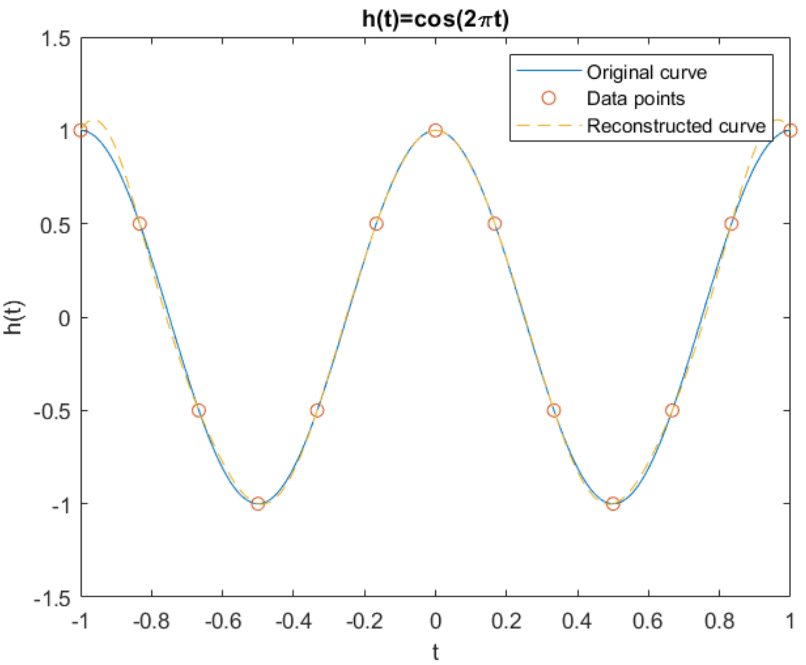
HW3 MATLAB code

1.

```
function h = hr(t, hn, N1, N2, T)
h = zeros(size(t));
n = N1:N2;
f = @(x) sum(hn.*sin(pi*(x-n*T)/T)./(pi*(x-n*T)/T));
for i = 1:length(t)
    h(i) = f(t(i));
end
```

2.

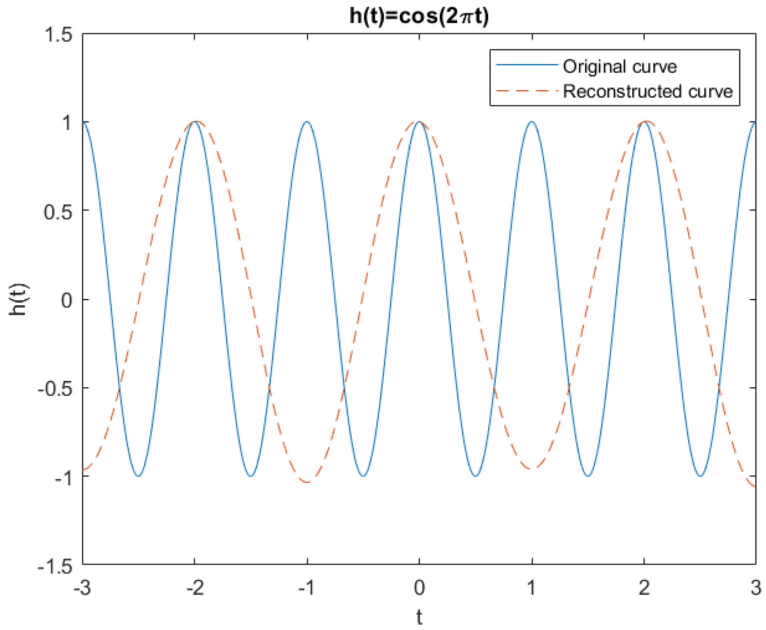
```
N1 = -6;
N2 = 6;
T = 1/6;
n = N1:N2;
td = n/6;
t = -1:0.001:1;
h = @(t) cos(2*pi*t);
hn = h(td);
hr = hr(t,hn,N1,N2,T);
plot(t,h(t));
hold on
plot(td,h(td),'o');
hold on
plot(t,hr,'--');
title('h(t)=cos(2\pit)');
xlabel('t'); ylabel('h(t)');
legend('Original curve', 'Data points', 'Reconstructed curve')
```



The constructed curve does not completely match the original curve, because the data values which between sampled data points h[n] are determined by interpolation of sinc function. If the number of h[n] goes larger, then the reconstructed curve will get closer to the original curve.

3.

```
N1 = -9;
N2 = 8;
T = 1/1.5;
n = N1:N2;
td = n/1.5;
t = -3:0.001:3;
h = @(t) cos(2*pi*t);
hn = h(td);
hr = hr(t,hn,N1,N2,T);
plot(t,h(t));
hold on
plot(t,hr,'--');
title('h(t)=cos(2\pit)');
xlabel('t'); ylabel('h(t)');
legend('Original curve', 'Reconstructed curve');
```



The sampling interval T_s is

$$T_s = \frac{1}{1.5} = \frac{2}{3}\ (\mathrm{s})$$

The sampling frequency f_s is the reciprocal of sampling interval T_s.

$$f_s = \frac{1}{T_s} = 1.5\ (\mathrm{Hz})$$

But the critical frequency of function h(t) is

$$f_c = 1\ (\mathrm{Hz})$$

Since $f_s < 2f_c$, aliasing happens.