به نام خدا

دانشگاه تهران

دانشکده‌ی مهندسی برق و کامپیوتر

پردازش سیگنال‌های زمان‌گسسته

گزارش پروژه 1

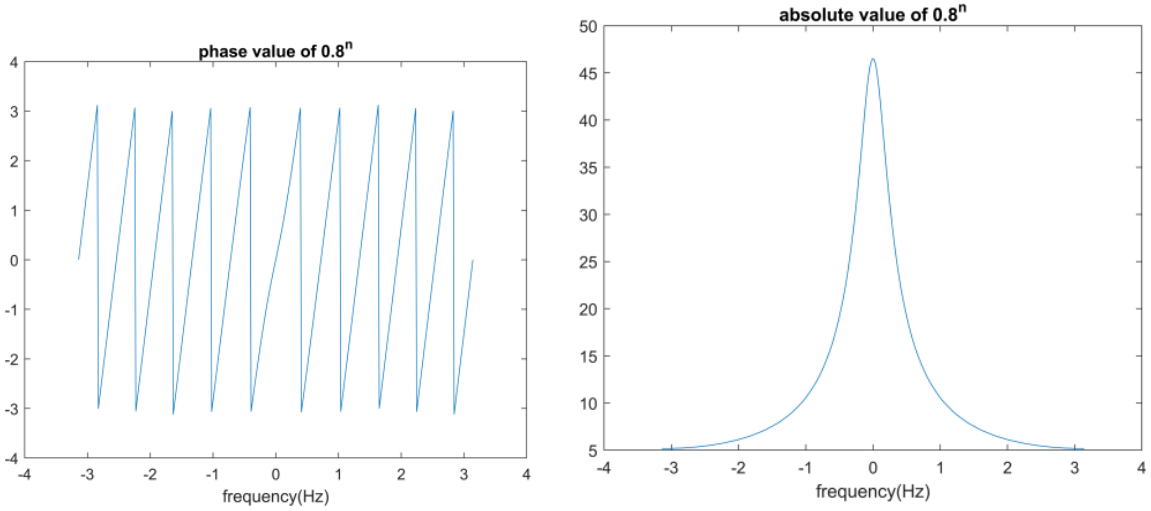
نام و نام خانوادگی	محمد عبائیانی
شماره‌ی دانشجویی	810198432

بخش اول:

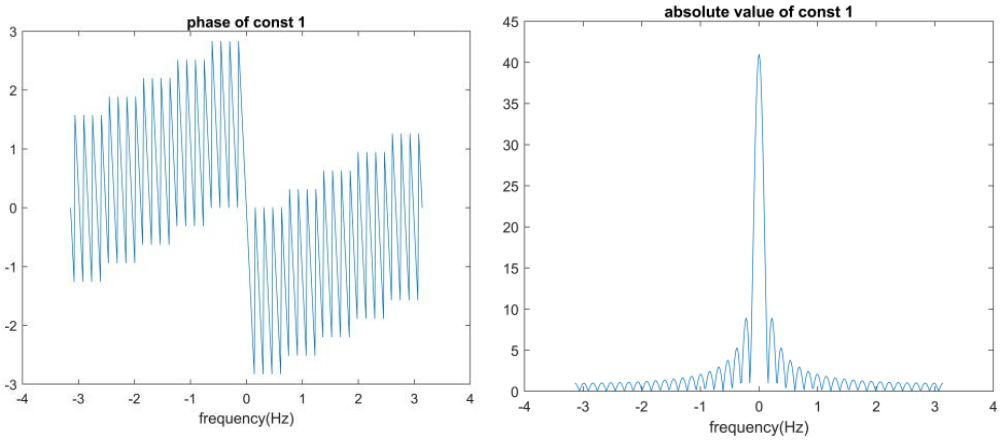
تابع تبدیل فوریه ی زمان گسسته به صورت زیر میباشد:

```
function X = DTFT(x,n)
k = -200:200;
X = x*(exp(-j*pi/200)).^(n'*k);
end
```

نمودار تبدیل فوریه ی توابع خواسته شده به صورت زیر است(شکل1 و 2):



شکل 1



شکل 2

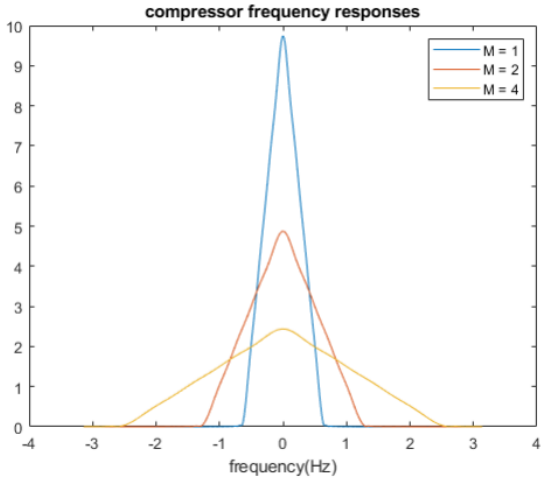
(تمام کد های به کار رفته در این پروژه به همراه خروجی ها به صورت فایل PDF جداگانه قرار داده شده است.)

2- تابع فشرده ساز به صورت زیر میباشد:

```
function y = compressor(x,M)
y = x(1:M:length(x));
end
```

برای اینکه از طولانی شدن محاسبات کامپیوتری جلوگیری شود از نوشتن حلقه خودداری شده است.

خروجی برای تابع $sinc^2$ به صورت زیر است(شکل 3):



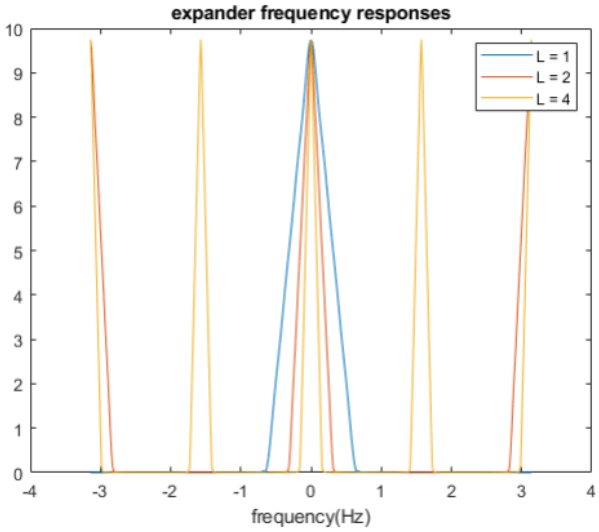
شکل 3

همانطور که مشخص است با فشرده شدن تابع فرکانس آن به همان نسبت گسترش پیدا میکند(نکته ی قابل توجه این است که مساحت پوشش داده شده در حوزه ی فرکانسدر فشرده سازی همواره ثابت است)

3- تابع باز کننده به صورت زیر میباشد:

```
function y = expander(x,L)
N = length(x)*L;
y = zeros(1,N);
y(1:L:N) = x;
end
```

خروجی خواسته شده به صورت زیر میباشد(شکل 4):



شکل 4

همانطور که مشاهده میشو با گستره شدن تابع در حوزه ی زمان فرکانس آن به همان نسبت فشرده میشود و همچنین دوره ی تناوب فرکانسی آن نیز کمتر خواهد شد.

-4

د) تابع درونیاب به صورت زیر میباشد:

```
function y = Interpolate(x,mode,n,fs,L)
switch mode
    case 1
        [Ts,T] =ndgrid(n,n(1:L:length(n)));
        y = sinc((Ts - T)*fs)*transpose(x(1:L:length(n)));
        y = transpose(y);
    case 2
        h = zeros(1,2*L -1);
        for t = -L:L
            h(t+L+1) = 1- abs(t)/L;
        end
        y = conv(x,h);
        y = y(L:length(y)-L-1);
    case 3
        y = spline(n(1:L:length(n)),x(1:L:length(n)),n);

end

end
```

الف) پس از استفاده از بلوک باز کننده تعدادی صفر بین هر داده قرار میگیرد، برای اینکه پیوستگی تابع حفظ شود نیاز است به نحوی داده های صفر میانی را با داده های قبل از اکسپند کردن مرتبط کرد که این کار با اینترپولین انجام میشود.

ب) فیلتر اول به صورت ایده آل عمل کرده و در حوزه ی زمان سیگنال را با تابع سینک کانوالو میکند(در حوزه ی فرکانس تابع rect میباشد)

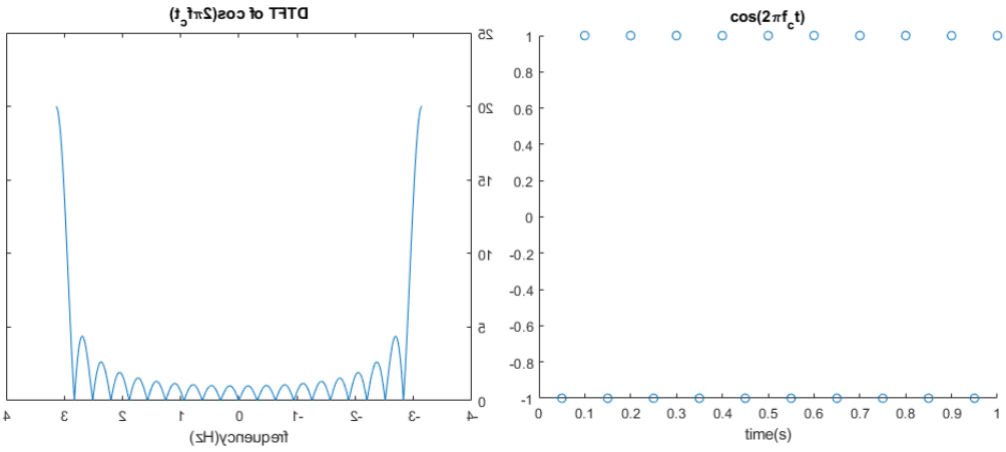
فیلتر دوم(خطی) هر داده را به صورت میانگین داده های قبل و بعد از خود تخمین میزند(برای اینکه داده هایی که میگیرد در هیچ حالتی صفر نباشد حداقل تعداد داده هایی که میگیرد دو برابر L یا ضریب بازکنندگی میباشد)

فیلتر سوم یا spline خطی بین داده های گرفته شده به دست می آورد به طوری که شیب این خط در تمام نقاط مینیمم شود. (میزان انحنای آن کمترین حالت ممکن باشد)

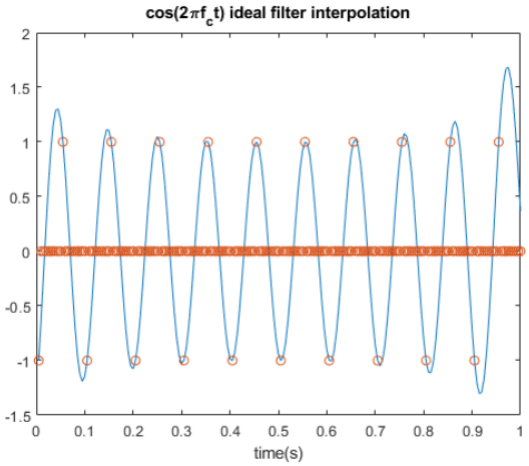
ج) تولید فیلتر ایده آل بسیار دشوار است و در عمل فیلتر ها با شیب مشخصی به ماکزیمم مقدار خود میرسند، بنابر این بخشی از فرکانس های ناخواسته نیز وارد فیلتر شده و بخشی از فرکانس های اصلی نیز تضعیف میشوند که این امر باعث اعوجاج سیگنال خروجی میشود.

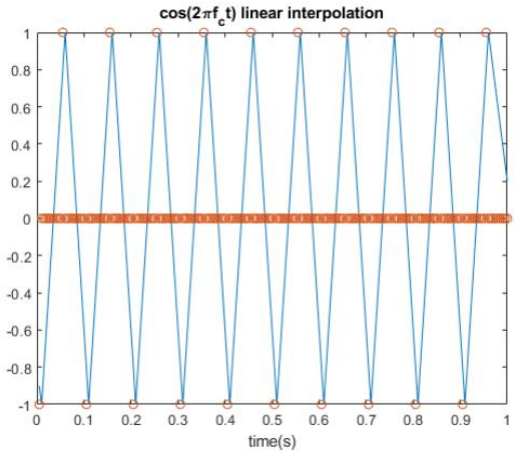
ه)

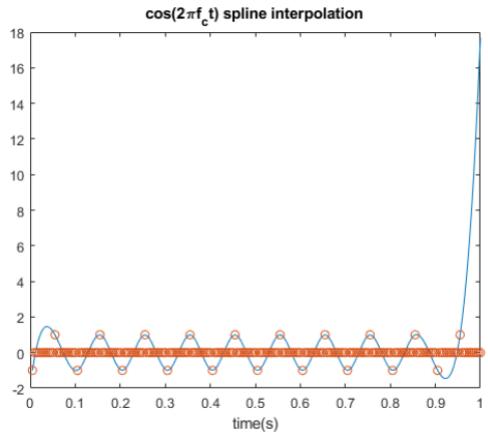
نمودار های این بخش به صورت زیر میباشند:



پس از اینترپولیشن نمودارهای زیر حاصل میشود(روش هر کدام ذکر شده است همچنین ضریب بازکنندگی 10 میباشد):





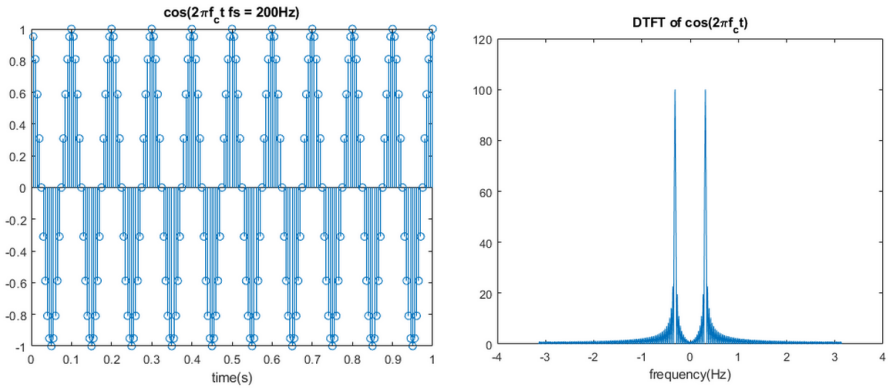


میانگین مربع خطاها:

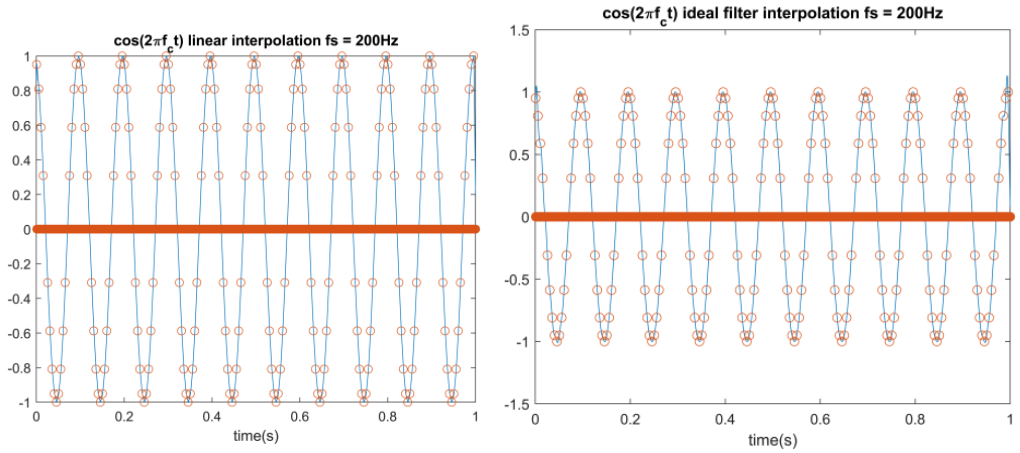
```
ans = "mse for ideal interpolation is 2.037515"
ans = "mse for linear interpolation is 1.485905"
ans = "mse for spline interpolation is 6.000488"
```

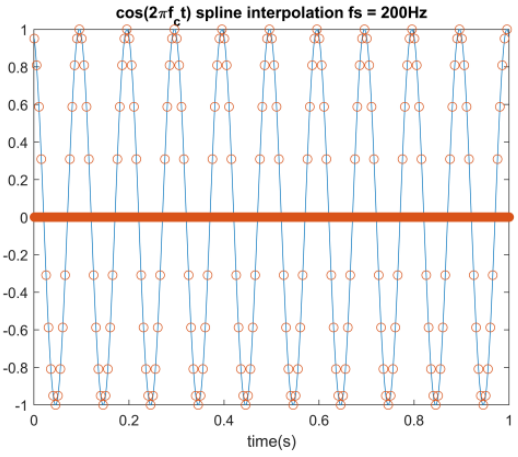
و)

برای فرکانس های 200 هرتز نمودار ها به صورت زیر میباشند:



توابع درونیاب:





میانگین مربع خطا:

```
ans = "mse for ideal interpolation is 0.041334"
ans = "mse for linear interpolation is 0.032213"
ans = "mse for spline interpolation is 0.039706"
```

همانطور که دیده میشود در fs =200 میانگین مربع خطاها بسیار کم خواهد شد و این سیگنال بسیار به کسینوس نزدیک میشود.

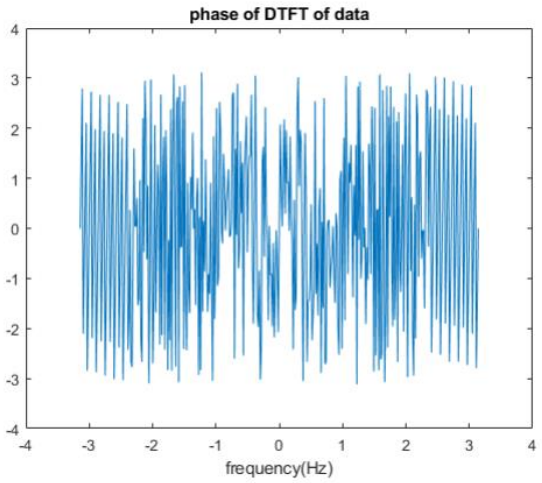
همچنین خطای هر سه روش به یکدیگر نزدیک میشود و کمترن خطا برای روش خطی است.

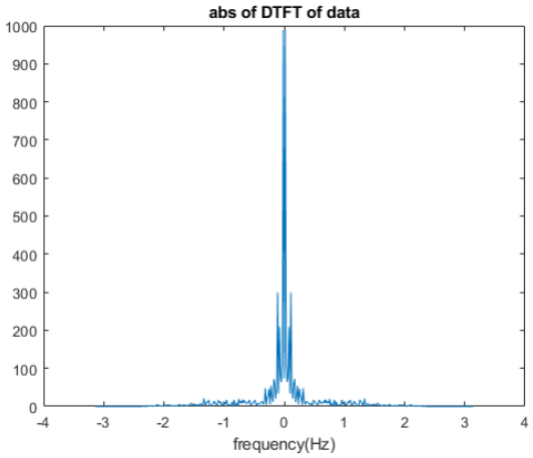
حداقل فرکانس fs باید از 4 برابر fc بیشتر باشدتا نقاط میانی نیز در تابع قرار گیرند...البته هرچقدر فرکانس بیشتر باشد به تابع اصلی نزدیک میشویم.

بخش دوم

1. ضمیمه شده در فایل کد

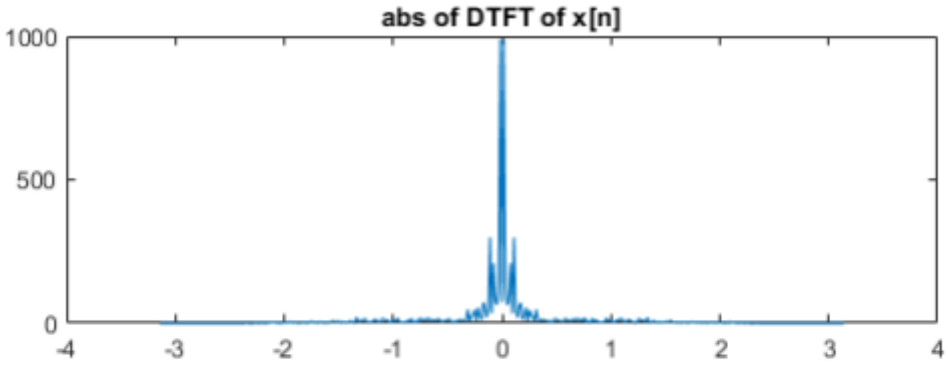
2. نمودار ها به شکل زیر میباشند:

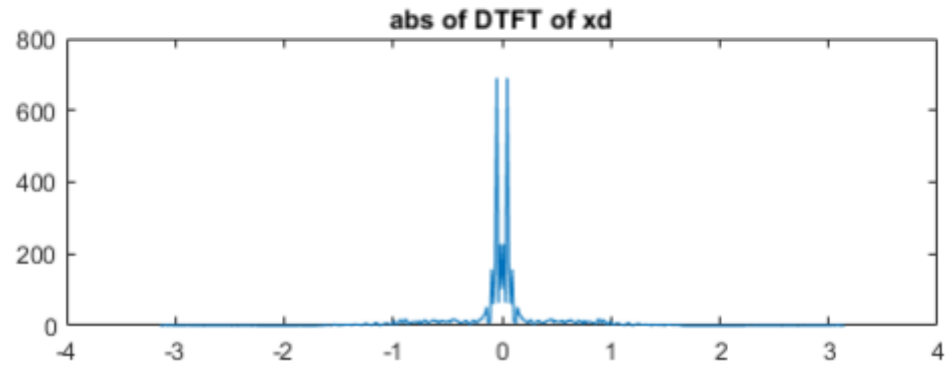




به دلیل بزرگ بودن مقدار L تبدیل فوریه توابع xe و ye نیاز به حافظه ی زیادی دارن:

تبدیل فوریه ی تابع ورودی و خروجی به شکل زیر است:





با استفاده از دستور rat مقدار L برابر 1223 و مقدار M برابر 882 میباشد.

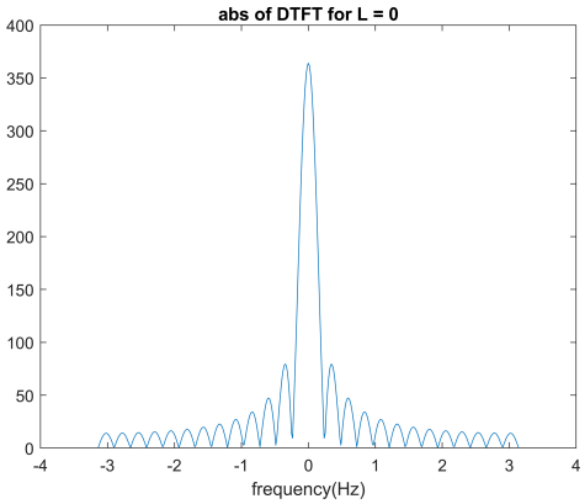
در فایل خروجی اگر درونیابی انجام نشود کیفیت صدا اف میکند و اگر درونیابی خطی نجام شود کیفیت صدا همانطور باقی میماند.

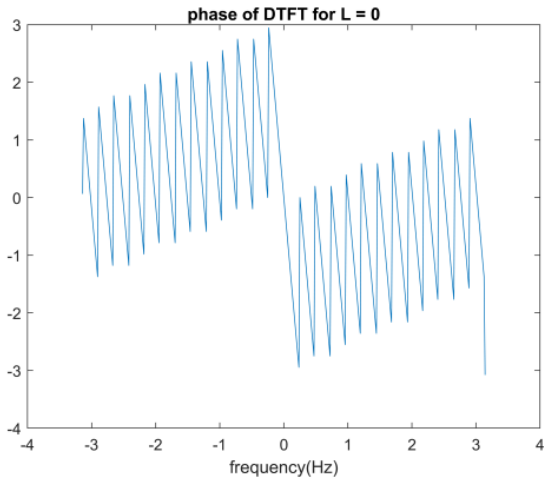
به صورت کلی صرف افزایش نرخ نمونه برداری باعث افزایش کیفیت صدا نخواهد شد.

بخش سوم

.1

نمودار های خواسته شده برای L=0 به شکل زیر هستند:



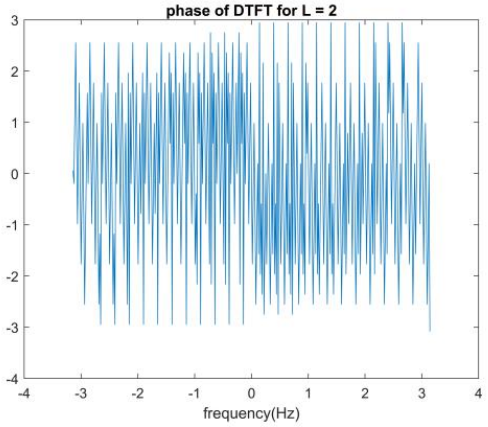


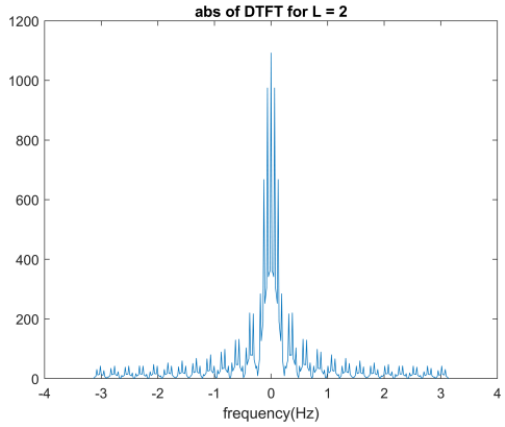
.2

```
ans =
"3dB bandwidth for s0 is 0.203694"
```

.3

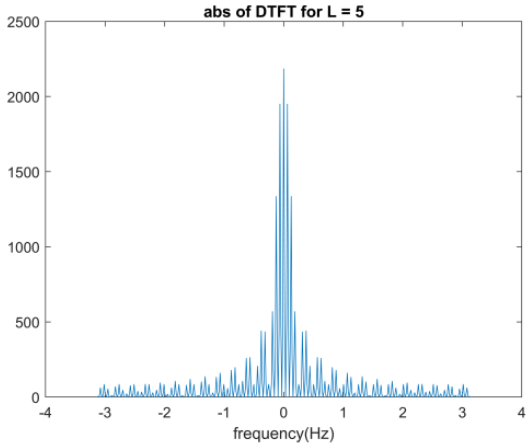
برای :L=2

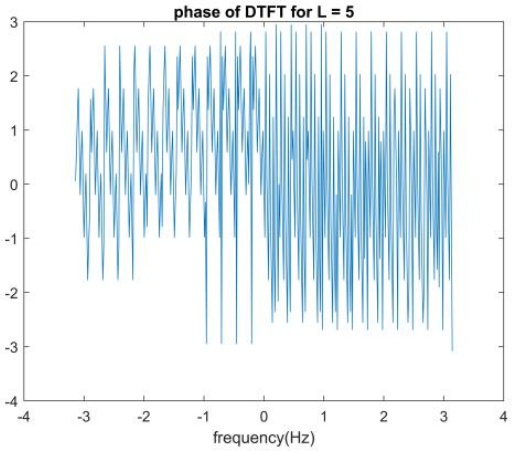





```
ans =
"3dB bandwidth for s2 is 0.047006"
```

برای L=5:






```
ans =
"3dB bandwidth for s5 is 0.047006"
```

.4

شکل تبدیل فوریه تقریبا یکسان میماند و پهنای باند نیز تقریبا ثابت خواهد ماند و سیگنال نیز همچنان پایین گذر خواهد بود،تنها رزلوشن و دامنه ی تبدیل فوریه افزایش پیدا میکند.

.5

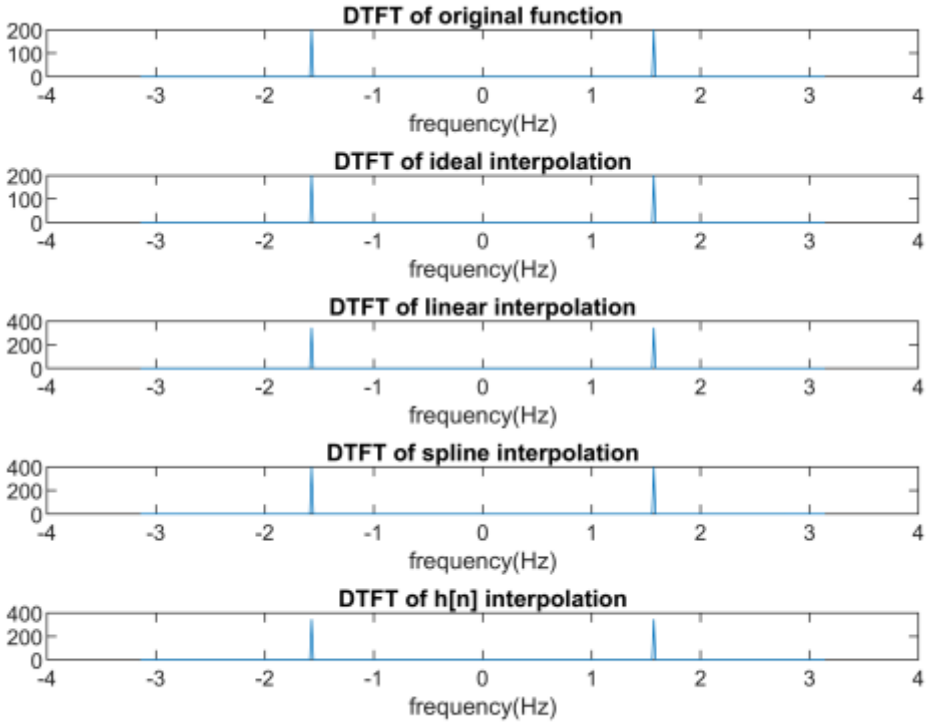
میتوان با توجه به شکل های قبل دریافت که برا L بینهایت نیز شکل تبدیل فوریه همین خواهد ماند و تنها رزلوشن نمونه ای آن بیشتر خواهد شد ، اما مشخصات آن ثابت خواهد ماند.

بخش چهارم

.1

پهنای باند برابر2*pi*fc/fs میباشد و برای دو برابر بودن نرخ نایکوئست بودن باید fs برابر 4fc باشد.

شکل تبدیل فوریه ی سگینال اصلی به همراه توابع درونیابی شده به شکل زیر میباشد:

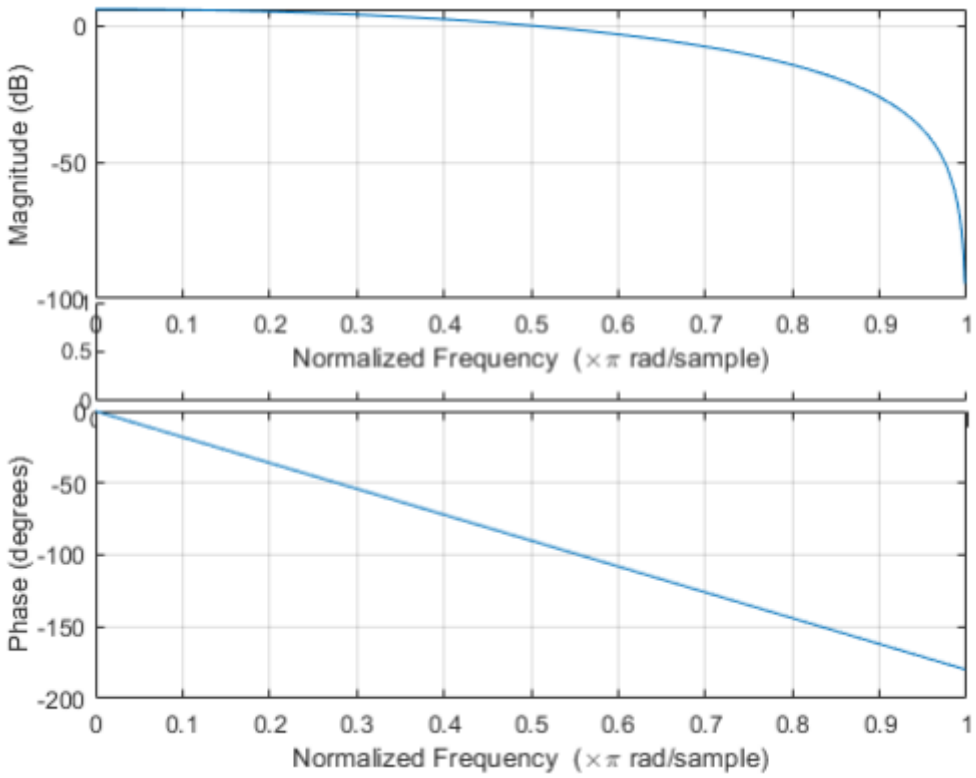


.5

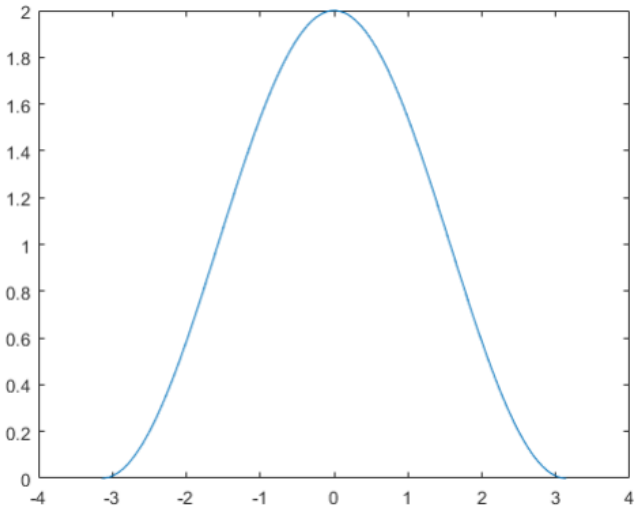
همانطور که مشاهده میشود شکل پاسخ های فرکانسی یکسان است اما دامنه های متفاوتی دارند که دامنه ی حالت ایده آل با دامنه ی سیگنال اصلی یکسان میباشد.

.3

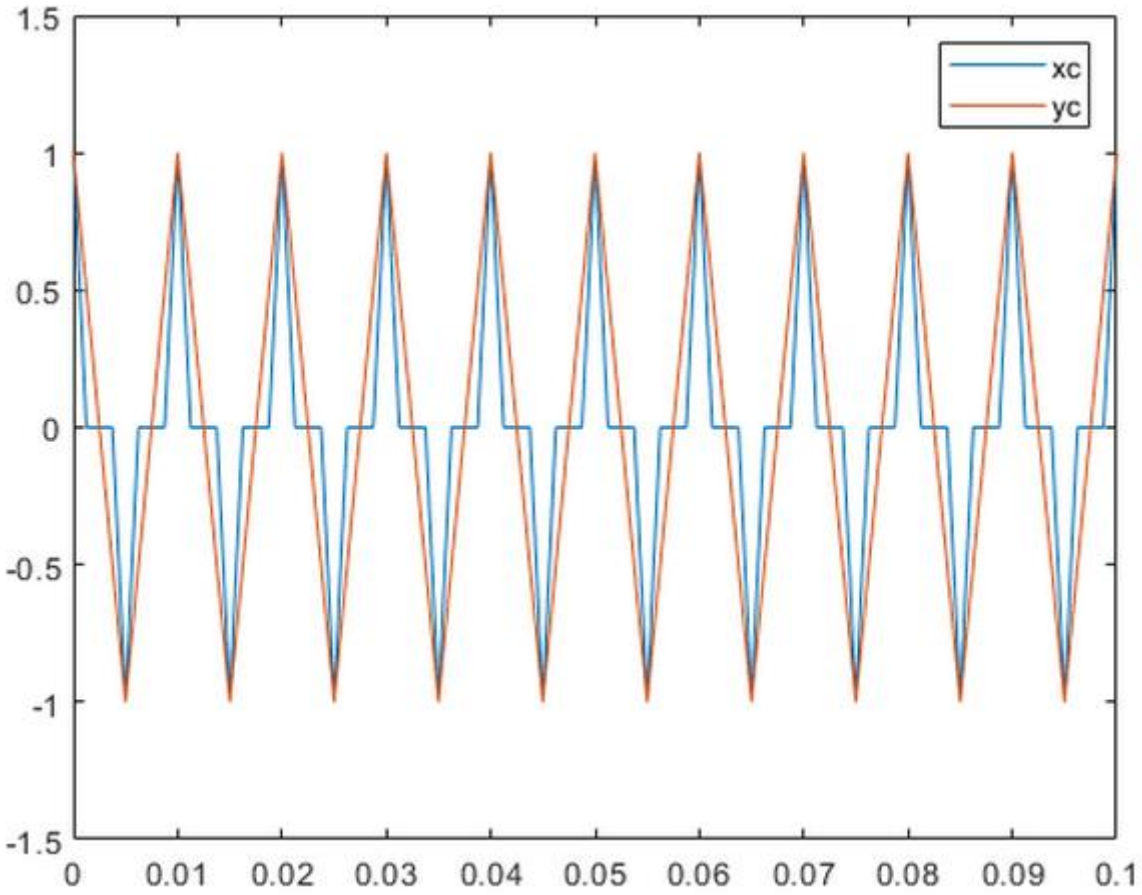
خروجی freqz:



پاسخ فرکانسی:



.4



.5

همانطور که از نتایج مشخط است سیگنال خروجی از فیلتر ایده آل مقادیر مشابهی با سیگنال ورودی دارد و تنها تفاوت آن این است که مقادیر آن در نقاط میانی مشخص شده اند.

.6

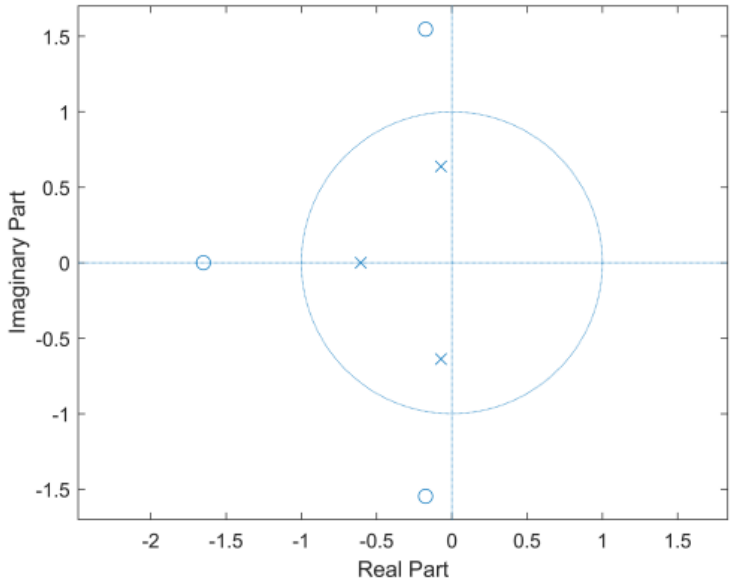
فیلتر داده شده در سوال به صورت خطی برای $L=2$ به خوبی عمل میکند اما برای مقادیر دیگر L کار نخواهد کرد.

بنابر این در این سیستم مشخص فیلتر داده شده قابل قبول میباشد.

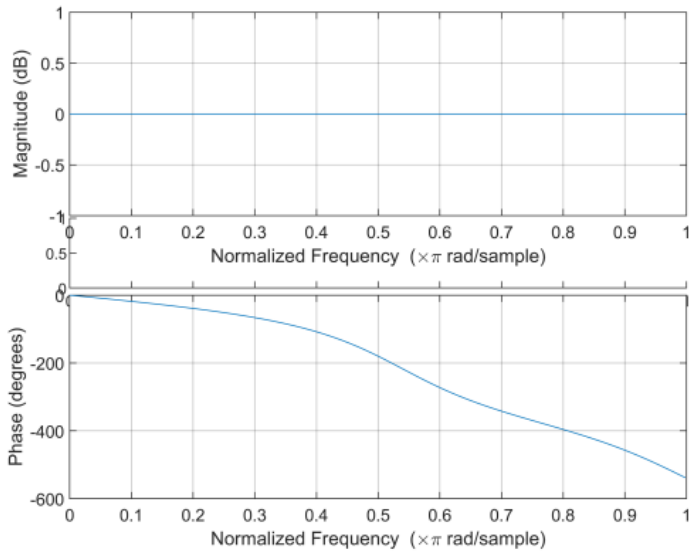
بخش پنجم

.1

نمودار قطب ها به شکل زیر است:



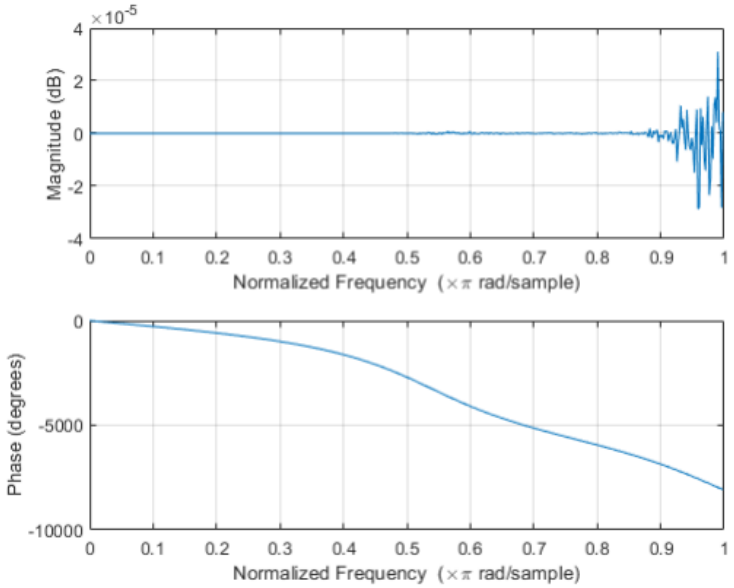
پاسخ فرکانسی:



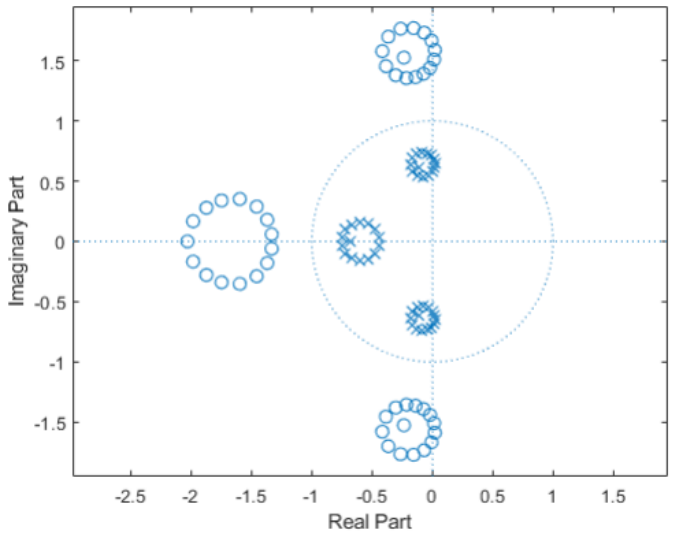
2. پس از شنده شدن خروجی متوجه میشویم که تفاوت چندانی در صدا ایجاد نشده و بنابراین میتوان گفت که تغییر فاز شنیده نخواهد شد.

.3

پاسخ فرکانسی:



قطب ها:



قطب ها و صفر های اضافه شده در سیستم حول قطب ها و صر های قبلی میباشند. با توجه به یکسان بودن تابه تبدیل کلی و به توان رسیدن آن به نظر میرسد که قطب ها و صفر ها نیز باید به توان برسند اما در اینجا مقداری با قطب و صفرهای قبلی فاصه دارند .

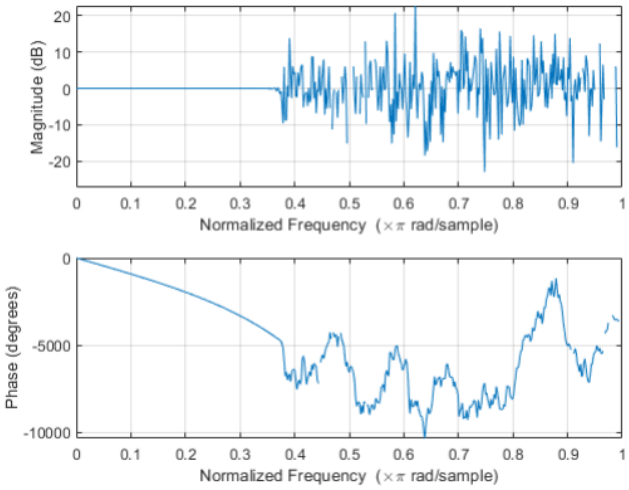
اختلاف به این دلیل است که راه کلی برای به دست آوردن جواب های معادله ی درجه 5 به بالا وجود ندارد و متلب این ریشه ها و قطب ها را به صورت تقریبی به دست می آورد.

.4

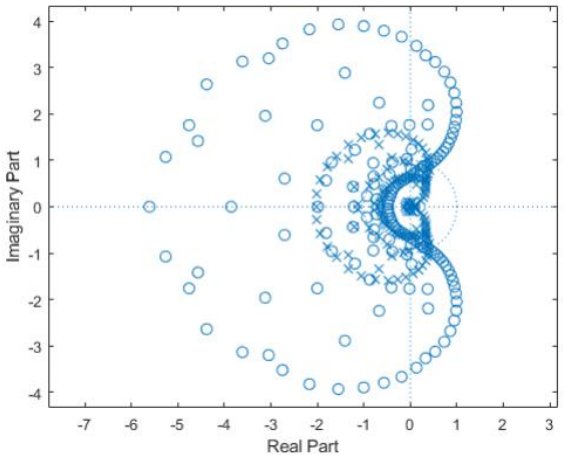
تن صدا کمی دچار تغییر شده است اما به صورت کلی همچنان همان صوت قبلی است. بهنظر میرسد که تغییر فاز بسیار شدید قابل شنیدن است.

.5

خروجی freqz:



قطب ها و صفرها:



در این بخش خروجی به کل دچار اختلال شده و صدای مشخصی قابل شنیدن نیست.

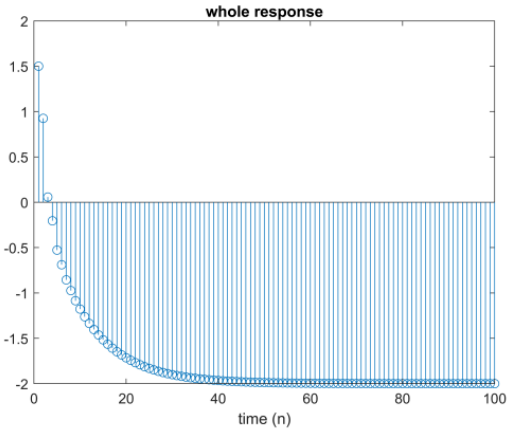
این میتواند به این دلیل باشد که علاوه بر فاز دامنه نیز به شدت دچار اعوجاج شده است. و همچنین این اعوجاج برای فاز نیز از حالت خطی خارج شده است.

بخش ششم

حل معادله ی کامل به روش اول:

در این روش از تبدیل z یک طرفه استفاده میکنیم و درواقع خروجی را به صورت مجموع پاسخ ورودی صفر و حالت صفر مینویسیم و با استفاده از خروجی های residuez به صورت جملات توانی مینویسیم.(توضیحات اضافه در فایل کد داده شده است)

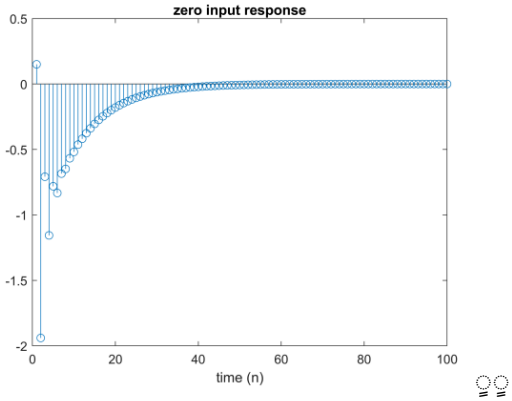
پاسخ به صورت زیر است:



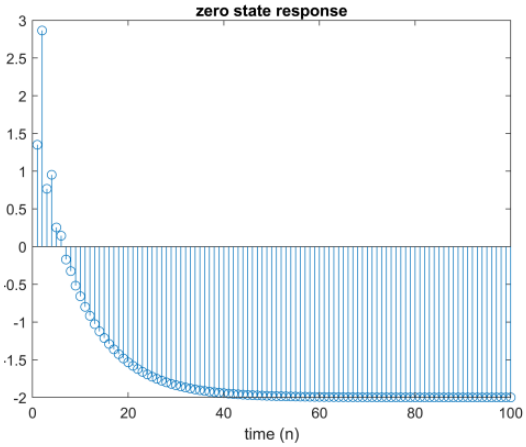

```
r = 4×1
   -2.0000
    2.1116
    1.7188
   -0.3304
p = 4×1
    1.0000
    0.9000
    0.5000
   -0.5000
k =

     []
```

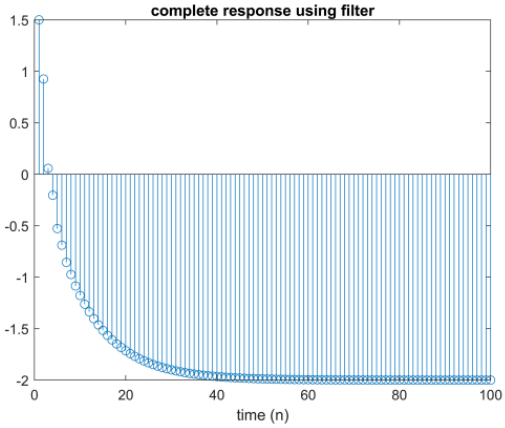
پاسخ گذرا یا ورودی صفر(با استفاده از filtic):



پاسخ ماندگار یا حالت صفر:



پاسخ کامل به رو دوم یا فیلتر:



همانطور که مشاهده میشود پاسخ هر دو روش یکسان میباشد. اما در روش اول ما پاسخ گذرا و ماندگار را به صورت جداگانه داریم اما در روش دوم همه ی چاسخ را به صورت تفکیک ناپذیر به دست می آوریم.

کد بخش دوم:

```
% complete response using filtic and filter
Y = [0 3];
X = [2 2];
x = (1/2).^n + 2;
xic = filtic(bn,an,Y,X);
yzi = filter(bn,an,x,xic);
stem(yzi)
title("complete response using filter")
xlabel("time (n)")
```