دانشگاه تهران
پردیس دانشکده‌های فنی

پردازش سیگنال های دیجیتال

گزارش فاز ۱ پروژه ۱

سید علیرضا جاوید

۸۱۰۱۹۸۳۷۵

استاد

دکتر بدیعی

۷ آذر ۱۴۰۱

فهرست مطالب

مقدمه

در این پروژه به پیاده سازی برخی بلوک های مقدماتی پردازش سیگنال در متلب می پردازیم و روش های درونیابی برای پردازش دیجیتال سیگنال و فیلتر های مناسب بکار رفته رفته در آن را مقایسه و بررسی می کنیم. همچنین به پیاده سازی تابع های مناسب برای محاسبه تبدیل فوریه گسسته زمان [۱] و تبدیل زد [۲] می پردازیم.

[۱] Discrete-Time Fourier Transform
[۲] Z Transform

۱ تبدیل فوریه

در درس دیدیم که در صورتی که سیگنال بصورت مطلق جمع پذیر باشد $\sum_{-\infty}^{\infty} |x[n]| < \infty$ در این صورت می توان تبدیل فوریه گسسته زمان را به صورت زیر تغریف کرد:

$$X\left(e^{j\omega}\right) \triangleq \mathcal{F}[x(n)] = \sum_{k=-\infty}^{\infty} x(k)e^{-j\omega k}$$

۱.۱ پیاده سازی تبدیل فوریه گسسته زمان

برای پیاده سازی در متلب بهتر است سعی کنیم به صورت برداری رابطه ای برای آن پیدا کنیم. اگر $\mathbf{x}$ یک بردار شامل تمامی مقادیر سیگنال باشد، و $\mathbf{k}$ بردار اندیس های متناظر بردار $\mathbf{x}$ باشد، می توان نوشت:

$$\mathbf{X} = \mathbf{x}\, e^{-jk^T\omega}$$

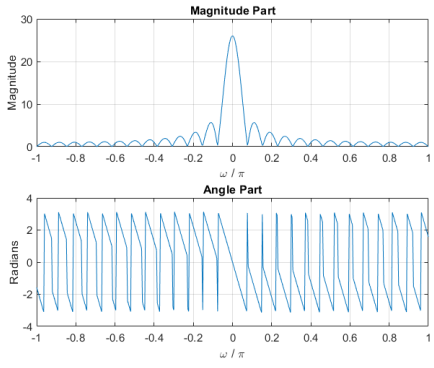
که یک ضرب برداری و نتیجه آن بردار تبدیل فوریه گسسته زمان سیگنال است. برای پیاده سازی در متلب داریم:

```
function X = dtft(x)
% we divide [-pi, pi] into N+1 points.
  k = 1 : length(x);
  N = 500;
  w = linspace(-pi,pi,N + 1);
  X = x * (exp(-1i*k'*w));
% if we want to plot dtft signal
  plot_dtft(X,w)
end
```

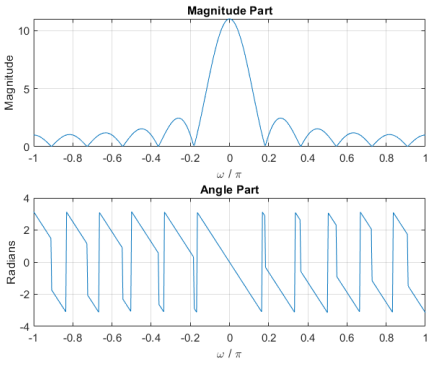
حالا برای هر یک از توابع داده شده تبدیل فوریه سیگنال را رسم و نتایج را بررسی می کنیم.

۱.

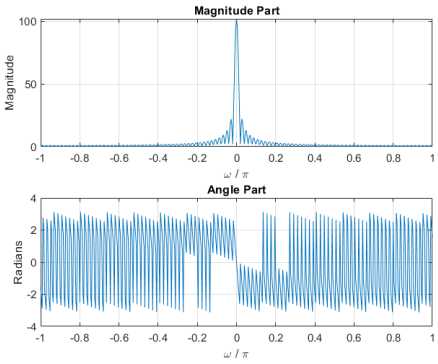
$$\mathcal{R}_M[n] = f(x) = \begin{cases} 1, & 0 \le n < M \\ 0, & \text{otherwise} \end{cases}$$



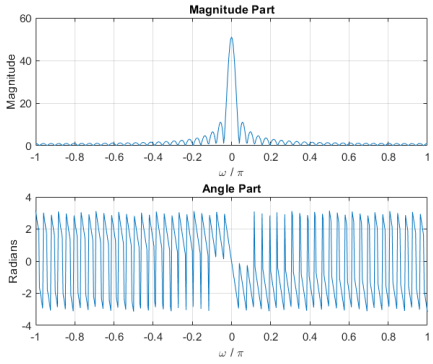
(آ) تبدیل فوریه سیگنال اول به ازای $M = 10$



(ب) تبدیل فوریه سیگنال اول به ازای $M = 25$



(ج) تبدیل فوریه سیگنال اول به ازای $M = 50$



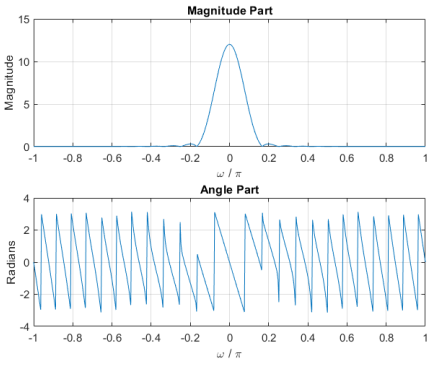
(د) تبدیل فوریه سیگنال اول به ازای $M = 101$

شکل ۱: تبدیل فوریه سیگنال اول

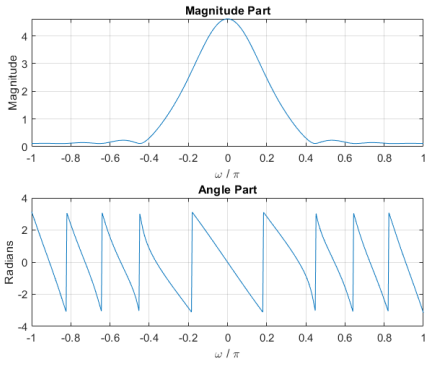
همانگونه که در شکل ۱ مشاهده می شود با افزایش مقدار M سیگنال به پله ثابت نزدیک تر شده و مقدار تبدیل فوریه نیز، تیزتر شده و به سیگنال ضربه میل می کند.

۲.

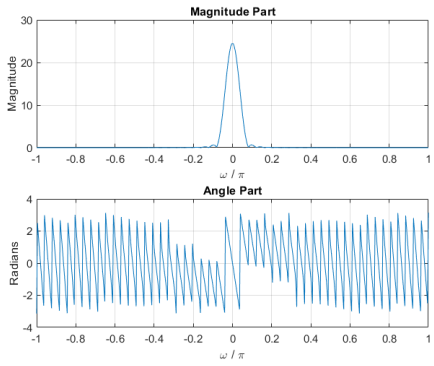
$$\mathcal{C}[n] = 0.5 \left[1 - \cos \frac{2\pi n}{M-1}\right] \mathcal{R}_M[n]$$



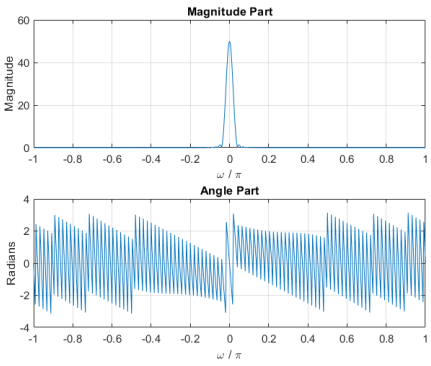
(آ) تبدیل فوریه سیگنال دوم به ازای $M = 10$



(ب) تبدیل فوریه سیگنال دوم به ازای $M = 25$



(ج) تبدیل فوریه سیگنال دوم به ازای $M = 50$



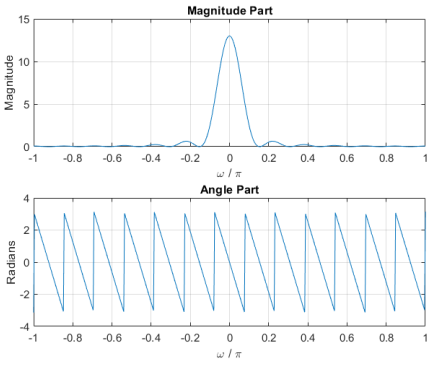
(د) تبدیل فوریه سیگنال دوم به ازای $M = 101$

شکل ۲: تبدیل فوریه سیگنال دوم

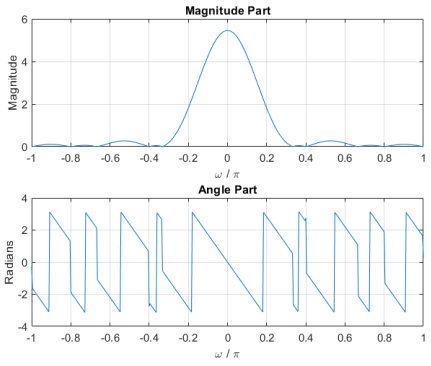
مانند مورد قبل با افزایش M همانطور که سیگنال در حوزه زمان باز تر و بزرگتر می شود در حوزه فرکانس کوچکتر شده و به تابع ضربه نزدیک می شود.

۳.

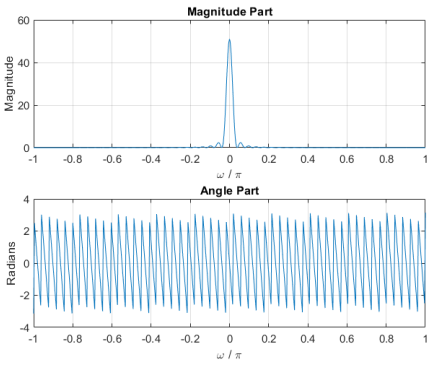
$$\mathcal{T}_M[n] = \left[1 - \frac{|M-1-2n|}{M-1}\right] \mathcal{R}_M[n]$$



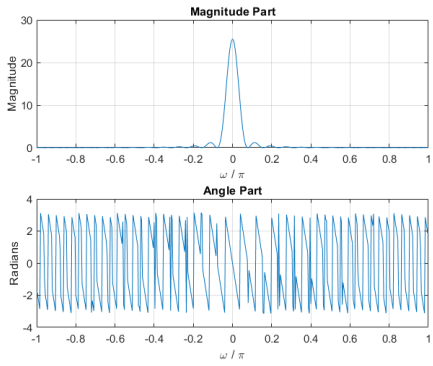
(آ) تبدیل فوریه سیگنال سوم به ازای $M = 10$



(ب) تبدیل فوریه سیگنال سوم به ازای $M = 25$



(ج) تبدیل فوریه سیگنال سوم به ازای $M = 50$



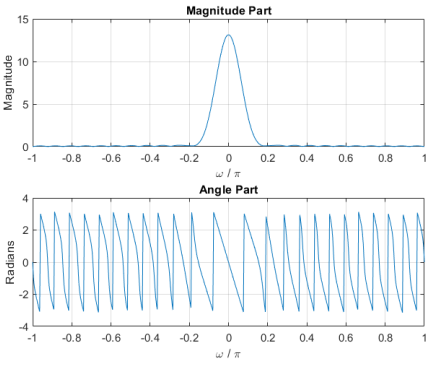
(د) تبدیل فوریه سیگنال سوم به ازای $M = 101$

شکل ۳: تبدیل فوریه سیگنال سوم

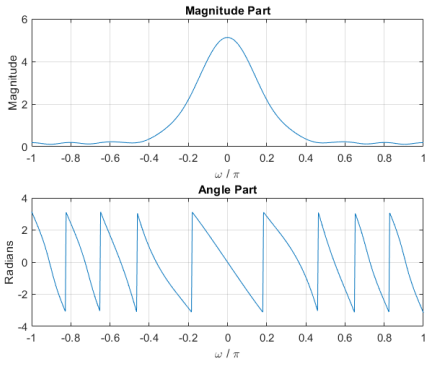
مانند مورد قبل با افزایش M همانطور که سیگنال در حوزه زمان باز تر و بزرگتر می شود در حوزه فرکانس کوچکتر شده و به تابع ضربه نزدیک می شود.

۴.

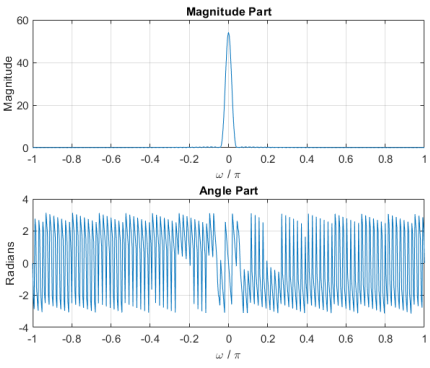
$$\mathcal{H}_M[n] = \left[0.54 - 0.46\cos\frac{2\pi n}{M-1}\right]\mathcal{R}_M[n]$$



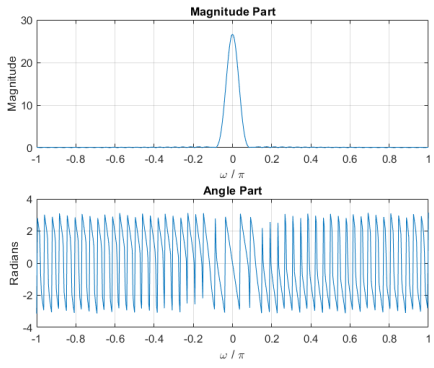
(آ) تبدیل فوریه سیگنال چهارم به ازای $M = 10$



(ب) تبدیل فوریه سیگنال چهارم به ازای $M = 25$



(ج) تبدیل فوریه سیگنال چهارم به ازای $M = 50$



(د) تبدیل فوریه سیگنال چهارم به ازای $M = 101$

شکل ۴: تبدیل فوریه سیگنال چهارم

مانند مورد قبل با افزایش M همانطور که سیگنال در حوزه زمان باز تر و بزرگتر می شود در حوزه فرکانس کوچکتر شده و به تابع ضربه نزدیک می شود.

۲.۱ پاسخ فرکانسی برای معادله تفاصلی

$$y[n] = \sum_{m=0}^{M} b_m x[n-m] - \sum_{\ell=1}^{N} a_l y[n-\ell]$$

$$H(e^{j\omega}) = \frac{Y(e^{j\omega})}{X(e^{j\omega})} = \frac{\sum_{m=0}^{M} b_m e^{-j\omega m}}{1 + \sum_{\ell=1}^{M} a_m e^{-j\omega \ell}}$$

همانطور که از رابطه بالا مشخص است، می توانیم جمع ها را به صورت ضرب داخلی برداری در متلب پیاده سازی کنیم تا سرعت و محاسبات بهتری داشته باشیم:

$$\mathbf{H} = \frac{\mathbf{b}\, e^{-j\mathbf{m}^T\omega}}{1 + \mathbf{a}\, e^{-j\mathbf{l}^T\omega}}$$

```
function H = diff_resp(a,b)
  kx = 1 : length(b);
  ky = 1 : length(a);
  N = 500;
  w = linspace(-pi,pi,N + 1);
  H = (b * exp(-1i*kx'*w)) ./ (1 + a *exp(-1i*ky'*w));
  plot_dtft(H,w)
end
```

.۱

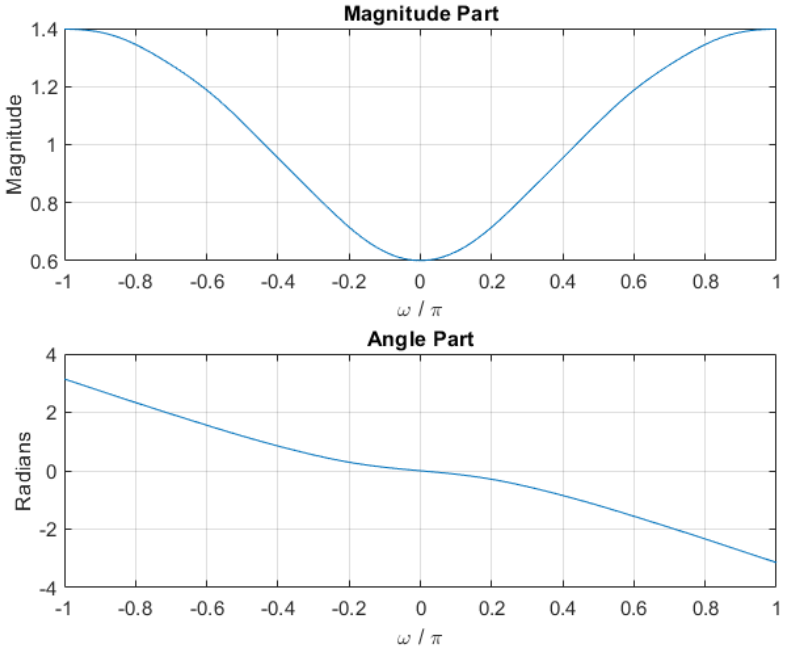
$$y[n] = x[n] - \sum_{l=1}^{6} (0.4)^{\ell} y[n-\ell]$$

با محاسبه تبدیل فوریه داریم:

$$Y(e^{j\omega}) = X(e^{j\omega}) - Y(e^{j\omega}) \sum_{l=1}^{6} (0.4)^{\ell} e^{-j\omega\ell}$$

$$H(e^{j\omega}) = \frac{1}{1 + \sum_{l=1}^{6} (0.4)^{\ell} e^{-j\omega\ell}}$$

در شکل ۵ می توانیم دامنه و فاز پاسخ فرکانسی معادله بخش اول را مشاهده کنیم.



شکل ۵: دامنه و فاز پاسخ فرکانسی معادله تفاضلی

۳.۱ محاسبه پاسخ معادلات تفاضلی

$$y[n] = \sum_{\ell=0}^{3} x[n-2\ell] - \sum_{m=1}^{3} (0.7)^m y[n-2m]$$

۱. برای محاسبه تئوری داریم:

$$H(e^{j\omega}) = \frac{\sum_{l=0}^{3} e^{-j2\omega\ell}}{1 + \sum_{m=1}^{3} (0.7)^m e^{-j2\omega m}}$$

$$x_1[n] = 1 + cos(\frac{\pi n + \pi}{2})$$

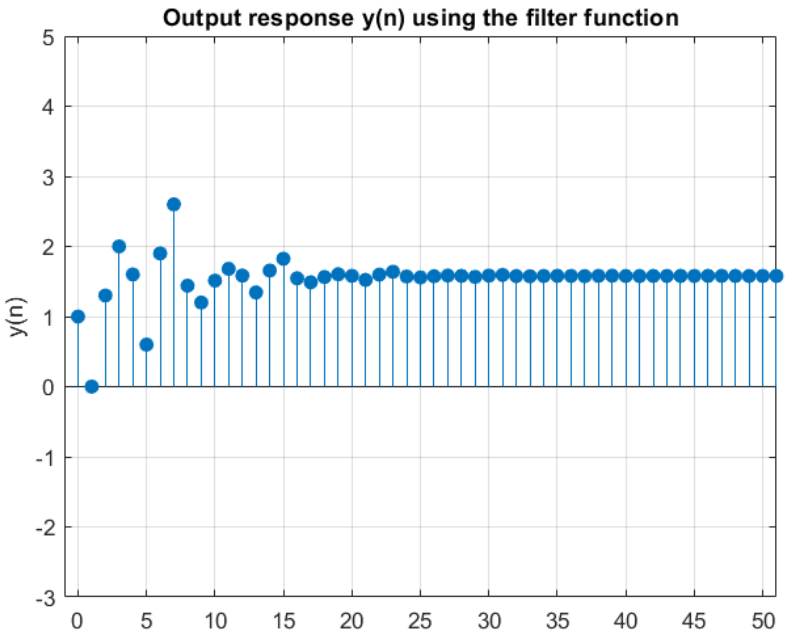
ورودی در حالت ماندگار به صورت $1+cos(\frac{\pi n+\pi}{2})$ پس برای بدست آوردن پاسخ حالت ماندگار[۳] باید مقدار پاسخ فرکانسی را تنها در فرکانس های $\omega = 0$ و $\omega = 0.5\pi$ بدست آوریم.

$$H(e^{j0}) = \frac{\sum_{l=0}^{3} 1}{1 + \sum_{m=1}^{3} (0.7)^m} = \frac{4}{1 + 0.7 + 0.49 + 0.343} = \frac{4}{2.533} = 1.58$$

[۳] response steady-state

$$H(e^{j0.5\pi}) = \frac{\sum_{l=0}^{3} e^{-j\pi\ell}}{1 + \sum_{m=1}^{3} e^{-j\pi m}(0.7)^m} = 0$$

بنابراین پاسخ جالت ماندگار برابر با $y[n] = 1.58$ می شود.
با شبیه سازی در متلب نیز به کمک تابع filter می توانیم نتیجه زیر را به دست آوریم: همانطور که



شکل ۶: پاسخ حالت ماندگار به ازای سیگنال ورودی اول

مشاهده می شود مقدار نهایی تقریبا برابر 1.58 است که با محاسبات تئوری نیز سازگار است.

۲.

$$x_2[n] = \sum_{\ell=0}^{5} (\ell + 1) \cos\left(\frac{\pi \ell n}{4}\right)$$

برای $x_2[n]$ نیز با توجه به ورودی باید مقدار پاسخ فرکانسی را در فرکانس های $\omega = \frac{\pi\ell}{4}$ به ازای $\ell = 0, 1, 2, 3, 4, 5$ بدست آوریم:

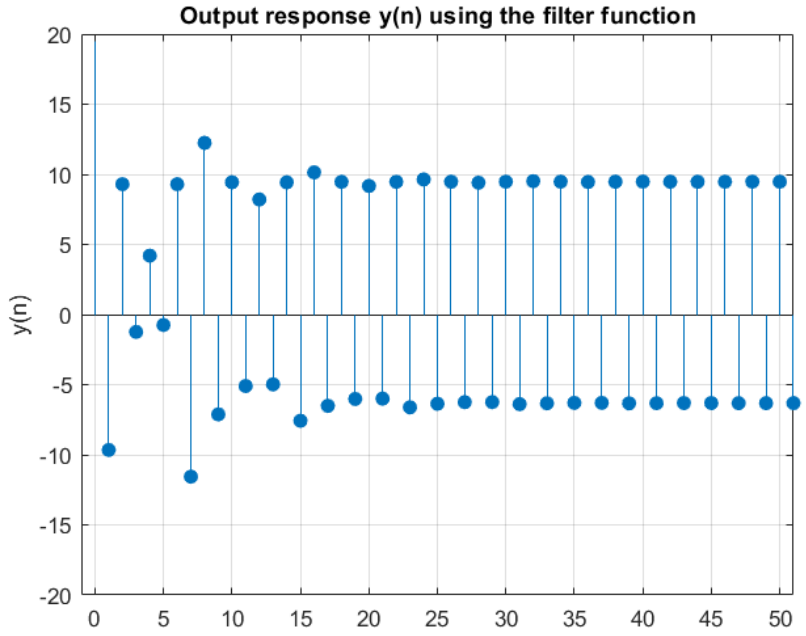
$$H(e^{j0}) = H(e^{j\pi}) = 1.58$$

$$H(e^{j0.25\pi}) = H(e^{j0.5\pi}) = H(e^{j0.75\pi}) = H(e^{j1.25\pi}) = 0$$

و در نهایت:

$$y[n] = 1.58 + 1.58 \times 5cos(\pi n) = 1.58 + 7.9cos(\pi n)$$

با شبیه سازی در متلب نیز به کمک تابع filter می توانیم نتیجه زیر را به دست آوریم:



شکل ۷: پاسخ حالت ماندگار به ازای سیگنال ورودی دوم

همانطور که مشاهده می شود حالت ماندگار به صورت متناوب و مطابق نتایج تئوری می باشد.

۲ درونیابی

همانطور که در درس دیدیم بلوک فشرده ساز [۴] نمونه های با توجه به ضریب داده شده از سیگنال برمیدارد و بقیه نمونه های سیگنال را دور می ریزد.

$$y[n] = x[nM]$$

۱.۲ پیاده سازی بلوک فشرده ساز

برای پیاده سازی در متلب تنها کافی است توجه کنیم که این بلوک، مقادیر سیگنال در غیر از ضرایب M را دور می ریزد.

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

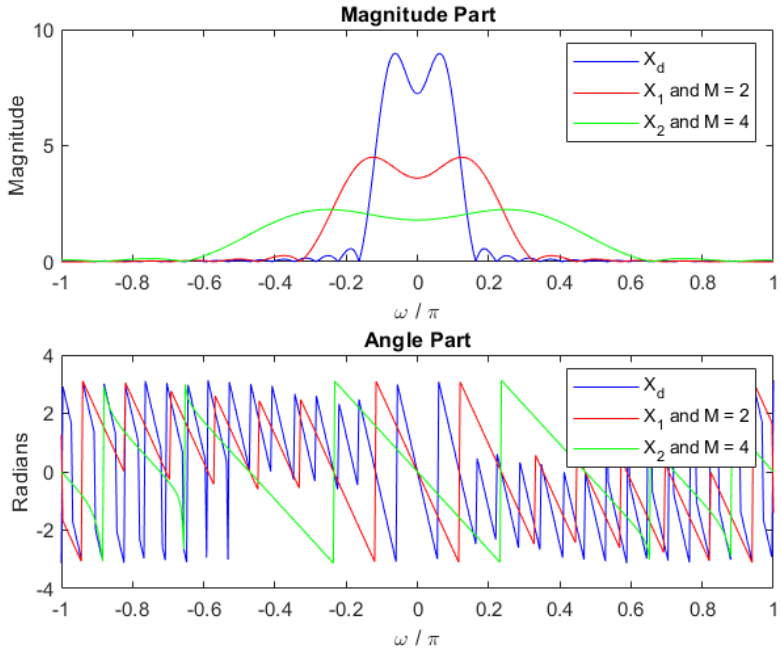
[۴] Compressor

برای نمونه برداری با فرکانس $8\ Hz$ کافی است تا در هر $\frac{1}{8}\ s$ از سیگنال نمونه برداری کنیم.

$$x_d[n] = x(nT_s)$$

$$Y(e^{j\omega}) = \frac{1}{M}\sum_{i=0}^{M-1} X\left(e^{j(\frac{\omega}{M} - \frac{2\pi i}{M})}\right)$$

پاسخ فرکانسی این سیگنال نیز، با رسم در متلب به صورت زیر بدست می آید.



شکل ۸: پاسخ فرکانسی سیگنال خروجی را به ازای هر $M \in \{2, 4\}$ به همراه پاسخ فرکانسی سیگنال $x_d[n]$

توجه شود که اختلاف نمودار با نمودار فیلتر پایین گذر ایده آل [۵] بدلیل محدود در نظر گرفتن سیگنال $sinc(n)$ می باشد.
با مقایسه نمودار های شکل ۸ می توان دید که همانطور که انتظار داریم فشرده ساز در حوزه فرکانس، پهن کننده فرکانس است و فرکانس سیگنال در M ضرب می شود.

[۵] Ideal Low-Pass Filter

۲.۲ بلوک باز کننده

بلوک باز کننده [۶] بین هر L نمونه سیگنال 0 اضافه می کند و رابطه آن به شکل زیر است:

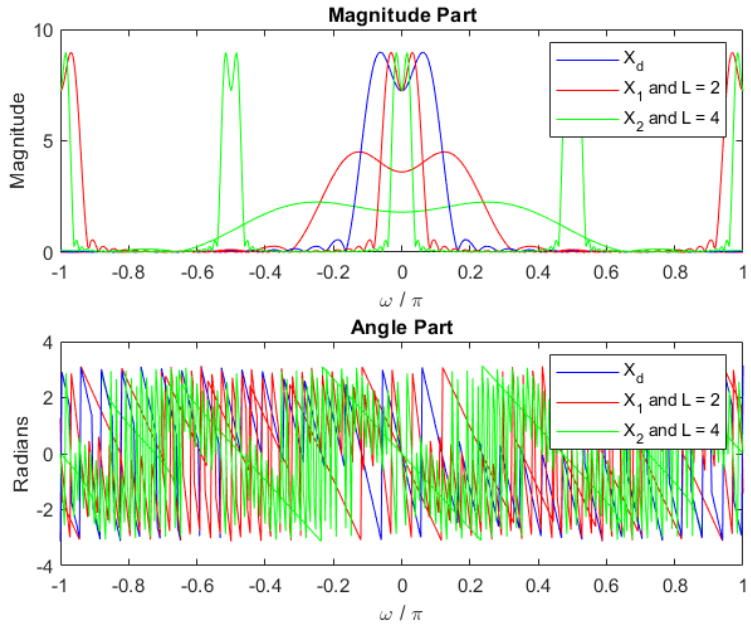
$$y[n] = \begin{cases} x\left[\frac{n}{L}\right] & , \quad n = 0, \pm L, \pm 2L, \dots \\ 0, & \text{otherwise} \end{cases}$$

$$Y(e^{j\omega}) = X(e^{j\omega L})$$

با توجه به این نکته برای پیاده سازی در متلب داریم:

```
function y = expander(x,L)
  for n=[1:L*length(x)]
    if mod(n,L) == 0
      y(n) = x(n/L);
    else
      y(n) = 0;
    end
  end
end
```

مشابه قسمت قبل با نمونه برداری و رسم در متلب داریم:



شکل ۹: پاسخ فرکانسی سیگنال خروجی را به ازای هر $L \in \{2, 4\}$ به همراه پاسخ فرکانسی سیگنال $x_d[n]$

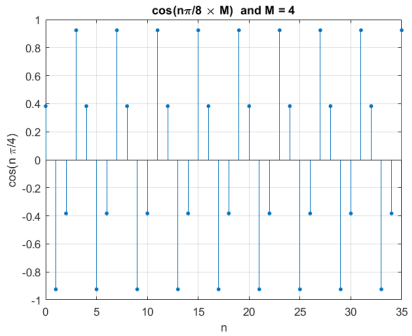
[۶] Expander

همانگونه که انتظار داریم تقسیم فرکانسی صورت می گیرد و سیگنال در حوزه فرکانس فشرده می شود. دوره تناوب سیگنال در حوزه فرکانس نیز کمتر می شود و باید از یک فیلتر برای رسیدن به سیگنال مطلوب استفاده کرد.

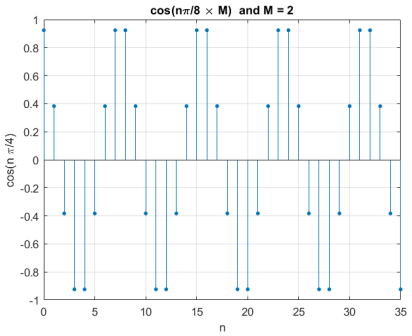
۳.۲ فشرده سازی سیگنال و مقایسه با decimate متلب

۱. با استفاده از روش بکار گرفته شده در قسمت اول داریم:

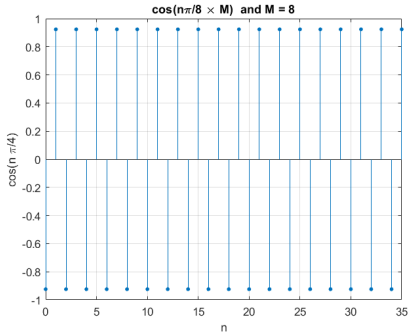
$$x[n] = cos(\frac{\pi n}{8})$$



(ب) سیگنال $x[n]$ فشرده شده به ازای $M = 4$



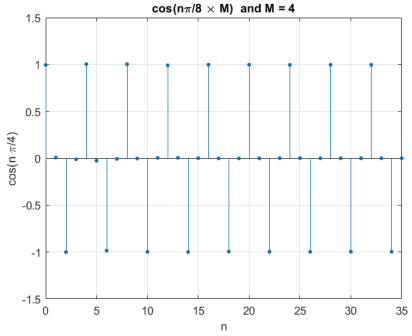
(آ) سیگنال $x[n]$ فشرده شده به ازای $M = 2$



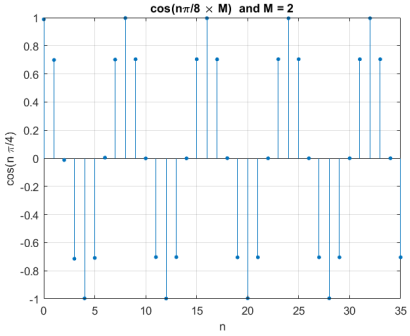
(ج) سیگنال $x[n]$ فشرده شده به ازای $M = 8$

شکل ۱۰: فشرده سازی $x[n]$ به ازای $M \in \{2, 4, 8\}$

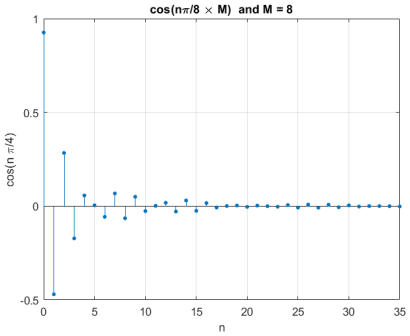
۲. حال این قسمت را با استفاده از تابع decimate متلب تکرار می کنیم.



(ب) سیگنال $x[n]$ فشرده شده به ازای $M = 4$



(آ) سیگنال $x[n]$ فشرده شده به ازای $M = 2$



(ج) سیگنال $x[n]$ فشرده شده به ازای $M = 8$

شکل ۱۱: فشرده سازی $x[n]$ به ازای $M \in \{2, 4, 8\}$ با تابع decimate

همانطور که مشاهده می شود خروجی این دو قسمت تا حدودی متفاوت است. در خصوص علت تفاوت آن ها می توان گفت که تابع decimate تنها به طور ساده نمونه هایی از سیگنال را دور نمی ریزد بلکه قبل از آن، نمونه ها از یک سیگنال پایین گذر عبور می کنند و سپس فشرده سازی می شود. این تفاوت به ازای $M = 8$ چشمگیر تر است زیرا پس از عبور از فیلتر پایین گذر با فرکانس قطع متناسب در تابع decimate سیگنال تقریبا 0 خواهد بود.

۴.۲ فیلتر درونیاب

در این بخش فیلتر ها را بررسی می کنیم.

۵.۲ بررسی انواع فیلتر ها

۱. **درونیاب ایده آل** در حقیقت یک فیلتر پایین گذر با بهره است که با تبدیل فوریه معکوس میتوانیم پاسخ ضربه این فیلتر را بدست آوریم:

$$H_1\left(e^{j\omega}\right) = L\,, \quad |\omega| \leq \frac{\pi}{L}$$

$$h_1[n] = \frac{1}{2\pi}\int_{-\frac{\pi}{L}}^{\frac{\pi}{L}} L\, d\omega = \frac{Lsin(\frac{n\pi}{L})}{n\pi} = sinc(\frac{n}{L})$$

ساخت این فیلتر ایده آل در عمل امکان پذیر نمی باشد، زیرا تابع sinc نامحدود است و هزینه ساخت فیلتر ایده آل بسیار گران و تقریبا ناممکن است. البته امکان ساخت تقریبی این فیلتر وجود دارد اما باز هزینه آن چشم گیر است.

$$x_a(t) = \sum_{n=-\infty}^{\infty} x[n]sinc(F_s(t - nT_s))$$

برای شبیه سازی در متلب از رابطه بالا با کمک بردار ها و ماتریس به شکل زیر عمل می کنیم.(در این کد زمان بزرگ تر t و زمان کوچکتر n می باشد و L نسبت نسبت فرکانس اولیه به ثانویه است. (

```
function y = ideal_interpolation(x,Ts,t)
  n = 0: length(x) - 1;
  Fs = 1/Ts;
  nTs = n *Ts;
  y = x*sinc(Fs*(ones(length(n),1)*t-nTs'*ones(1,length(t))))...
    ;
end
```

۲. **درونیاب خطی(مرتبه ۱)** در حقیقت نمونه های هر سیگنال را با یک چند جمله ای خطی به هم وصل می کند و پاسخ ضربه آن مطابق زیر می باشد.

$$h_2[n] = \begin{cases} 1 - \frac{|n|}{L}, |n| \leq L \\ 0, \quad \text{otherwise} \end{cases}$$

این درونیاب عملی می باشد و تقریب خوبی از پاسخ ارائه می دهد اما برای افزایش دقت بهتر است درونیابی با مرتبه های بالاتر انجام گیرد. درونیابی های مرتبه بالاتر هزینه بیشتری نیز خواهند داشت. شبیه سازی این تابع نیز در متلب به کمک تعریف تابع انجام می شود.

```
function [y] = linear_interpolation (x,t,L)
  n = 0: length(x) - 1;
  xl = expander(x,L);
  n = 0:length(t) - 1;
  x_lin = [zeros(1,L-1) xl zeros(1,L-1)];
  n_lin = [-L+1:0 n length(n):length(n)+L-1];
  y = zeros(1,length(n));
  for i = 0:length(t) - 1
    y(i+1) = sum(x_lin((i+1):(i+2*L-1)).*(1-abs(i-n_lin((i+1)...
    :(i+2*L-1)))/L));
  end
end
```

۳. **درونیاب مرتبه ۰** این درونیاب به صورت ساده تنها در قطار ضربه، ضرب می کند و سیگنال ها را عبور می دهد. رابطه این درونیاب مطابق زیر به سادگی نوشته می شود.

$$h_3[n] = \begin{cases} 1 & 0 \leq n < L \\ 0, & \text{otherwise} \end{cases}$$

این درونیاب عملی، تقریبی با دقت کمتری از درونیاب خطی می باشد و در کار های با دقت بالا نمی تواند بکار گرفته شود اما هزینه ساخت آن کمتر است. در حقیقت این روش مقدار تابع را در نقاط میانی برابر با آخرین سمپل قرار می دهد.

$$\hat{x}_a(t) = x[n] \quad , \quad nT_s \leq n \leq (n+1)T_s$$

```
function y = zero_order_interpolation(x,t,L)
  n = 0: length(x) - 1;
  xl = expander(x,L);
  y = zeros(1,length(n));
  for i = 0:length(t) - 1
    if xl(i + 1) ≠ 0
      y(i+1) = xl(i + 1);
    else
      y(i+1) = y(i);
    end
  end
end
```

۴. **درونیاب اسپلاین معکبی** به صورت معمول تقریب بهتری از 2 مورد قبل می باشد، که رابطه آن به صورت زیر نوشته می شود:

$$h_4[n] = \begin{cases} (a+2)|n/L|^3 - (a+3)|n/L|^2 + 1 & 0 \leq n \leq L \\ a|n/L|^3 - 5|n/L|^2 + 8a|n/L| - 4a & L \leq n \leq 2L \\ 0, & \text{otherwise} \end{cases}$$

در این روش نقاط میان هر سمپل با یک چند جمله ای درجه 3 درونیابی خواهند شد. ضرایب این چند جمله ای با تقریب حداقل مربعات بدست می آید. این درونیاب یک روش عملی و کاربردی برای درونیابی می باشد. شبیه سازی این تابع نیز در متلب به کمک تعریف تابع انجام می شود.

```
function y = spline_interpolation(x,t,Ts)
  n = 0: length(x) - 1;
  nTs = n*Ts;
  y = spline(nTs,x,t);
end
```

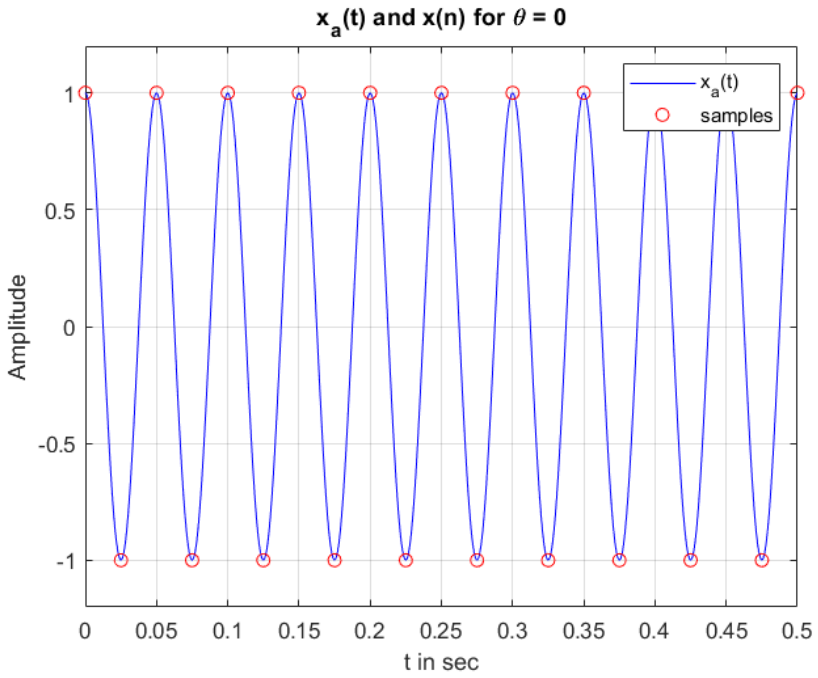
۶.۲ سمپلینگ از سیگنال آنالوگ

$$x_a(t) = cos(40\pi t + \theta)\ ,\ 0 \geq t \geq 0.5$$

نمونه برداری و گام های خواسته شده را به ازای دو θ انجام می دهیم.

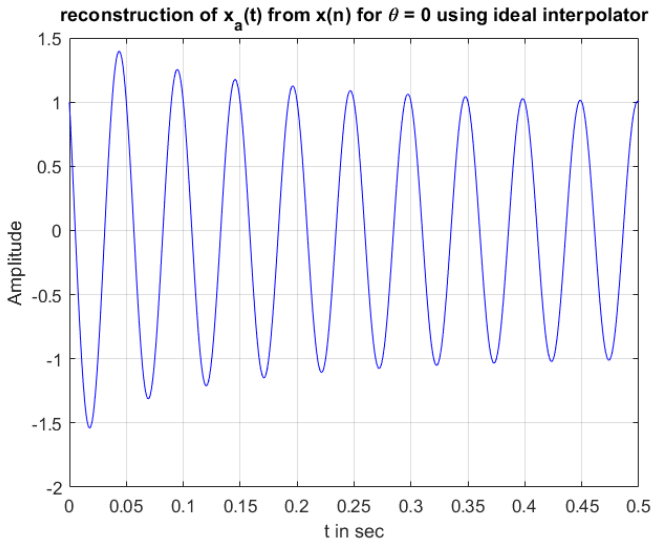
۱. $\theta = 0$

(آ) با نمونه برداری با $F_s = 40\,Hz$ ، 51 سمپل از سیگنال تقریبا آنالوگ نمونه بر می داریم که در شکل زیر نشان داده می شود. نقاط قرمز بیانگر سمپل های انتخاب شده است.

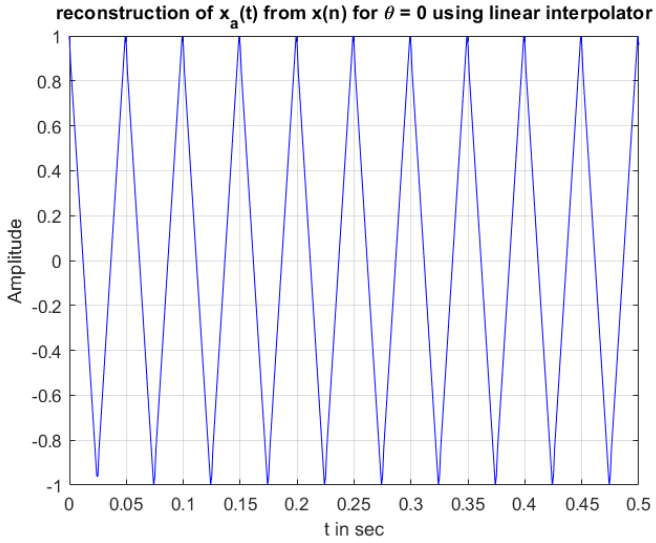


شکل ۱۲: سیگنال نمونه برداری شده

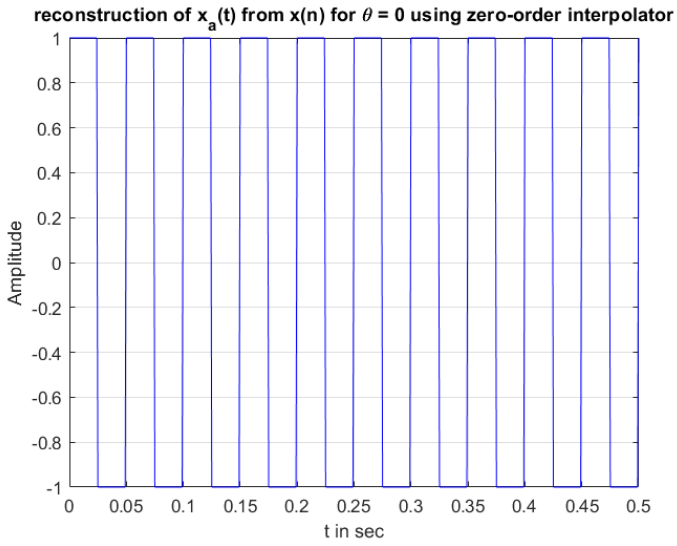
(ب) درونیابی برای هر سیگنال را مطابق زیر انجام می دهیم.



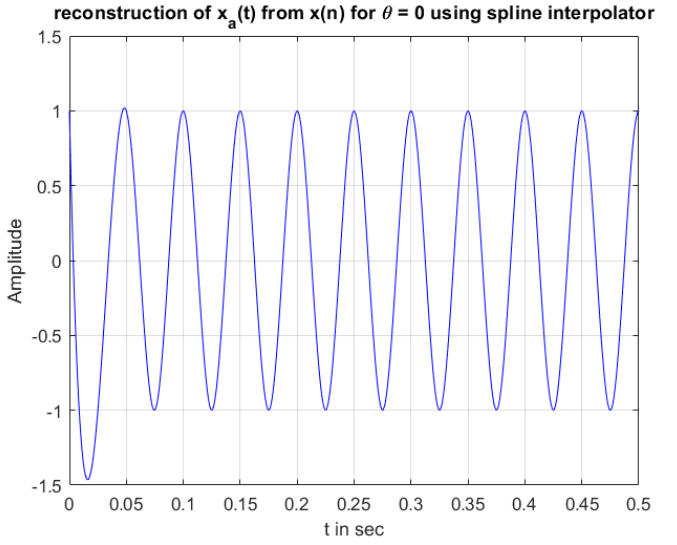
شکل ۱۳: سیگنال درون یابی شده با درونیاب ایده آل



شکل ۱۴: سیگنال درون یابی شده با درونیاب خطی



شکل ۱۵: سیگنال درون یابی شده با درونیاب مرتبه 0



شکل ۱۶: سیگنال درون یابی شده با درونیاب spline

اگر به شکل سیگنال ها توجه شود در ابتدای سیگنال بازسازی شده یک اختلاف با سیگنال اصلی وجود دارد. شکل نمودار در درونیابی مرتبه 0 و مرتبه 1 دچار تغییرات چشمگیری می شود.
برای بدست آوردن خطای حداقل مربعات درونیابی و مقایسه آن از تابع immse استفاده می کنیم.

```
MSE for theta = 0

MSE for sinc interpolation: 0.168096
MSE for linear interpolation: 0.028683
MSE for zero order interpolation: 1.417166
MSE for spline interpolation: 0.056610
```

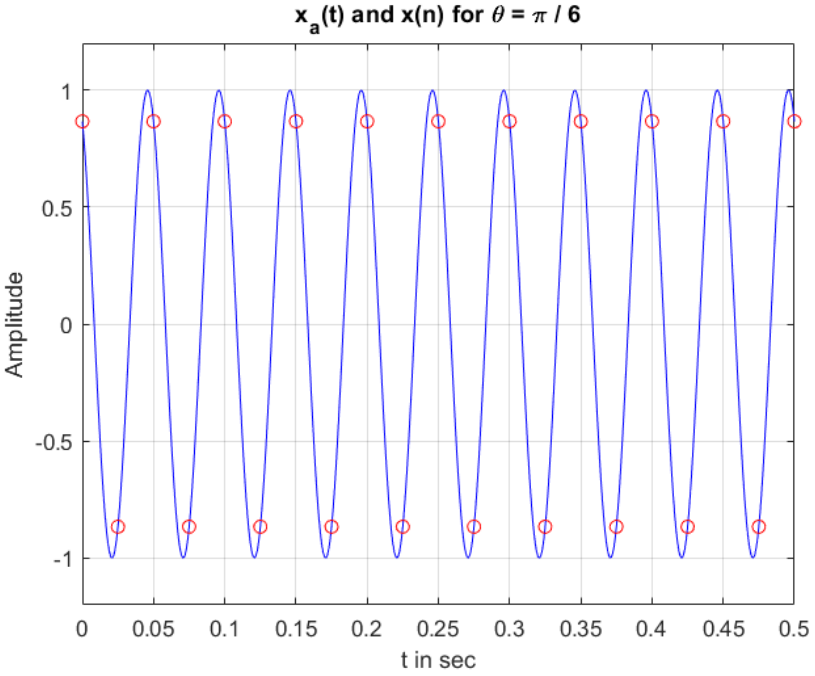
شکل ۱۷: خطای حداقل مربعات هر درونیاب به ازای $\theta = 0$

(ج) در این مورد خطای درونیاب خطی از همه کمتر است. اما اگر در شکل توجه کنید سیگنال بازسازی شده به خوبی نمی تواند شکل سینوسی سیگنال اولیه را بازسازی کند. در بیشتر اوقات زمانی که تعداد نمونه های گسسته زیاد باشد، تقریب اسپلاین مکعبی برای بازیابی مناسب تر است.

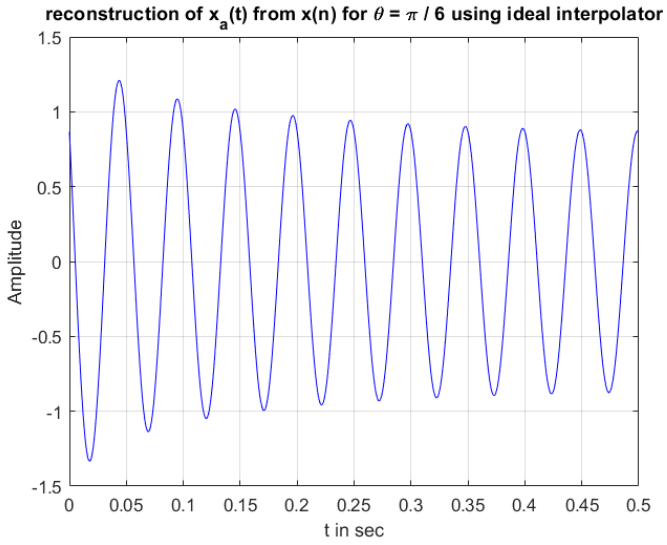
$$\theta = \frac{\pi}{6}$$

۲. (آ) شیوه مورد قبل را تکرار می کنیم. با نمونه برداری با $F_s = 40\ Hz$ ، 51 سمپل از سیگنال تقریبا آنالوگ نمونه بر می داریم که در شکل زیر نشان داده می شود. نقاط قرمز بیانگر سمپل های انتخاب شده است.

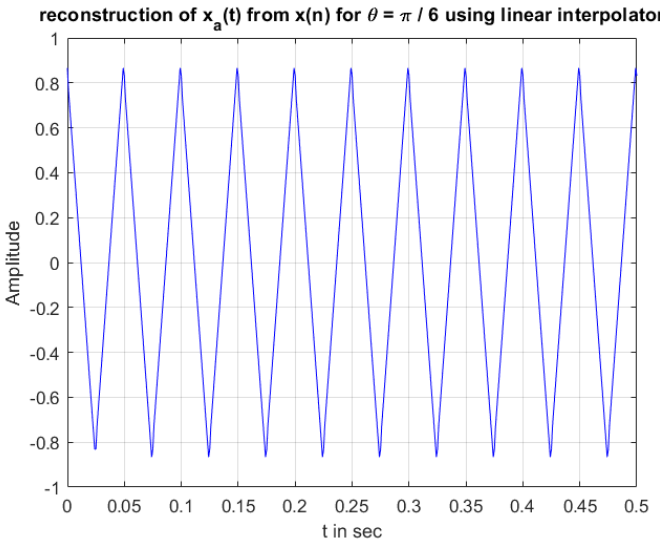
(ب) درونیابی برای هر سیگنال را مطابق زیر انجام می دهیم.



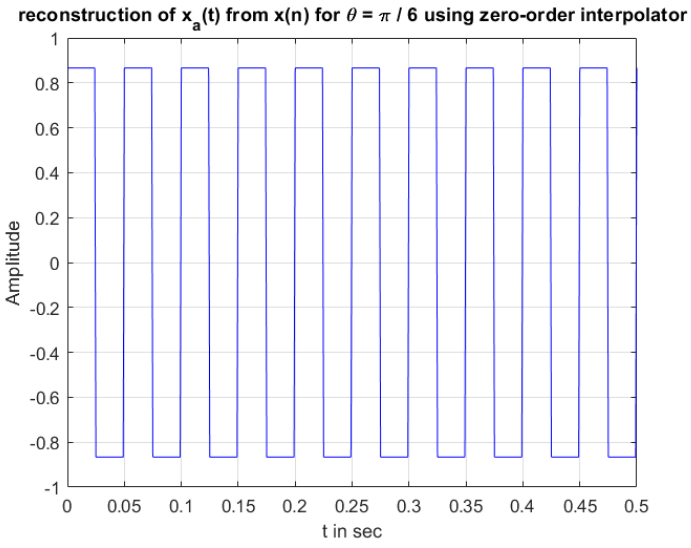
شکل ۱۸: سیگنال نمونه برداری شده



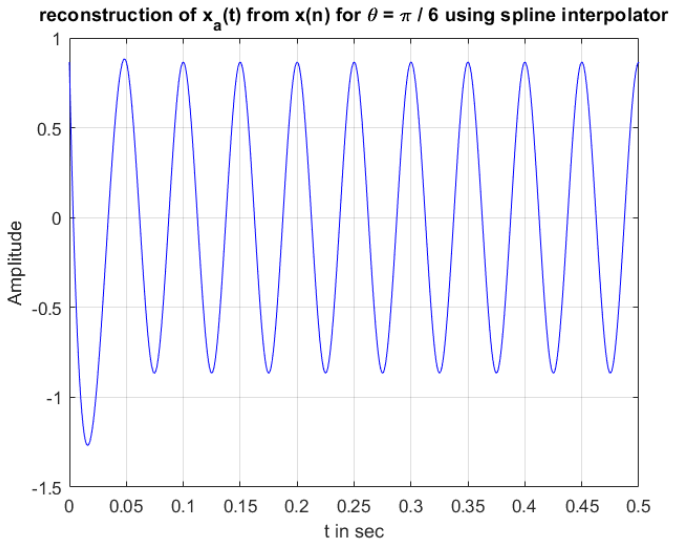
شکل ۱۹: سیگنال درون یابی شده با درونیاب ایده آل



شکل ۲۰: سیگنال درون یابی شده با درونیاب خطی



شکل ۲۱: سیگنال درون یابی شده با درونیاب مرتبه 0



شکل ۲۲: سیگنال درون یابی شده با درونیاب spline

اگر به شکل سیگنال ها توجه شود در ابتدای سیگنال بازسازی شده یک اختلاف با سیگنال اصلی وجود دارد. شکل نمودار در درونیابی مرتبه 0 و مرتبه 1 دچار تغییرات چشمگیری می شود. همچنین سیگنال بازسازی شده به مقدار 1 و -1 نمی رسد.

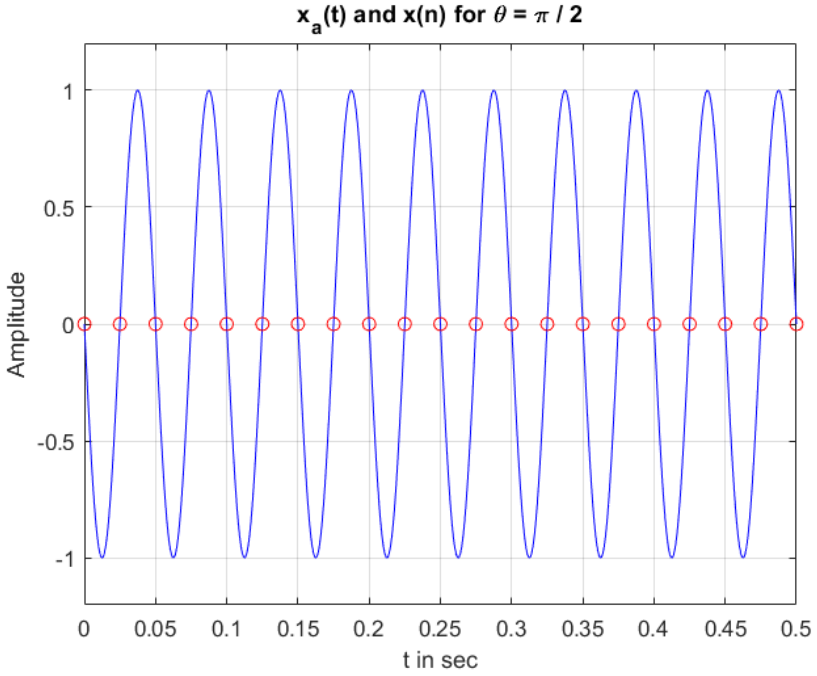
```
MSE for theta = pi / 6

MSE for sinc interpolation: 0.034535
MSE for linear interpolation: 0.103402
MSE for zero order interpolation: 1.737129
MSE for spline interpolation: 0.125596
```

شکل ۲۳: خطای حداقل مربعات هر درونیاب به ازای $\theta = \frac{\pi}{6}$

(ج) در این مورد خطای درونیاب، درونیاب ایده آل از همه کمتر است. تفاوت این قسمت آن است که نمونه برداری در نقاط ماکزیمم و مینیمم گرفته نشده است.

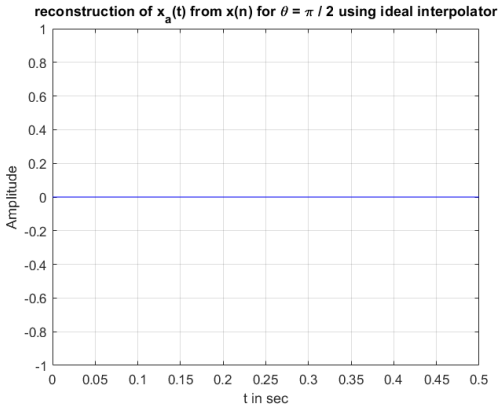
۳. $\theta = \frac{\pi}{6}$

(آ) شیوه مورد قبل را تکرار می کنیم. با نمونه برداری با $F_s = 40\ Hz$ ، 51 سمپل از سیگنال تقریبا آنالوگ نمونه بر می داریم که در شکل زیر نشان داده می شود. نقاط قرمز بیانگر سمپل های انتخاب شده است.

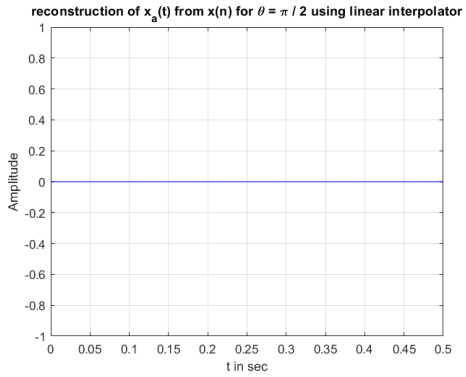


شکل ۲۴: سیگنال نمونه برداری شده

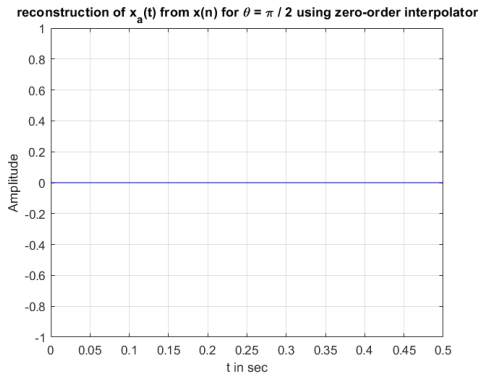
(ب) درونیابی برای هر سیگنال را مطابق زیر انجام می دهیم.



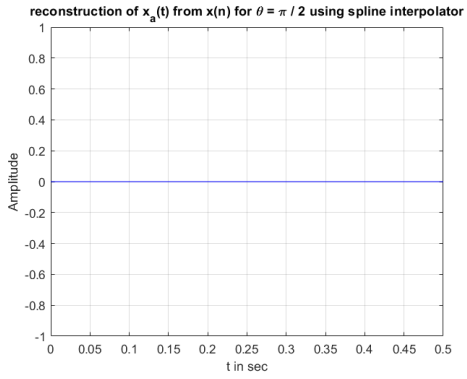
شکل ۲۵: سیگنال درون یابی شده با درونیاب ایده آل



شکل ۲۶: سیگنال درون یابی شده با درونیاب خطی



شکل ۲۷: سیگنال درون یابی شده با درونیاب مرتبه 0



شکل ۲۸: سیگنال درون یابی شده با درونیاب spline

```
MSE for theta = pi / 2

MSE for sinc interpolation: 0.499002
MSE for linear interpolation: 0.499002
MSE for zero order interpolation: 0.499002
MSE for spline interpolation: 0.499002
```

شکل ۲۹: خطای حداقل مربعات هر درونیاب به ازای $\theta = \frac{\pi}{2}$

(ج) در این مورد خطای درونیاب ها یکسان است زیرا از نقاطی نمونه برداشته شده که همگی مقدار 0 دارند.

(د) همانطور که مشاهده می شود مقدار سیگنال بازسازی شده به شدت به میزان فاز θ وابستگی دارد. به صورتی که در فاز $\theta = \frac{\pi}{6}$ سیگنال به مقدار 1 یا -1 نخواهد رسید و اندازه آن کمتر خواهد بود. در فاز $\theta = \frac{\pi}{2}$ نیز چون تمامی نقاط نمونه برداری شده دارای اندازه 0 هستند، با درونیابی به نتیجه مناسبی نمی توان رسید. به صورت کلی اگر سیگنال به ازای $f = 2\ samples\,per\,cycle$ نمونه برداری شود(مانند این مسئله)، سیگنال بازسازی شده نهایی به فاز θ وابسته خواهد بود. در این مورد بیشترین اندازه سیگنال تقریبا برابر $cos(\theta)$ می باشد که نشان دهنده این وابستگی است.

۳ تبدیل z

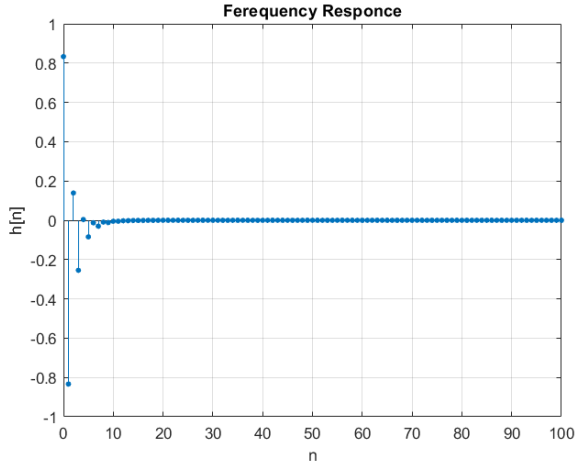
$$H(z) = \frac{Y(z)}{X(z)} = \frac{\frac{5}{6} - \frac{35}{36}z^{-1} - \frac{1}{2}z^{-2}}{1 - \frac{1}{6}z^{-1} - \frac{1}{3}z^{-2}}$$

$$y[n] - \frac{1}{6}y[n-1] - \frac{1}{3}y[n-2] = \frac{5}{6} - \frac{35}{36}x[n-1] - \frac{1}{2}x[n-2]$$

حالا به بررسی سیستم می پردازیم:

۱. پاسخ ضربه سیستم، همان خروجی سیستم به ازای ورودی $x[n] = \delta[n]$ می باشد. برای محاسبه آن روش iterative با شرایط اولیه ایستا را بکار می گیریم.

$$y[n] = \frac{5}{6} - \frac{35}{36}x[n-1] - \frac{1}{2}x[n-2] + \frac{1}{6}y[n-1] + \frac{1}{3}y[n-2]$$

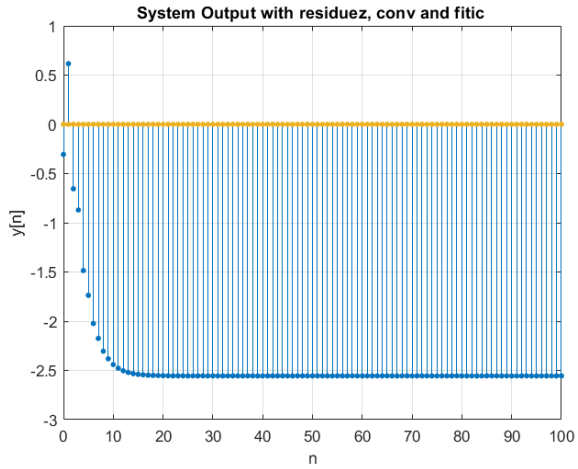


شکل ۳۰: پاسخ ضربه سیستم به ازای $0 \geq n \geq 100$

۲.

$$x[n] = (2 - \frac{4}{5}(\frac{1}{2})^n)$$

در این قسمت به کمک توابع residuez conv، و filtic به ازای $n \geq 0$ پاسخ کامل سیستم را محاسبه می کنیم.

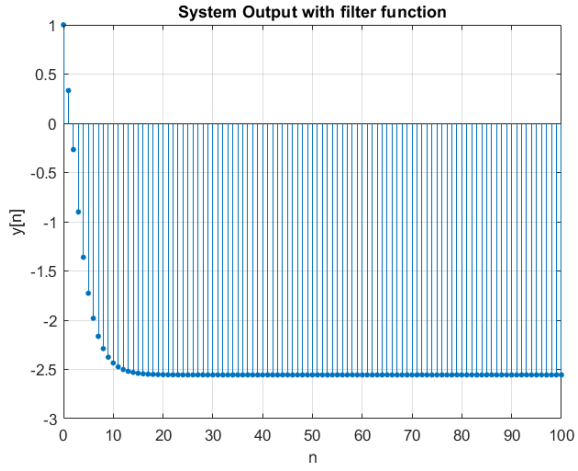


شکل ۳۱: پاسخ کامل سیستم به ازای

همچنین با توجه به مقادیر خروجی توابع می توان نوشت:

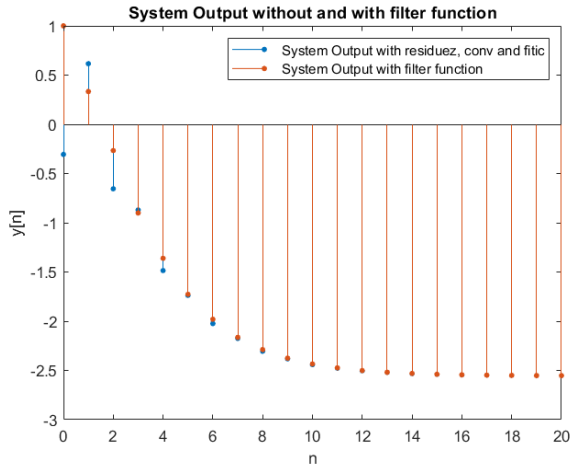
$$y\left[n\right]=-2.5556\ u\left[n\right]+6.8825\left(\frac{2}{3}\right)^{n}u\left[n\right]-0.8992\left(-\frac{1}{2}\right)^{n}u\left[n\right]-3.7333\left(\frac{1}{2}\right)^{n}u\left[n\right]$$

۳. به کمک تابع filter نیز به سادگی پاسخ سیستم را محاسبه می کنیم:



شکل ۳۲: پاسخ کامل سیستم به ازای با کمک تابع filter

برای مقایسه با خروجی قبل برای ۲۰ نمونه اول نیز داریم:



شکل ۳۳: مقایسه خروجی سیستم در روش ۱ و ۳

همانطور که در شکل مشخص است، خروجی سیستم به ازای ۲ روش تقریبا یکسان است.

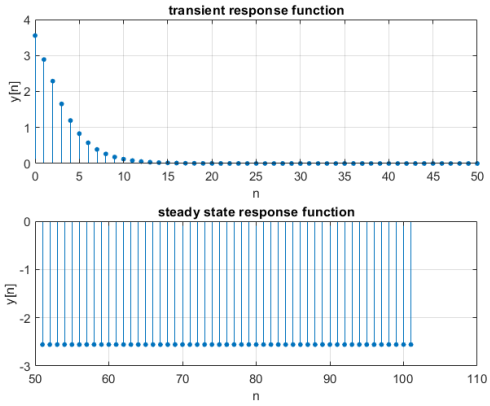
۴. با توجه به خروجی بدست آمده، پاسخ گذرا و ماندگار را جدا می کنیم.

(آ) **پاسخ حالت ماندگار** :

$$y[n] = -2.5556\, u[n]$$

(ب) **پاسخ حالت گذرا** :

$$y[n] = 6.8825 \left(\frac{2}{3}\right)^n u[n] - 0.8992 \left(-\frac{1}{2}\right)^n u[n] - 3.7333 \left(\frac{1}{2}\right)^n u[n]$$



شکل ۳۴: پاسخ گذرا و حالت ماندگار سیستم