_{in}=0.01$

Eqn $P_{in} = \text{dbm}(V_{in})$

Eqn $\Delta P = m1 - m2$

Eqn $IIP3 = \Delta P/2 + P_{in}$

Eqn $V_{out} = V_{outplus} - V_{outminus}$

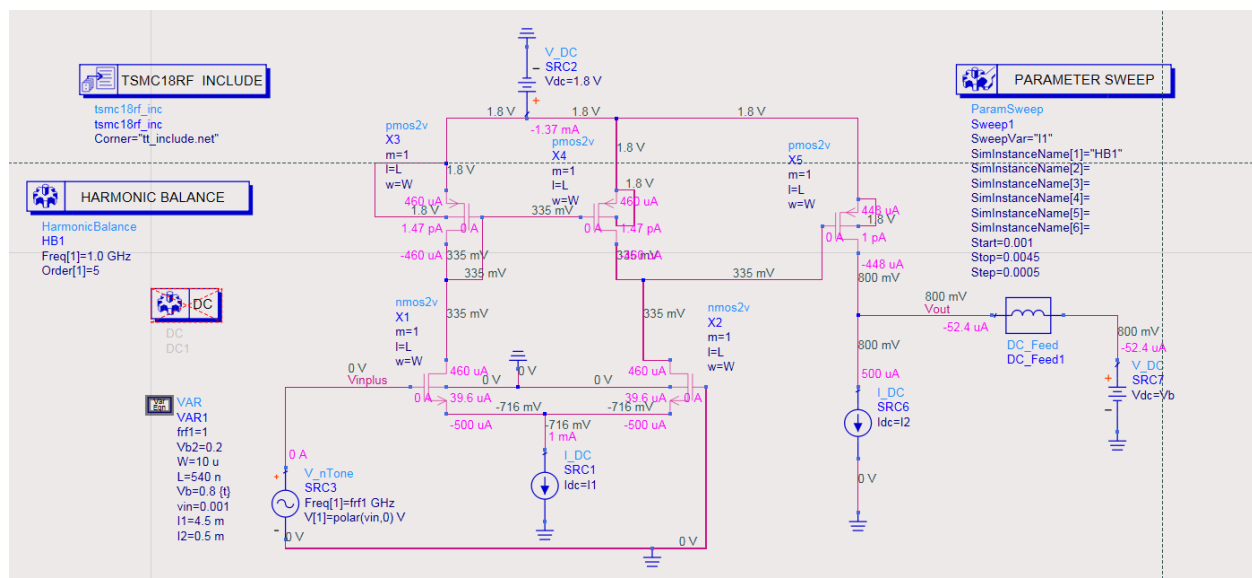
Eqn $\text{gain} = \text{db}(V_{out}[1]) - \text{db}(V_{inplus}[1])$

freq	IIP3
<invalid>	11.075

همانطور که مشاهده میشود IIP3 افزایش یافت ولی 1dB compression point تغییر چندانی نداشت زیرا دامنه مؤلفه های مزاحم در V_{in} های کوچک بسیار کوچک تر شد که نشان میدهد خطیگی مدار افزایش یافته است.

پ) همانطور که مشاهده میشود بهره به دلیل اینکه همه ولتاژ ورودی دو سر گیت سورس نمی افتد کاهش یافت.

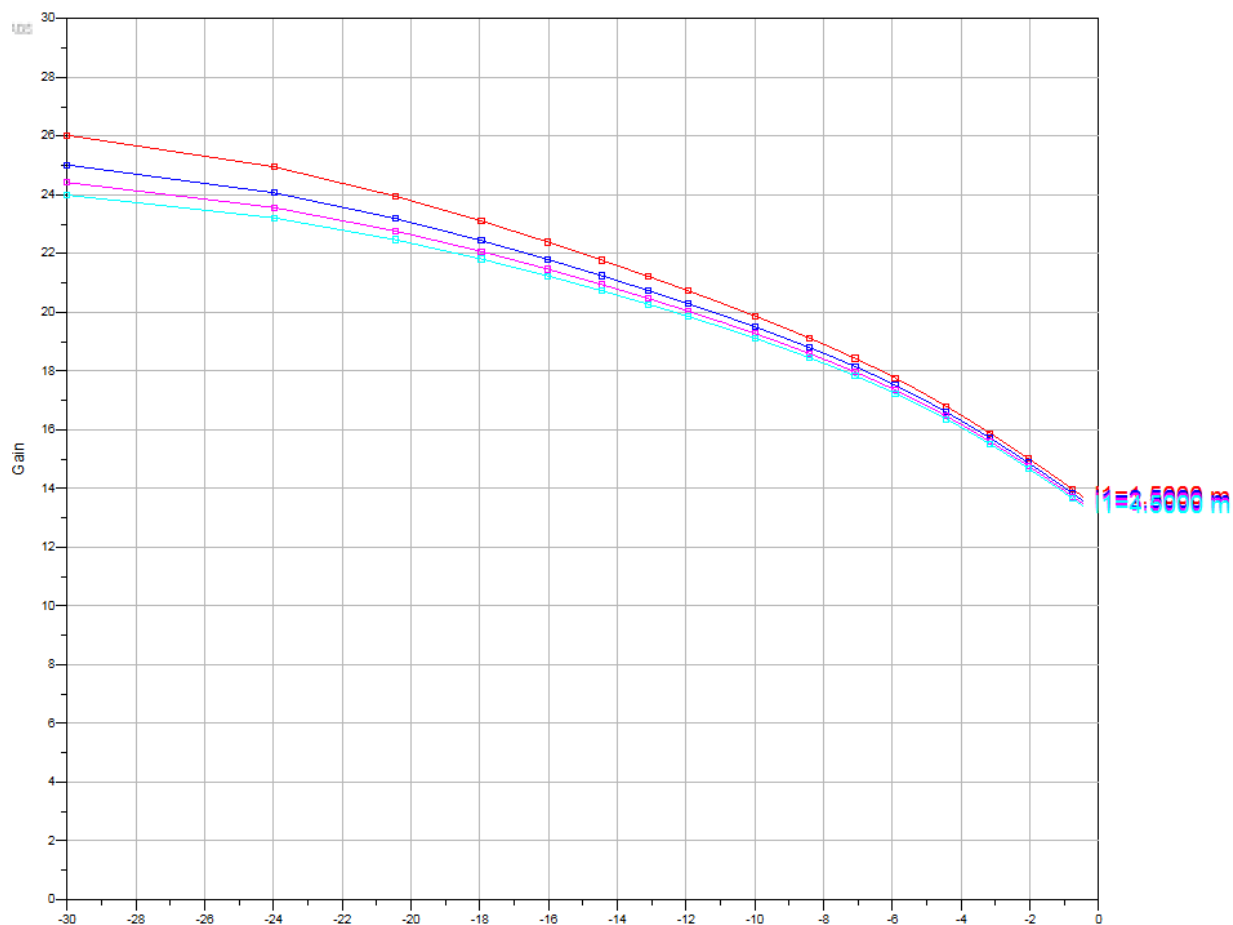
-۲



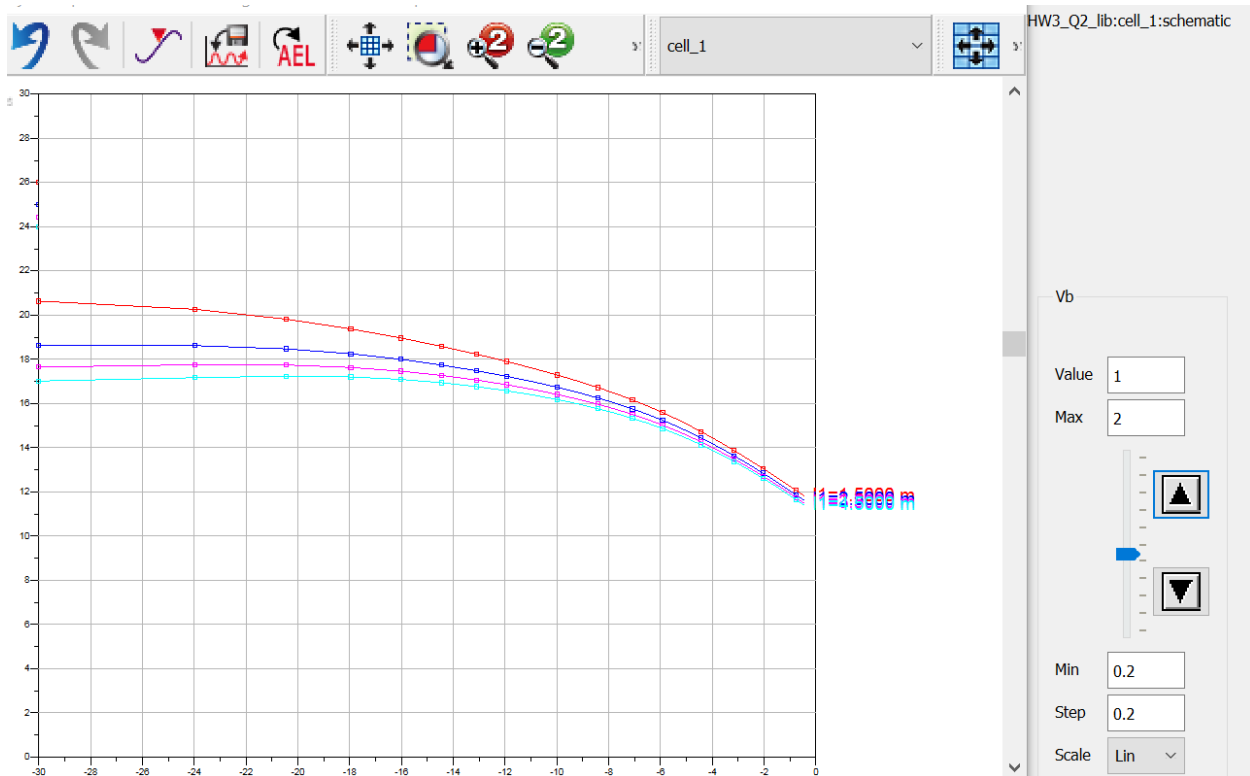
در اینجا برای اینکه بتوانیم به حداکثر 1db compression point برسیم سه متغیر داریم که میتوانند این را تعیین کنند یکی جریان مصرفی کل دو تقویت کننده که در اینجا با دو جریان ($I_1 + I_2 = 5\text{mA}$) نمایش داده ایم و دیگری V_b که در انتهای DC_Feed قرار گرفته است. باید به نحوی باشد که از DC_Feed جریانی نگذرد اینگونه است که جریان I_2 در بالا مشخص میشود. مقدار جریان را آنقدر بالا پایین کردیم تا بهترین مقداری که میتوانستیم برای آن قرار دهیم $I_2 = 0.5\text{mA}$ بود. حالا جریان I_1 را با توجه به شرط سوال سوییچ میکنیم تا ببینیم به اعضای کدام حالت بیشترین P1db compression point را خواهیم داشت.

چندین نمودار را رسم میکنیم تا حالات مختلف را ببینیم همچنین در این حال V_b را نیز با مقادیر مختلف را tune میکنیم:

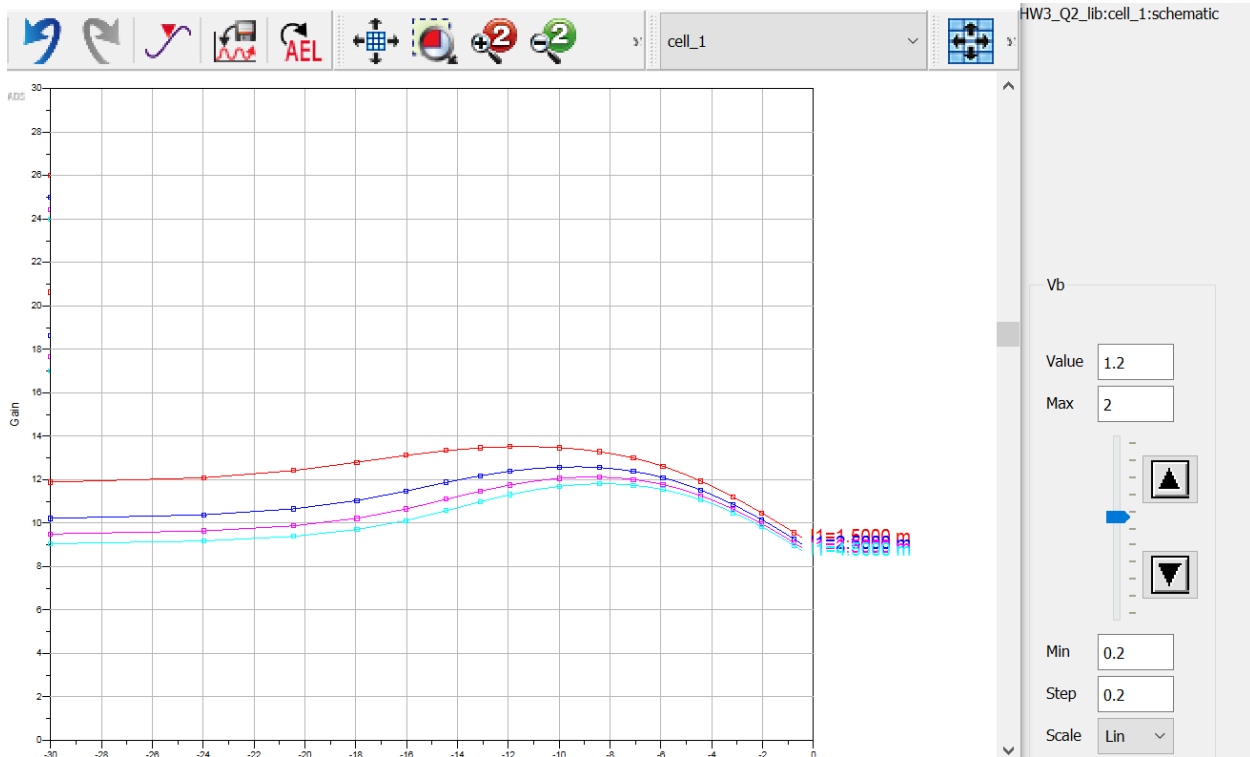
به ازای $V_b = 0.8\text{V}$ داریم:



به ازای $V_b = 1\text{V}$ داریم:



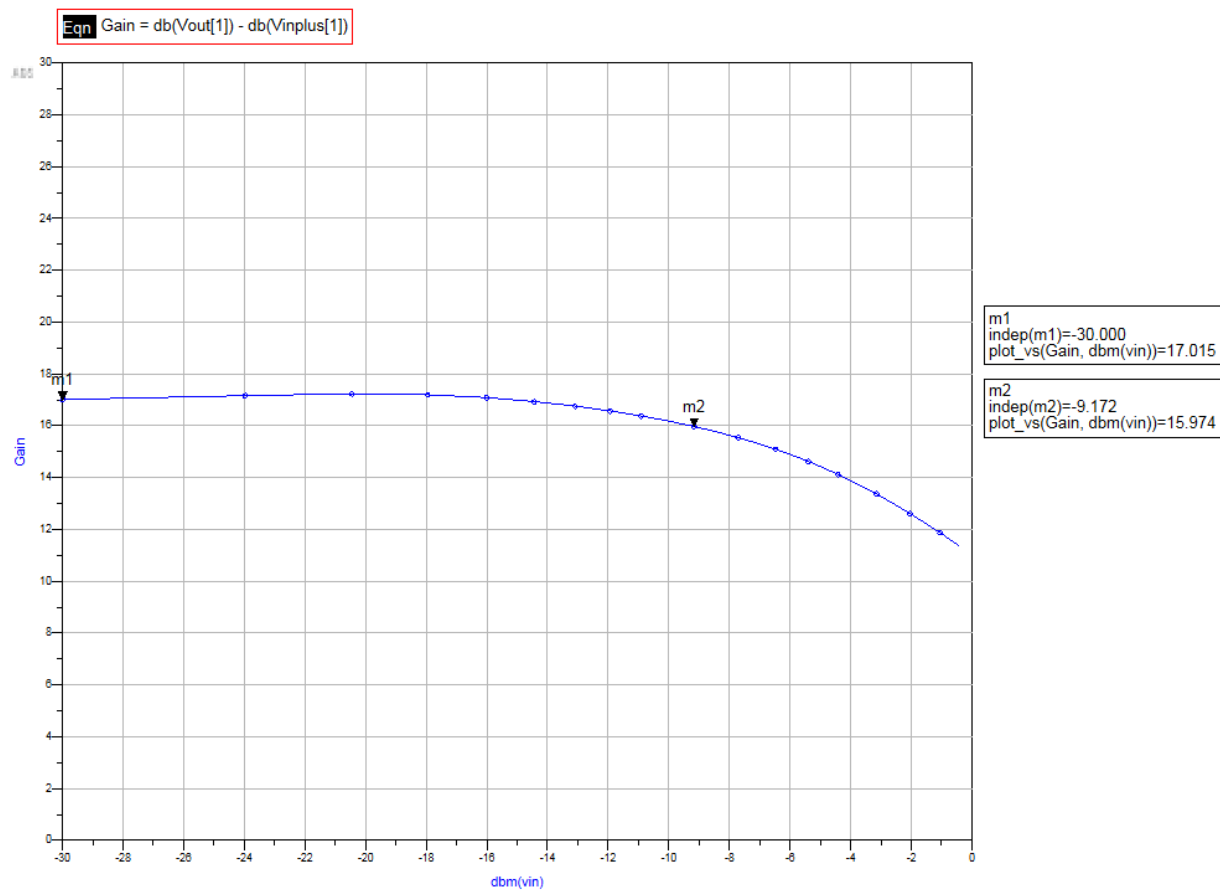
به ازای $V_b = 1.2 \text{ V}$ داریم:



همانطور که میبینیم به ازای $V_b = 1.2V$ نمودار از حالت خطی خود خارج میشود. و بنابراین بهترین حالتی که برای P1db compression point داشتیم همان به ازای $V_b = 1V$ بود. بنابراین مقادیر خود را به این صورت تنظیم میکنیم:

$$V_b = 1V, I_1 = 4.5mA, I_2 = 0.5mA$$

(ب)



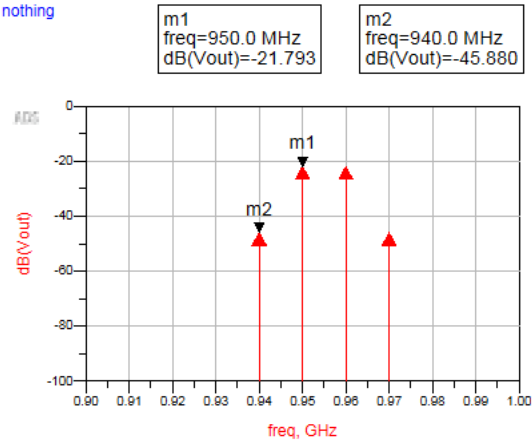
همان طور که میبینیم:

بهره دیفرانسیلی در حدود $\text{Gain} = 17$ میباشد

P1db compression point = -9.17dbm

حال برای IIP3 مانند سوال قبل نمودار را تغییر میدهیم و خواهیم داشت:

nothing



حال که این را رسم کردیم (دو فرکانس که داده بودیم 950MHz,960MHz) بودند، در نهایت:

$$\text{Eqn } \text{Pin} = \text{dbm}(0.01)$$

$$\text{Eqn } \text{deltaP} = m1 - m2$$

$$\text{Eqn } \text{IIP3} = \text{deltaP}/2 + \text{Pin}$$

$$\text{Eqn } \text{gain} = \text{db}(\text{Vout}[1]) - \text{db}(\text{Vinplus}[1])$$

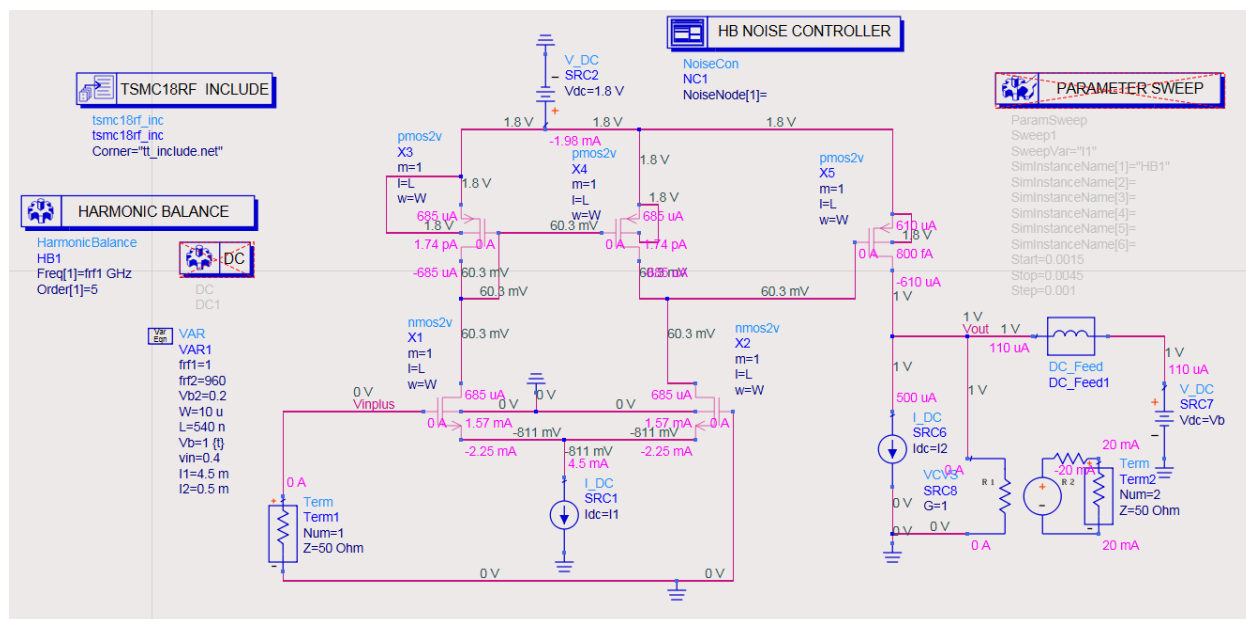
freq	IIP3
<invalid>	-17.956

مقدار $\text{IIP3} = -17.956$ شد.

اگر مقدار vin را تغییر دهیم مقادیر بهتری نیز برای IIP3 لحاظ خواهد شد که این خواسته سوال نبوده است.

(پ)

طبق توضیحات ویدیو خواهیم داشت:



مقدار Noise figure را به ازای فرکانس های مختلف در پایین رسم کرده ایم که البته فرکانس مد نظر ما فرکانس 1GHz بوده است که $nf(2) = 15.822$ میباشد.

noisefreq	nf(2)	NFssb
1.000 GHz	15.822	15.822
2.000 GHz	19.077	19.077
3.000 GHz	22.830	22.830
4.000 GHz	26.343	26.343
5.000 GHz	29.453	29.453
6.000 GHz	32.184	32.184
7.000 GHz	34.600	34.600
8.000 GHz	36.762	36.762
9.000 GHz	38.721	38.721
10.00 GHz	40.519	40.519

-۳

گفته شده که $\text{bitrate} = 1\text{kbit/sec}$ میباشد بنابراین مد نظرمان در متلب را به صورت از صفر تا ۰.۰۰۱ ثانیه در فرستنده در نظر میگیریم.

به خاطر اینکه باید برای ارسال داده باینری 1 از فرکانس 99MHz و برای ارسال داده باینری 0 از فرکانس 101MHz استفاده کنیم، این یعنی اینکه نرخ نمونه برداری باید بسیار بالا باشد، و درعین حال حداقل مثلاً قرار است 10 Bit را ارسال کنیم تا نتایج را ببینیم اما علی الحساب $N_{\text{Send}} = 20$ قرار میدهیم تا ببینیم نتیجه چه میشود و در نهایت اعمال خواهیم کرد.

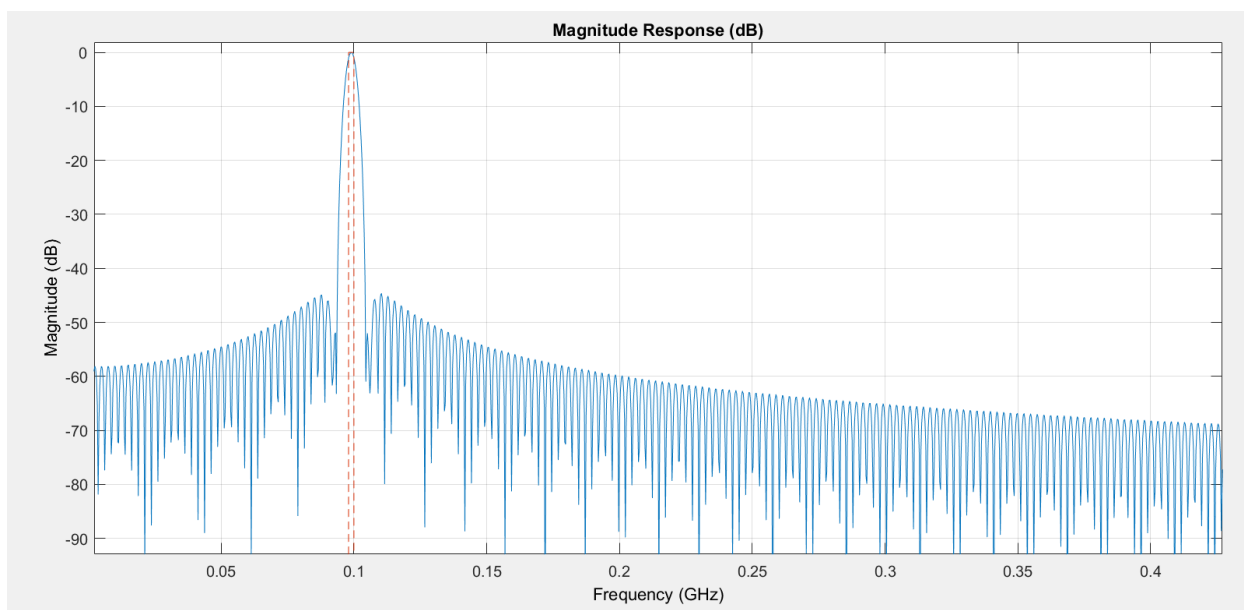
اگر مثلاً برای فرکانس 101MHz نگاه کنیم که در هر دوره تناوب بخواهیم ۵۰ نقطه نمونه برداری کنیم نرخ نمونه برداری 5050MHz خواهد شد!!

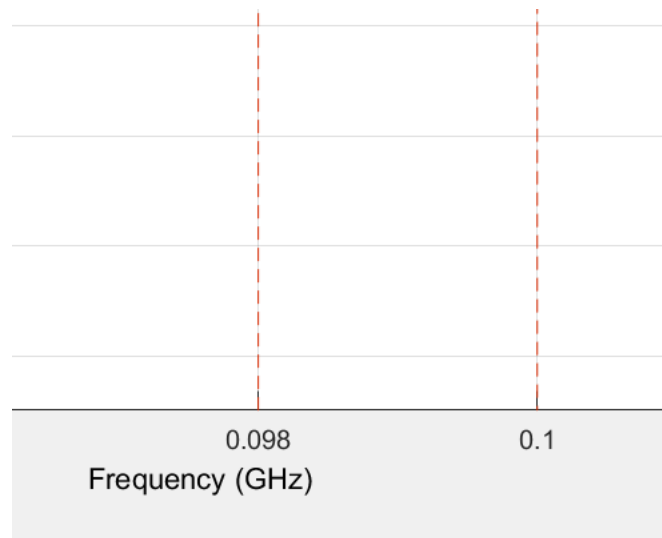
حال دو فیلتر FIR باند گذر با استفاده از تابع 'designfilt' از Signal Processing Toolbox طراحی شده اند. این فیلترها طوری طراحی شده اند که فقط فرکانس هایی را در محدوده خواسته شده سوال ما یعنی حول دو فرکانس بیان شده طراحی شده اند

BandPassFilter1: طراحی شده با فرکانس های قطع ۱۰۰ مگاهرتز و ۱۰۲ مگاهرتز.

BandPassFilter2: با فرکانس های قطع ۹۸ مگاهرتز و ۱۰۰ مگاهرتز طراحی شده است.

حالا اندازه پاسخ فرکانسی را رسم میکنیم که به شکل زیر است:

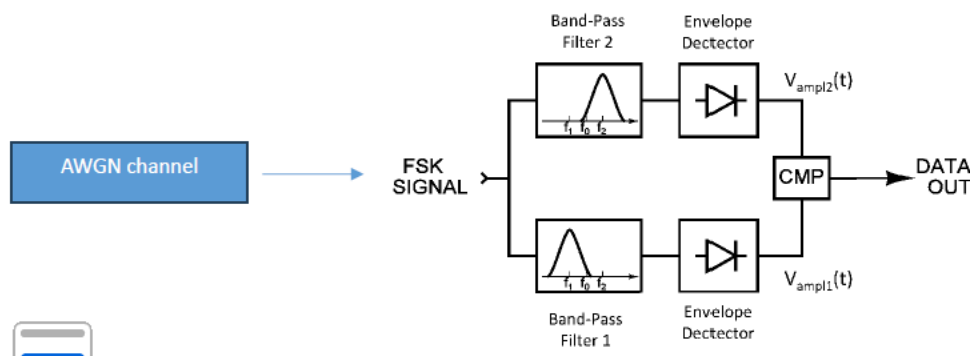




همانطور که میبینیم در حدود فرکانس هایی که تعریف کردیم فیلتر ها قرار داده شده اند و فیلتر 99MHz را که مثلا میبینیم که در فرکانس دیگر میتواند با توجه به مقدار آن آن را تقریبا حذف کند.

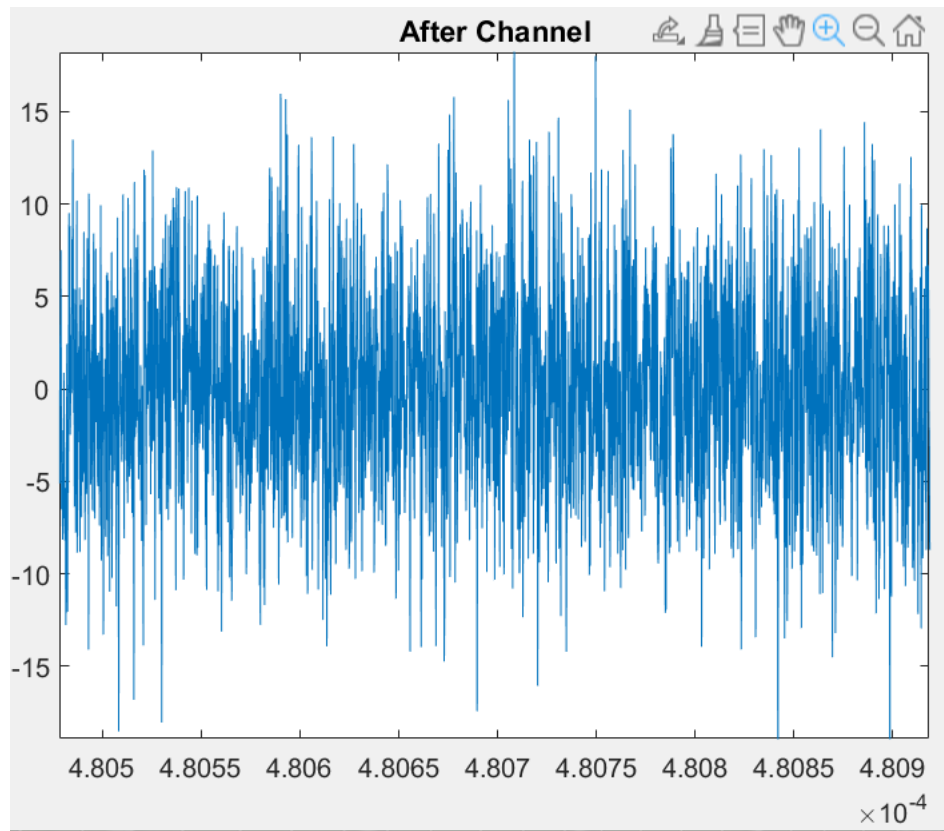
الگوریتم کلی ما به این شکل است که بیت ها را یکی یکی میفرستیم (که این بیت ها را به صورت تصادفی تولید کرده ایم در یک for ساده این کار را انجام میدهم) سپس با توجه به مقدار آن فرکانس مد نظر را ست میکنیم. حال باید سیگنال را از کانال رد کنیم. نویز را اضافه میکنیم (نویز سفید ادتیو) بعد هم که SNR را تنظیم میکنیم:

شکل سیگنال ورودی که واضح است حال بقیه مراحل این شکل پایین را رسم خواهیم کرد:

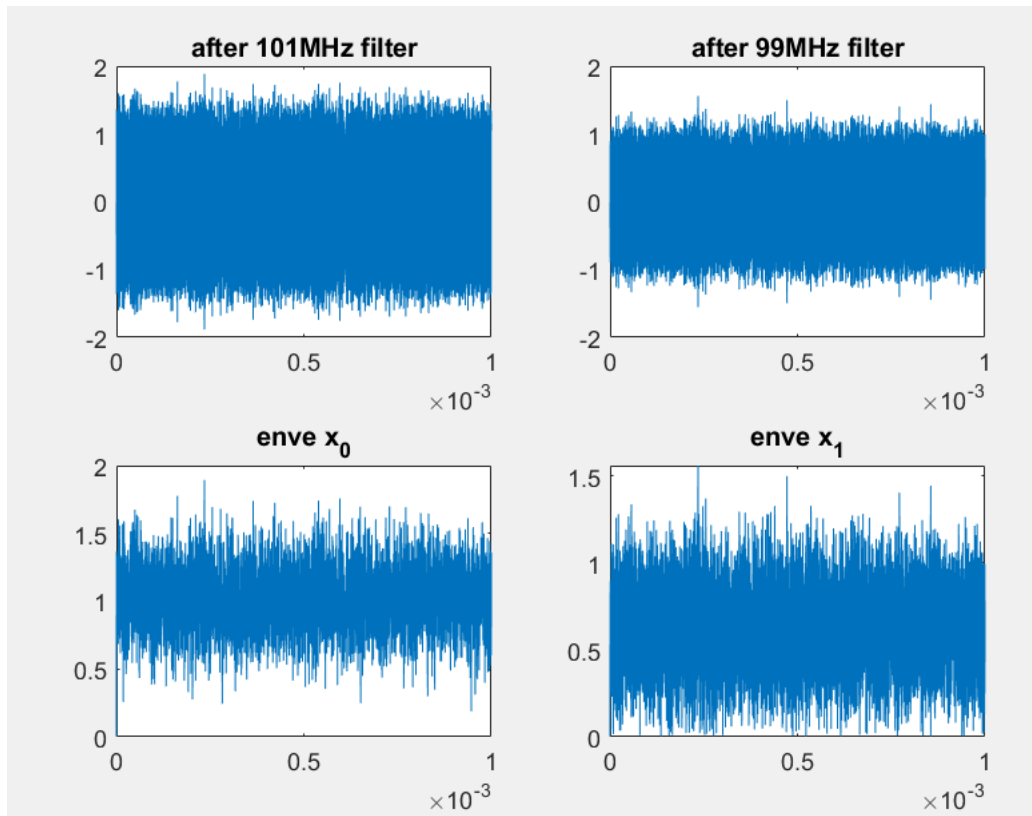


شکل ها به اعضای حالت های مختلف:

چون سیگنال به نویز را کم کردیم دامنه سیگنال در نویز گم شد و دامنه افزایش یافت و با این که نویز ها اجازه نمیدهند سیگنال را ببینیم اما حالت سینوسی همچنان کم و بیش دیده میشود.



دامنه برای فرکانس بزرگتر بیشتر است همان طور که در پایین میبینیم و پس از رد شدن از envelope detector دامنه سیگنال x_0 کمی بالاتر از سیگنال x_1 میباشد برای شناسایی و تخصیص حالت صحیح به بیت های ارسالی نقطه وسط را در نظر میگیریم و مقدار درست را تقریب میزنیم:



حال باید bit error rate را طبق خواسته سوال بررسی کنیم و نسبت تعداد بیت‌های غلط به کل بیت‌ها را ببینیم به ازای $N_Send = 20$ و اضا نتیجه خوبی داریم:

```
SNR =
-15

bit Error Rate =
fx 0
```

اما اگر تعداد بیت‌های ارسالی را بیشتر کنیم خطا بیشتر میشود اما نه آن چنان زیاد: چون زمان خیلی طول میکشد به همین $N_Send = 500$ بسنده کردم:

```
SNR =
-15

Bit Error Rate =
0.008
fx >>
```