

به نام خدا

عرفان رفیعی اسکویی - 98243027

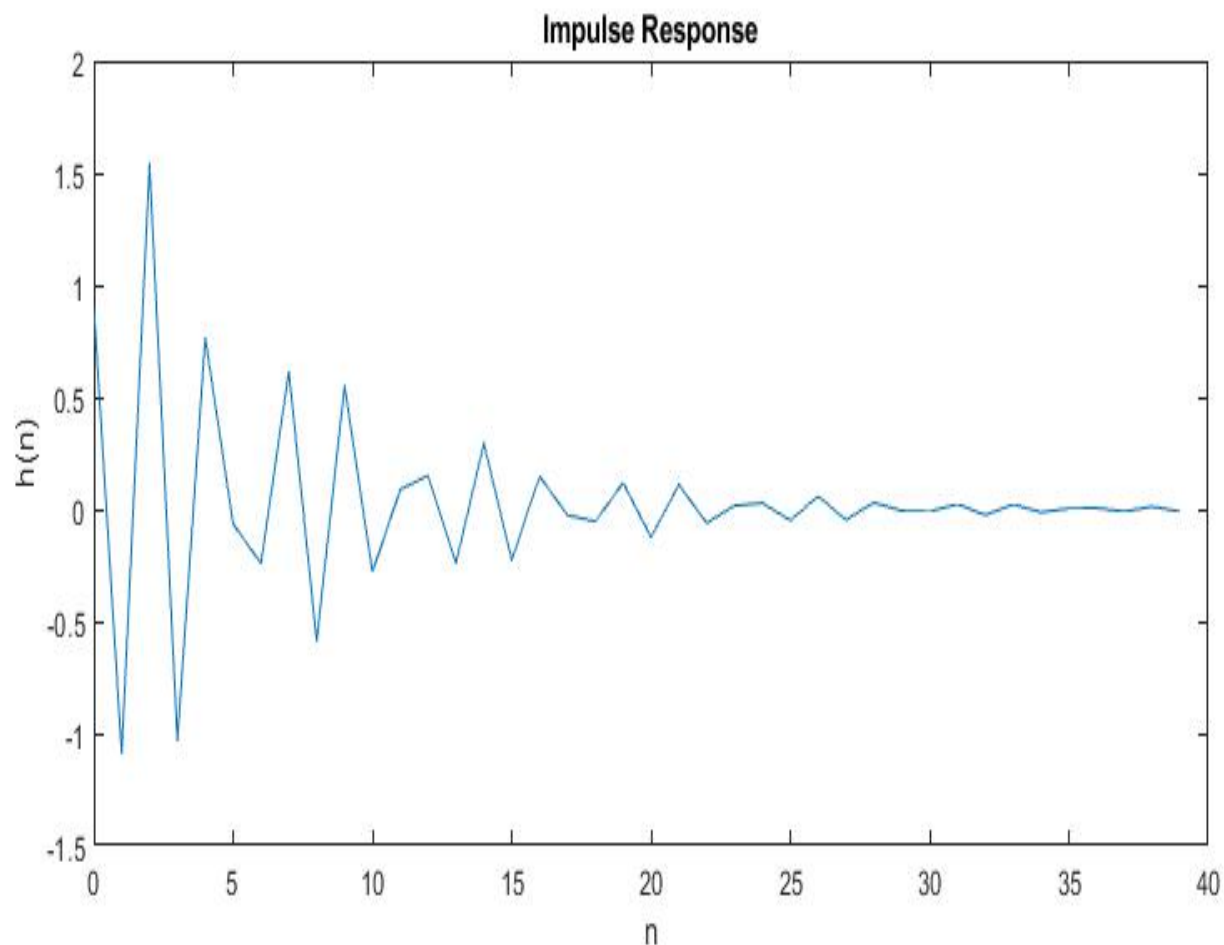
سوال اول

با استفاده از دستور `Impz()` میتوان ضرایب صورت و مخرج یک فیلتر گسسته را داد و در خروجی ضرایب فیلتر در حوزه زمان را دریافت کرد. دستور `filter` نیز به همین منظور استفاده می شود.

برای اینکه پاسخ ضربه این سیستم را رسم کنیم ضرایب ورودی را در `b` و ضرایب خروجی را در متغیر `a` به تابع `impz()` به صورت زیر داده ایم:

```
[h t] = impz([0.9 -0.45 0.35 0.002], [1 0.71 -0.46 -  
0.62]);  
plot(t(1:40), h(1:40))  
title('Impulse Response')  
xlabel('n'); ylabel('h(n)')
```

که ضرایب فیلتر در متغیر `h` ذخیره شده و همانطور که خواسته شده ۴۰ سمپل ابتدایی آنرا به صورت زیر رسم نمودیم:



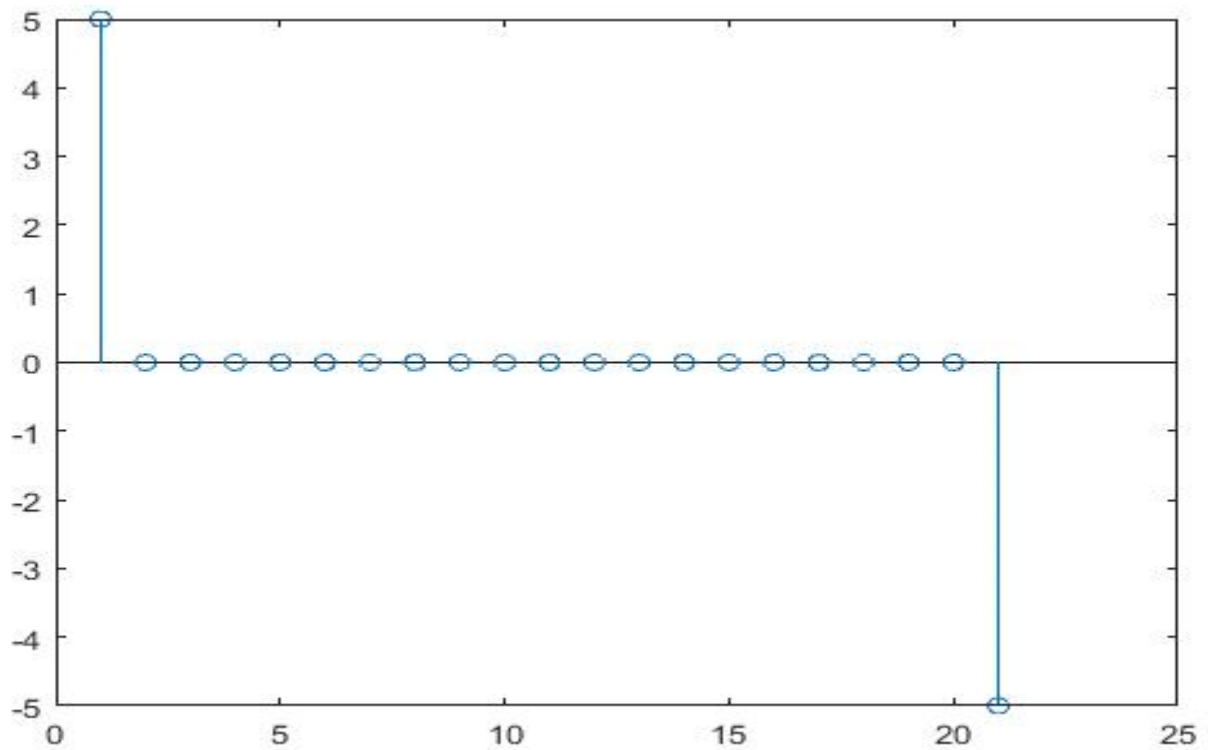
سوال دوم

در سوال دوم در قسمت اول سیگنالی که به ما داده شده دارای ۲۰ سمپل غیر صفر است که انها را در متغیر x ذخیره نمودیم. در حلقه `for` هر سمپل را از قبلی کم کردیم و در متغیر y ذخیره نمودیم که نتیجه ان به صورت زیر شد:

```
n = 1:21;
x = [5*ones(1,20) 0];
y(1)= x(1);
for i = 2:21

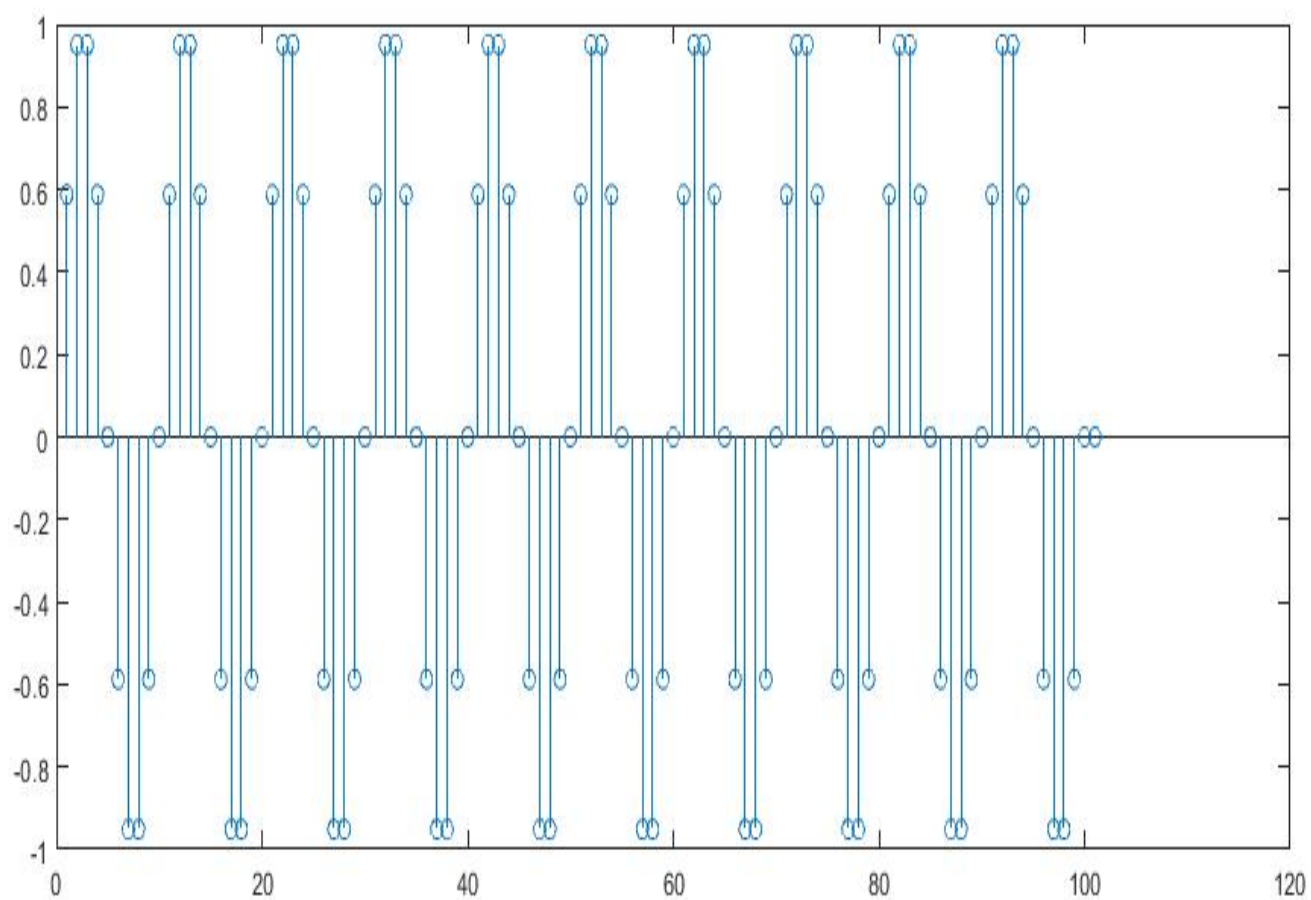
    y(i) = x(i)-x(i-1)
end
figure
```

`stem(n,y)`

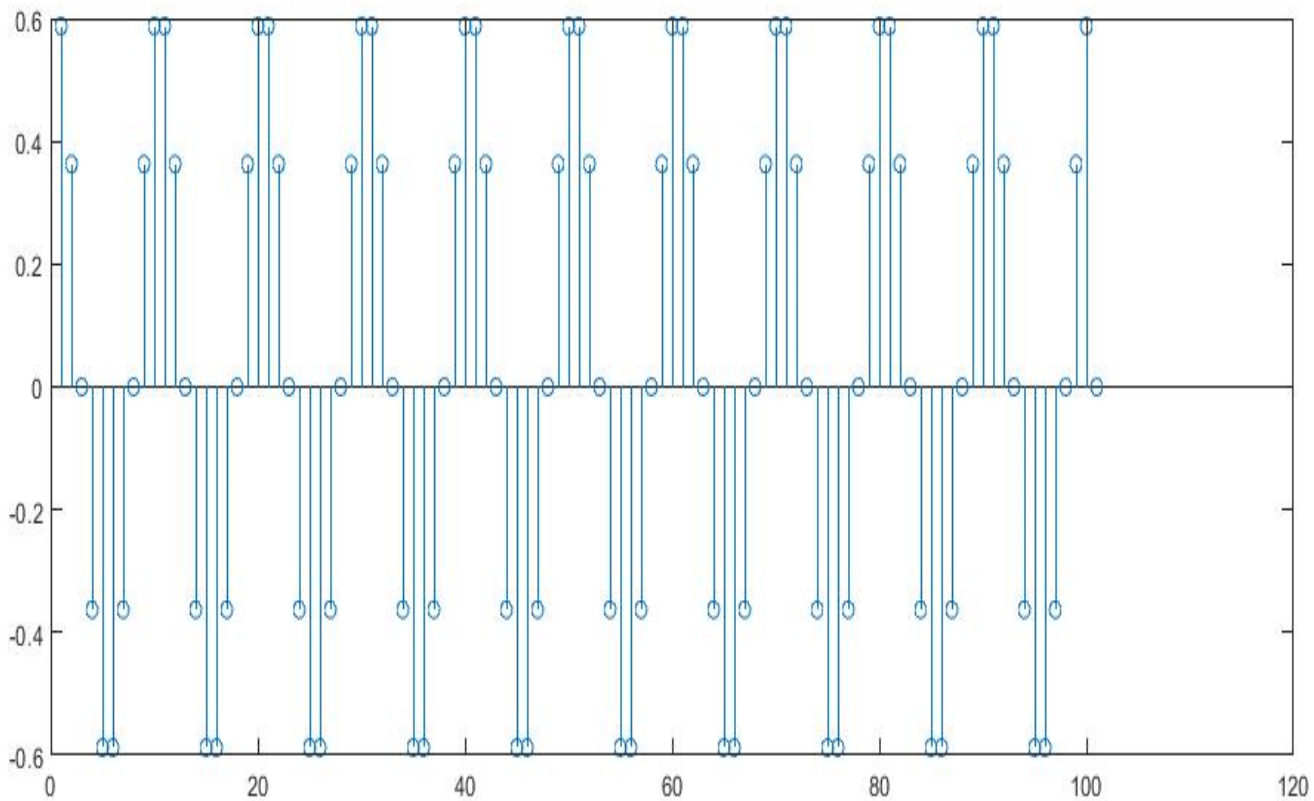


در قسمت دوم از ما خواسته شده سیگنال سینوسی با ۱۰۰ سمپل تولید کنیم و تفاضل گیر را رو آن پیاده کنیم. شکل ورودی تفاضل گیرنده به صورت زیر است:

```
n = 1:101;  
x = sin(n.*pi/5) .* [ones(1,100) 0];  
y(1)= x(1);  
  
for i = 2:101  
    y(i) = x(i)-x(i-1)  
end  
figure  
stem(n,y)
```



و شکل خروجی آن نیز:



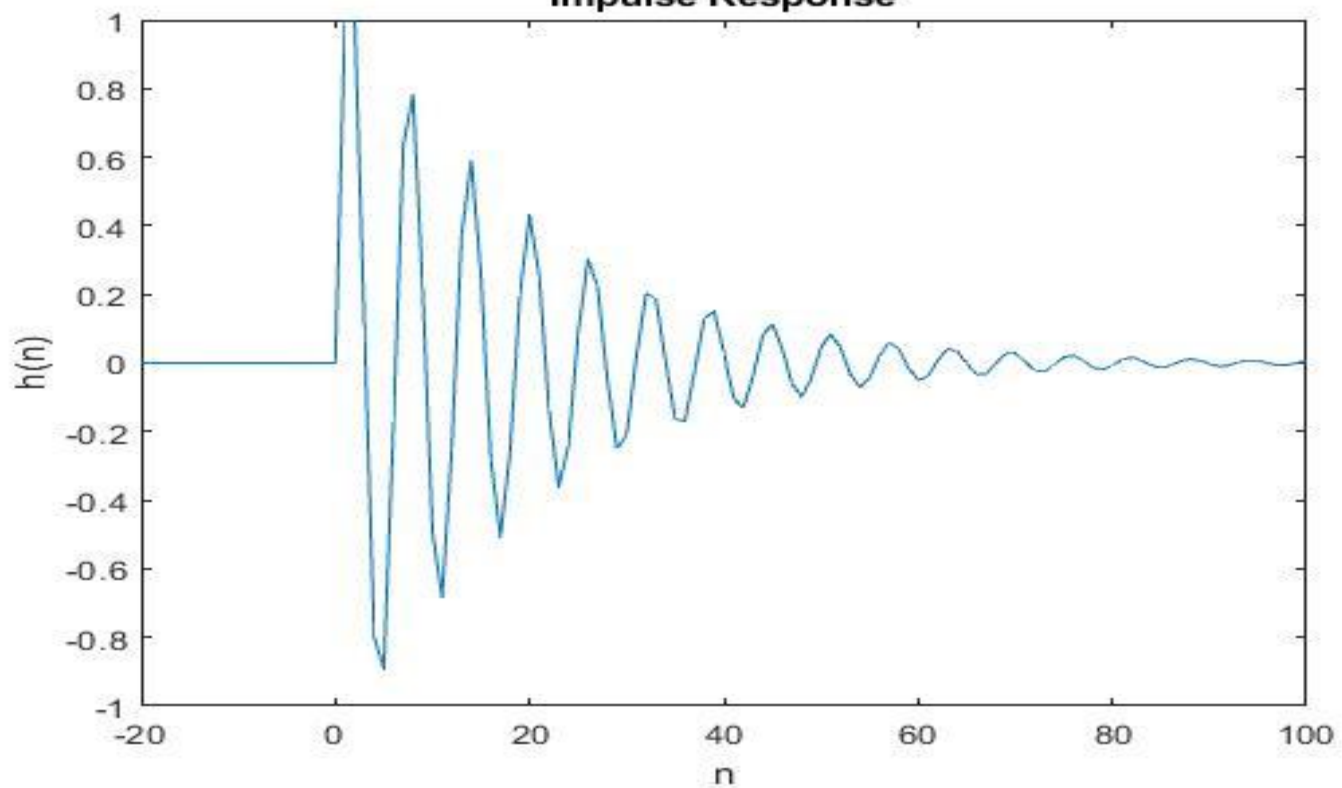
سوال سوم

در این سوال همانند سوال اول پاسخ ضربه را به دست می آوریم.

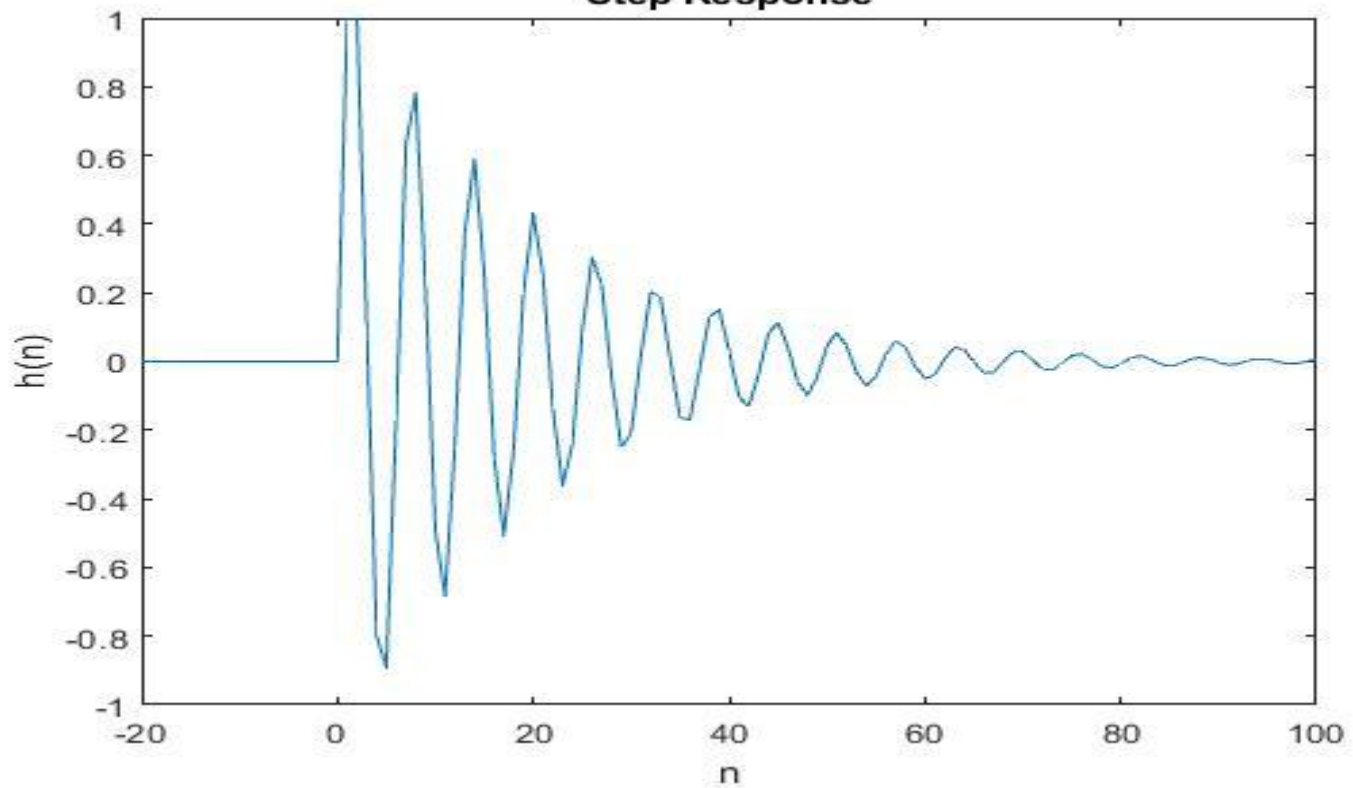
```
[h t] = impz([1],[1 -1 0.9]);
plot([-20:100],[zeros(1,21) h(1:100)'])
s(1)= h(1);
for i = 2:120
    s(i)= h(i)+s(i-1)
end
figure
plot(-20:100,[zeros(1,21) h(1:100)'])
```

و با استفاده از حلقه for و رابطه بین پاسخ پله و ضربه، پاسخ پله را نیز پیدا کردیم و در بازه خواسته شده رسم نمودیم.

Impulse Response



Step Response



چون پاسخ ضربه مقادیر منفی را در بر می‌گردد و حالت تقریباً مقارنی دارد باعث شده که شکل پاسخ نیز مشابه پاسخ ضربه باشد.