

HW4

Computer Assignment

Tohid Beheshti _____810100100

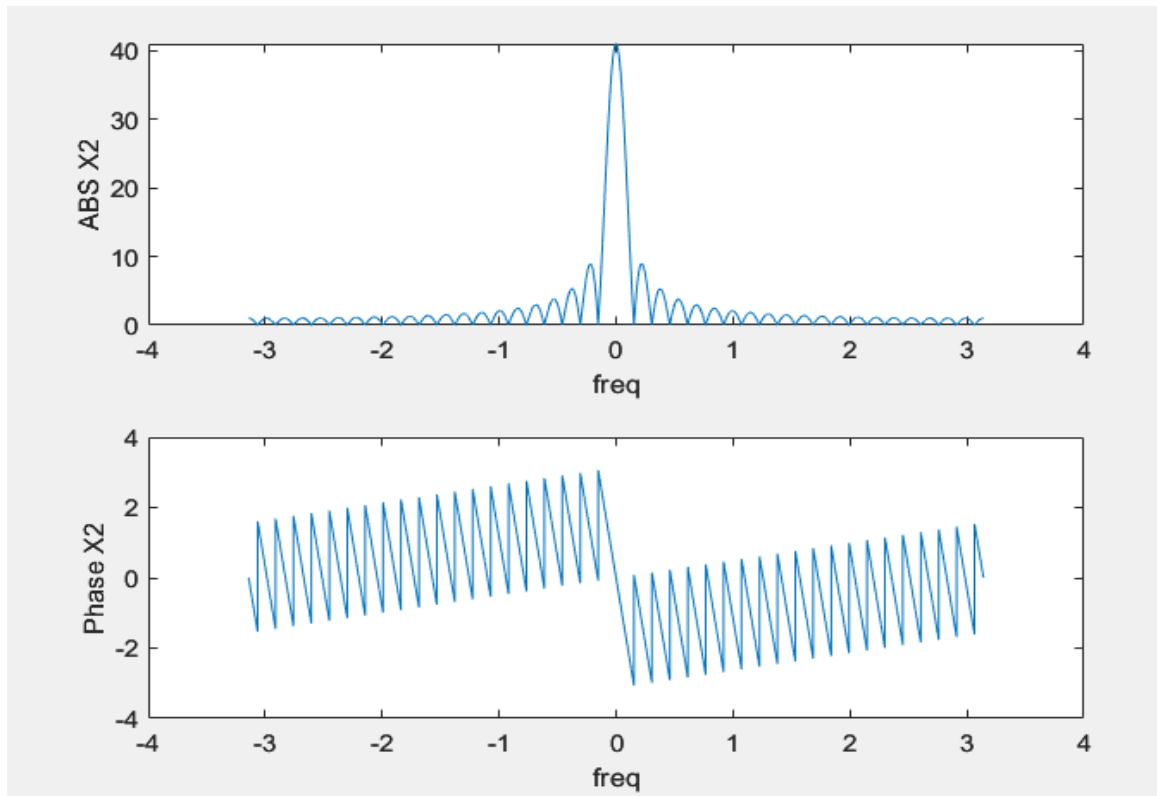
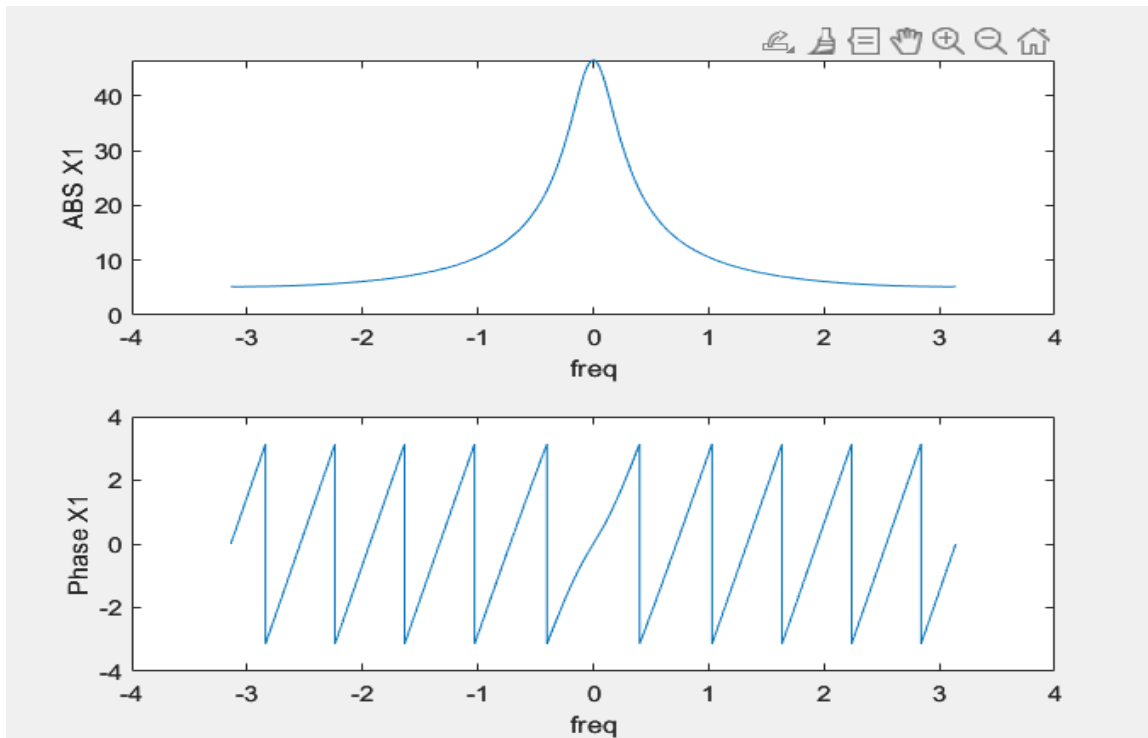
قسمت الف)

در ابتدا تابعی برای DTFT نوشته که در آن تبدیل فوریه گسسته را بدست میاوریم.

کد مربوط به تابع :

```
function Res = DTFT(x_n, n1, n2, omega)
    len = length(omega);
    Ans = zeros(1,len);
    intv = (n1:n2);
    for j = 1 : len
        Ans(j) = Ans(j) + (x_n .* exp(-1i * intv * omega(j)));
    end
    Res = Ans;
end
```

اندازه و زاویه پاسخ فرکانسی توابع داده شده :

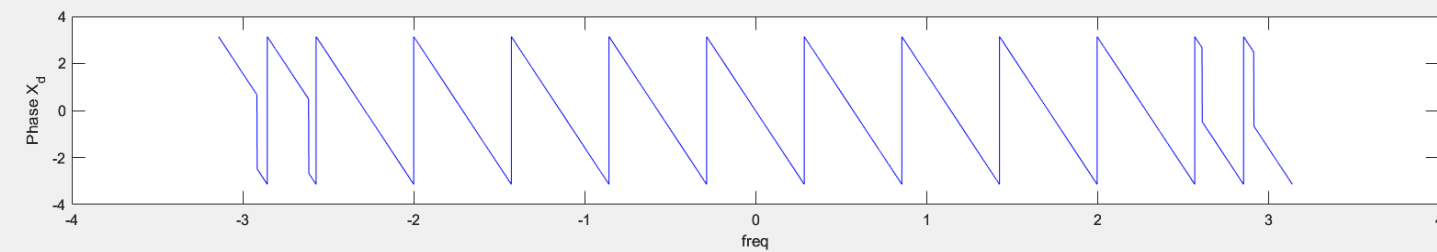
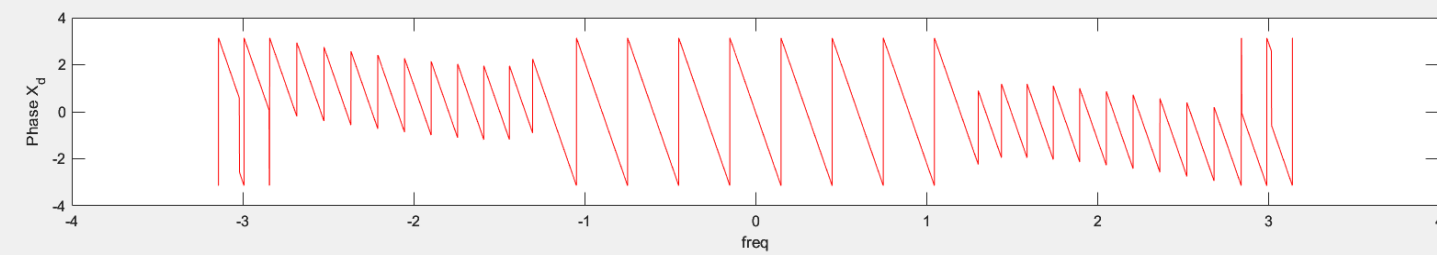
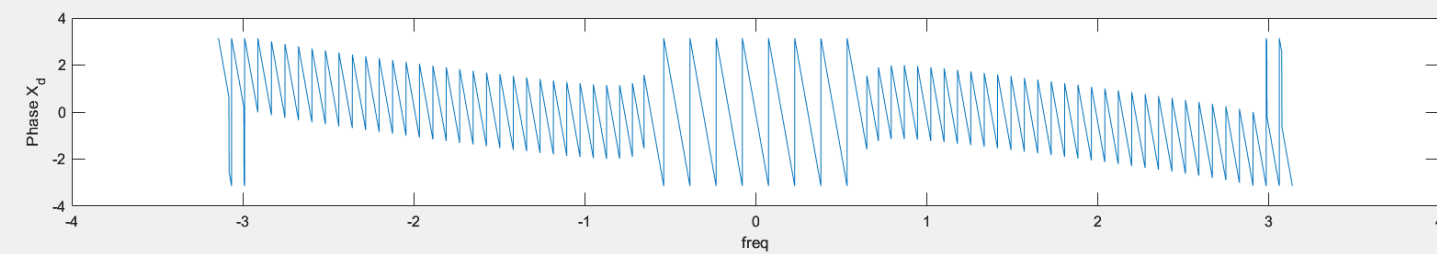
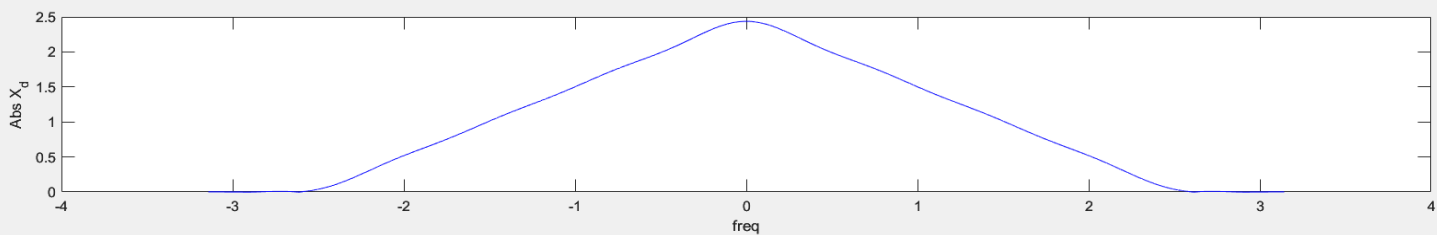
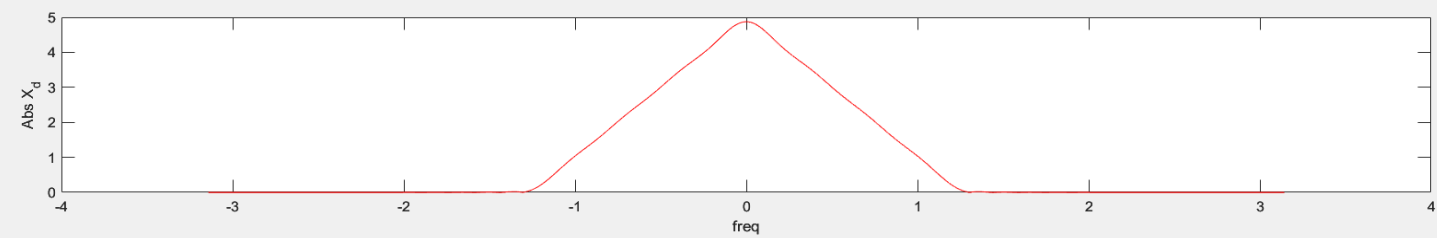
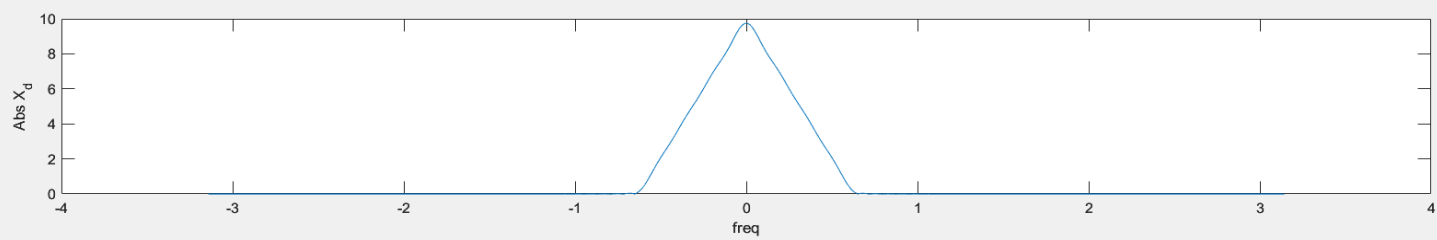


قسمت ب)

در این بخش تابع `compressor` ای نوشته که از سیگنال گسسته بانرخ M نمونه برداری میکند و بدین صورت سیگنالمان را فشرده سازی میکنیم .

```
function Res = compresor(x, M)
    len = length(x);
    comp_x = x(1 : M : len);
    Res = comp_x;
end
```

اکنون تابع داده شده (sinc^2) برای فشرده سازی را به تابع `compressor` داده میشود و سپس با کمک تابع `DTFT` پاسخ فرکانسی آن برای M های مختلف به صورت دامنه و فاز نمایش داده میشود.



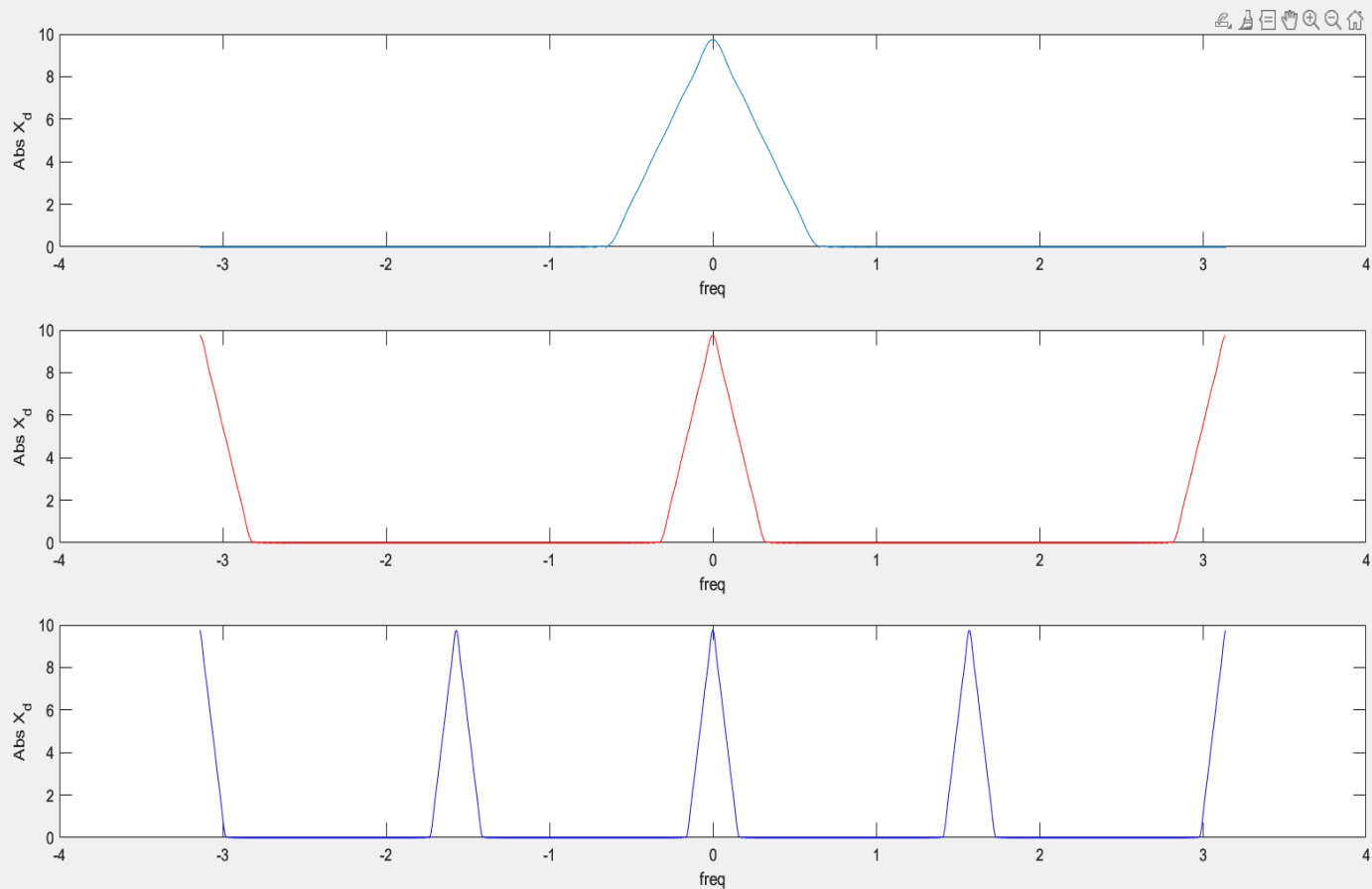
همانطور که میدانستیم پاسخ فرکانسی تابع (sinc^2) که در حوزه زمان به شکل کانولوشن دو تابع مربعی مییاشد، به صورت تابع مثلثی است و همچنین انتظار داریم با فشرده سازی آن با ضریب M در حوزه فرکانس با همان ضریب M باز شود و دامنه آن $1/M$ شود که همانطور که در شکل ها مشخص است کاملاً انتظارات ما محقق شده است و شکل پاسخ فرکانسی در فشرده سازی با ضریب 4 عریض تر و کوتاه تر از حالتی است که ضریب فشرده سازی 2 است .

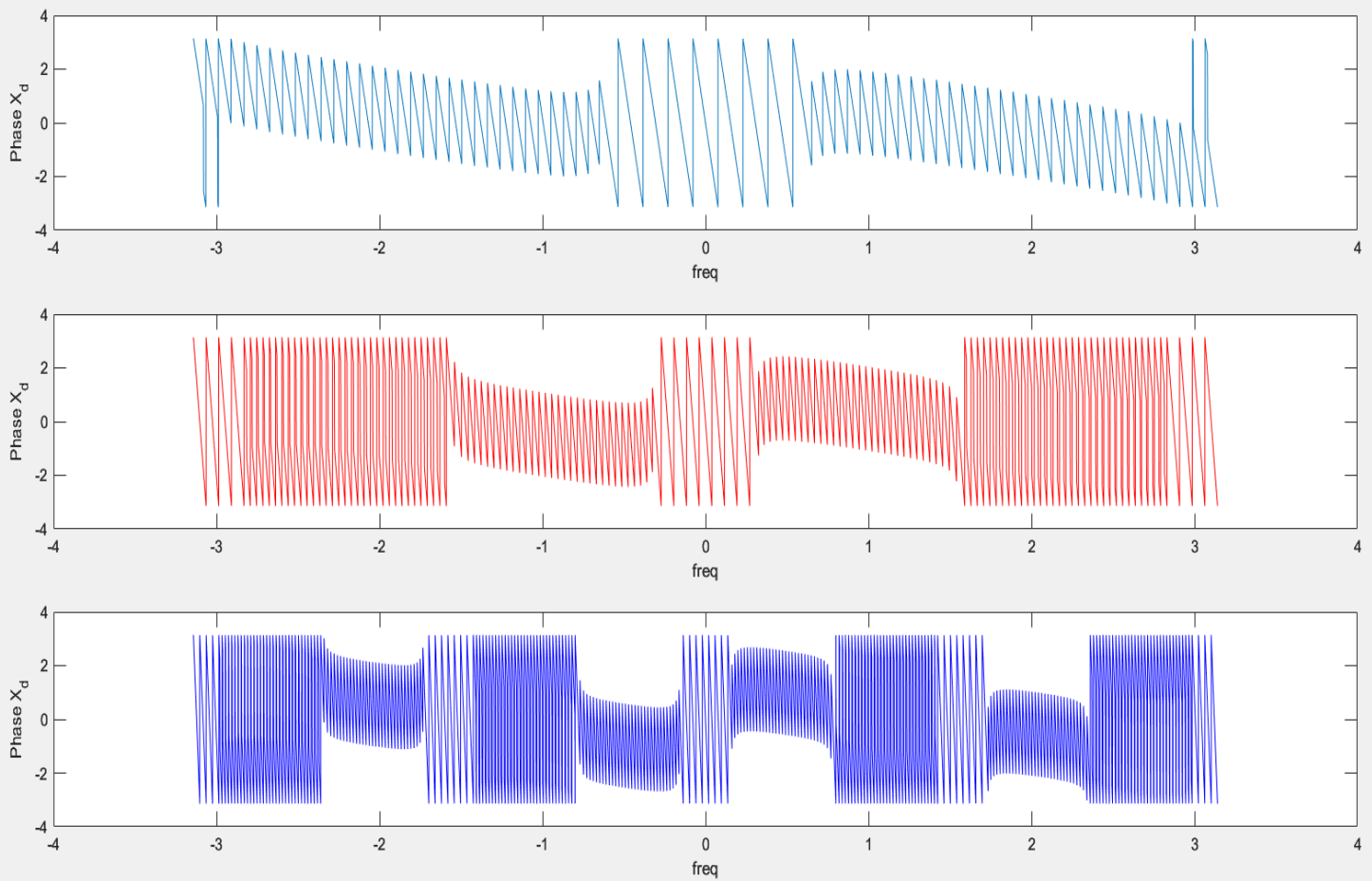
قسمت ج)

در این بخش تابع expander ای نوشته که سیگنال گسسته را با ضریب L باز میکند. درواقع اگر تعداد سمپل های سیگنالمان n تا باشد بعد از عبور از این بلوک $n*L$ تا میشود که $n*(L-1)$ تا از این سمپل ها صفر است و به این صورت سیگنالمان را expand میکنیم.

```
function Res = Expander(x, L)
    len_x = length(x);
    exp_1 = zeros(1, L * len_x);
    len_exp = length(exp_1);
    exp_1(1: L :len_exp) = x;
    Res = exp_1;
end
```

اکنون تابع داده شده برای فشرده سازی (sinc^2) را به تابع compressor داده میشود و سپس با کمک تابع DTFT پاسخ فرکانسی آن برای M های مختلف به صورت دامنه و فاز نمایش داده میشود.





همانطور که میدانستیم پاسخ فرکانسی تابع به صورت تابع مثلثی است و همچنین انتظار داریم با `expand` کردن آن با ضریب `L` در حوزه فرکانس با همان ضریب `L` جمع شود و همانطور که در شکل ها مشخص است کاملاً انتظارات ما محقق شده است و شکل پاسخ فرکانسی بعد از `expand` کردن با ضریب 2 (جمع تر شده) و وقتی ضریب `expand` ما 4 است شکل باز هم باریکتر و جمع تر شده از حالت قبل میشود.

قسمت د)

در این بخش ما باید یک بلوک برای درونیاب طراحی کنیم که برای اینکار سه روش مختلف درونیابی ارایه شده است که باید در یک تابع هر سه روش درونیابی ایده آل ، خطی و اسپلاین مکعبی پیاده سازی شود .

```
function Res = interpolator(x, L, mode)
len_x = length(x);
Res = zeros(1, L * len_x);
    if (mode == 1)
        len_res = length(Res);
        Res(1:L:len_res) = x;
        Res = L .* Res;

    elseif (mode == 2)
        for j = 1:len_x
            Res((j-1)*L+1 : j*L) = (1-abs(j-(1:L))/L).*x(j);
        end

    elseif (mode == 3)
        L1 = 1:len_x;
        l2 = 1:1/L:len_x+1-1/L;
        Res = spline(L1, x, l2);

    end
end
```

سوال ۱ :

۱- درونیاب ایده آل : این نوع درونیابی به سادگی و به کمک یک فیلتر پایین گذر ایده آل در حوزه فرکانس انجام میشود که gain این فیلتر برابر ضریب باز شدگی (L) است و فرکانس قطع آن نیز π/L است و بدین شکل سمپل های صفر درون سیگنال را درونیابی میکند .

۲- درون یاب خطی: در این روش درونیابی از انجایی که سمپل های سیگنال اصلی را داریم برای محاسبه مقدار سمپل های صفر از مقادیر سمپل های سیگنال اصلی کمک میگیریم و مقادیر با نسبت خطی بین این مقادیر ایجاد میکنیم ، به بیان ساده تر درواقع مقادیر اصلی سیگنال گسسته را با خطوط راست به همدیگر متصل میکنیم و صفرهای ایجاد شده را با استفاده از موقعیت آنها مقداررهی مینماییم.

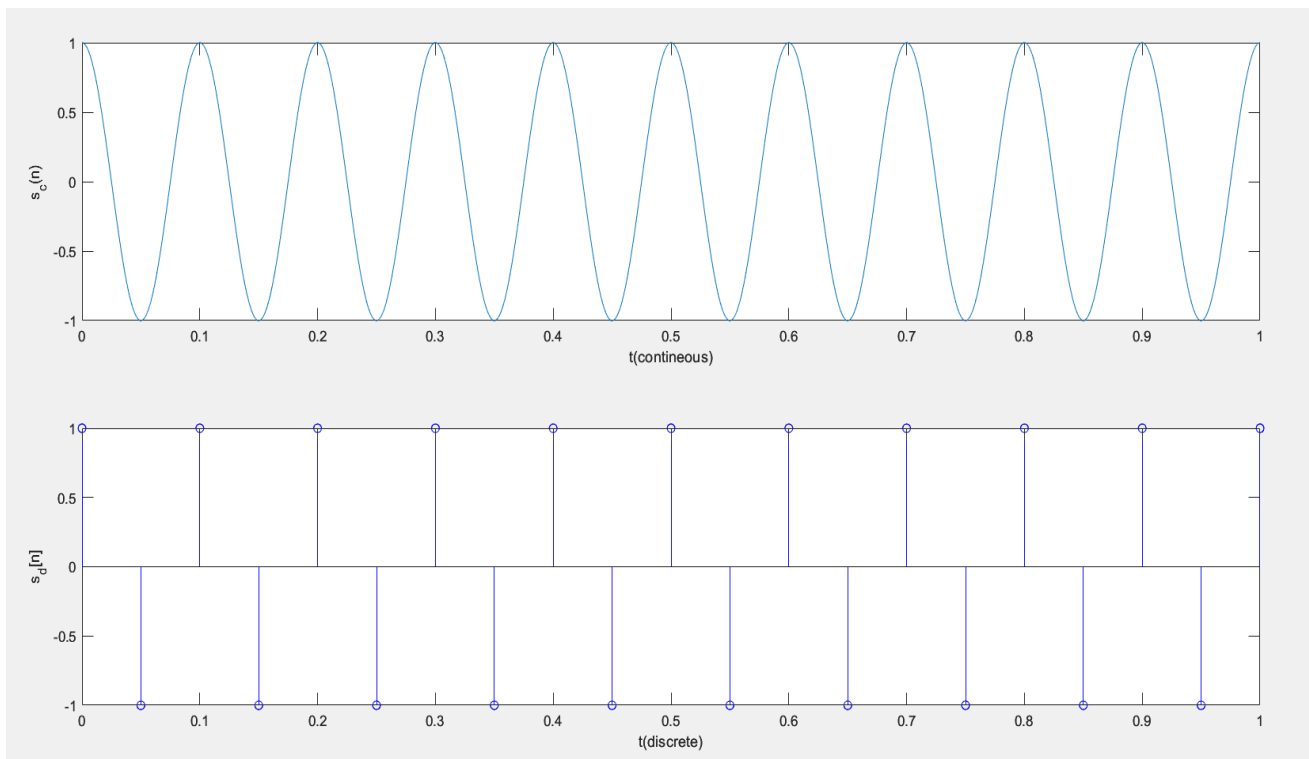
۳- درون یابی اسپلاین مکعبی : به طور کلی درونیابی اسپلاین را نسبت به درون یاب چند جمله ای ترجیح میدهیم زیراوقتی از چندجمله ای های مرتبه پایین استفاده میشود میتوان به خطای کمتری دست یافت. روش درونیابی اسپلاین است که در آن، درونیاب، نوع خاصی از یک چندجمله ای تکه ای است که یک (Spline) نامیده می شود. در اسپلاین مکعبی چند جمله ای های مکعبی را به صورت تکه ای با سمپل های اصلی مطابقت داده و درنتیجه نسبت به درونیاب خطی که دارای شکستگی های زیادی بود شکل نرمتری به ما میدهد .

سوال ۲ :

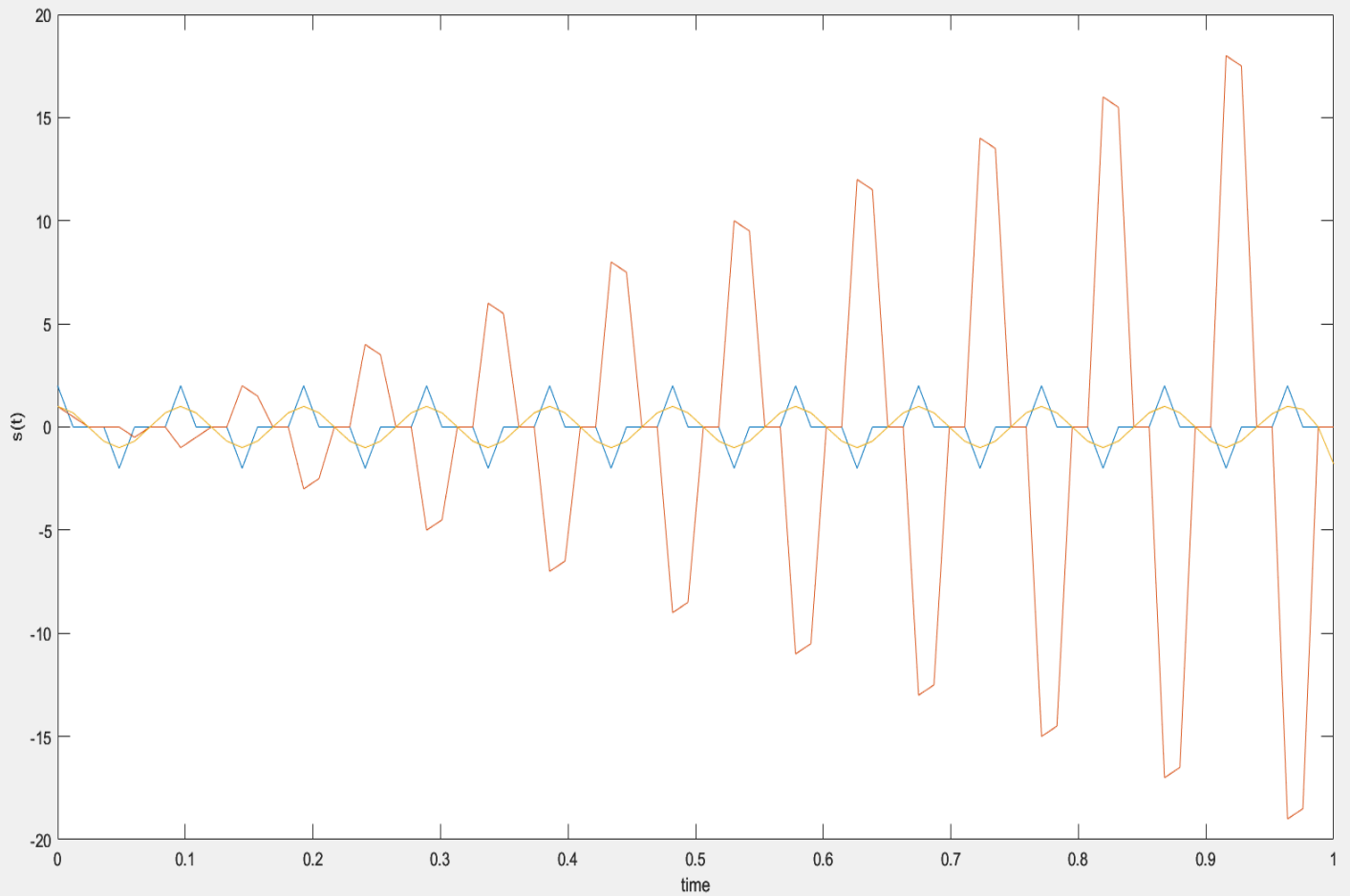
درونیاب ایده ال که به صورت یک فیلتر پایین گذر پیاده سازی میشود به صورت عملی قابل پیاده سازی نیست زیرا اولاً این فیلتر از نوع ایده ال است یعنی و باید به شکل کاملاً مربعی باشد که در عمل با استفاده از کامپوننت های مداری ساخت چنین فیلتری بسیار گران قیمت است و ثانیاً این درونیاب دارای پاسخ ضربه نامتناهی است بنابراین برای ساخت کامل سیگنال باید به نهایت ضربه داشته باشیم که عملی نیست .

سوال ۳ :

طبق خواسته سوال ابتدا سیگنال $s(t)$ را به صورت پیوسته (درواقع با فرکانس نمونه برداری بالا ، به طوری که پیوسته بنظر برسد) و همچنین به صورت گسسته (نمونه برداری شده از سیگنال پیوسته اول) نمایش داده میشود.



اکنون سیگنال گسسته را به سه روشی که تا اینجا شرح دادیم درونیابی میکنیم :



که در آن سیگنال قرمز با استفاده از درونیابی ایده ال و سیگنال آبی به کمک درونیابی خطی و سیگنال نارنجی با استفاده از درونیابی اسپلاین مکعبی تشکیل شده است ، اکنون با مقایسه این سیگنال ها میتوان دید که سیگنال درونیابی شده با استفاده از روش سوم (اسپلاین مکعبی) از دقت بسیار خوبی نسبت به سیگنال هایی که از دو روش دیگر بازیابی شده اند دارد و همچنین نرم بودن شکل آن نسبت به حالت خطی (به صورت خطوط شکسته میباشد) کاملاً مشهود میباشد.