

# Homework 1

16340305 郑先淇

## M2.6

Write a Matlab program to plot a continuous-time sinusoidal signal and its sampled version, and verify Figure 2.28. You need to use the hold function to keep both plots.

## 解答

思路：首先画出 sin 函数，然后使用采样函数对其进行采样。

代码如下：

```
编辑器 - C:\Users\admin\Desktop\大三下\Junoir_study_last\数字信号处理\作业\M2_6.m
M2_6.m x +
1 - f=1;Um=1;nt=2;%输入信号频率、振幅和显示周期数
2 - N = 8; %采样点数
3 - T = 1/f;% 信号周期
4 - dt = T/N; %采样时间间隔
5 - n = 0 : nt*N - 1; %建立离散信号的时间序列
6 - tn = n*dt;
7 - y = Um*sin(2*f*pi*t); %声明sin函数
8 - yn = Um*sin(2*f*pi*tn);%对原模拟信号进行采样
9 - t = 0:0.01:2;
10 - plot(t,y); %绘制函数
11 - hold; %锁定绘图
12 - axis([0,nt*T,min(y),max(y)]);%限定横坐标和纵坐标的显示范围
13 - title('f(t) = sin(2πt)'); %加标题
14 - xlabel('t'); %横坐标符号
15 - ylabel('f(t)'); %纵坐标符号
16 - stem(tn,yn);
```

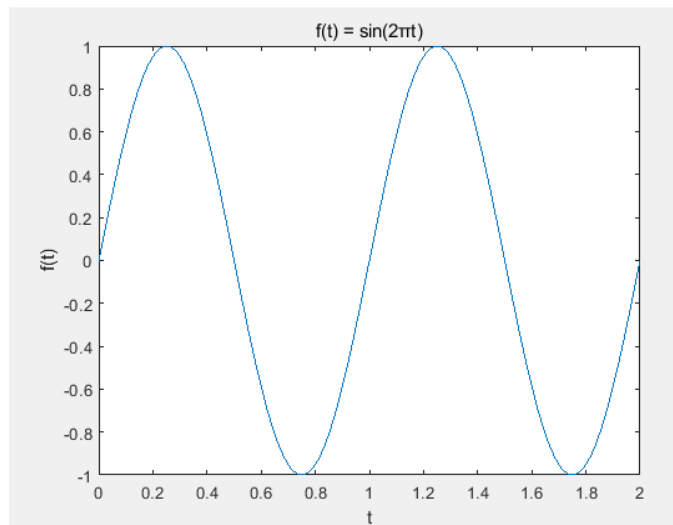
代码解析：

第一步先确定sin函数的自变量的范围： $t = 0:0.01:2;$

然后声明函数为 $y = Um \cdot \sin(2 \cdot f \cdot \pi \cdot t);$

通过 `plot(t,y)` 函数可以画出函数的图像，

使用 `title`、`xlabel`、`ylabel` 函数为图像添加标题和横纵坐标，实现效果如下：



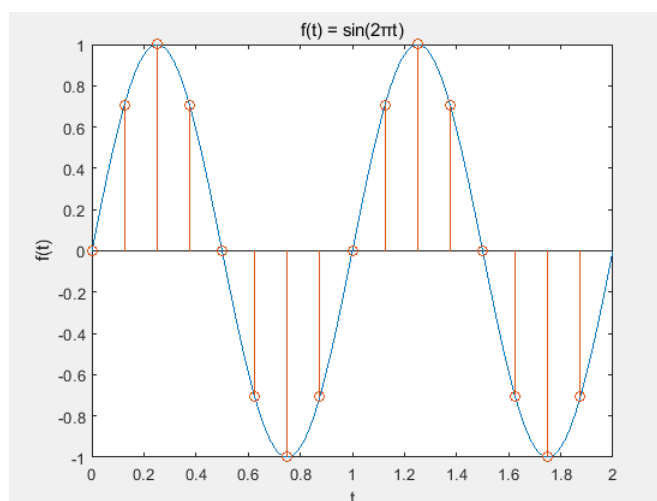
采样之前需要锁定已画好的 `sin` 函数的图,使用 `hold` 函数；  
然后进行采样操作，确定采样信号的频率(采样点数)，  
采样时间间隔，建立离散信号的时间序列：

```
2 - N = 8; %采样点数
3 - T = 1/f;% 信号周期
4 - dt = T/N; %采样时间间隔
5 - n = 0 : nt*N - 1; %建立离散信号的时间序列
6 - tn = n*dt;
```

声明采样信号的函数，绘制图像：

```
yn = Um*sin(2*f*pi*tn);
stem(tn,yn);
```

结果如下：



## M2.7

Using the program developed in the previous problem, verify experimentally that the family of continuous-time sinusoids given by Eq.(2.65) lead to identical sampled signals.

## 解答

题目所求即为验证下式：

$$x_{a,k}(t) = A \cos(\pm \Omega_0 t + \phi) + k \Omega_T t), k = 0, \pm 1, \pm 2, \dots$$

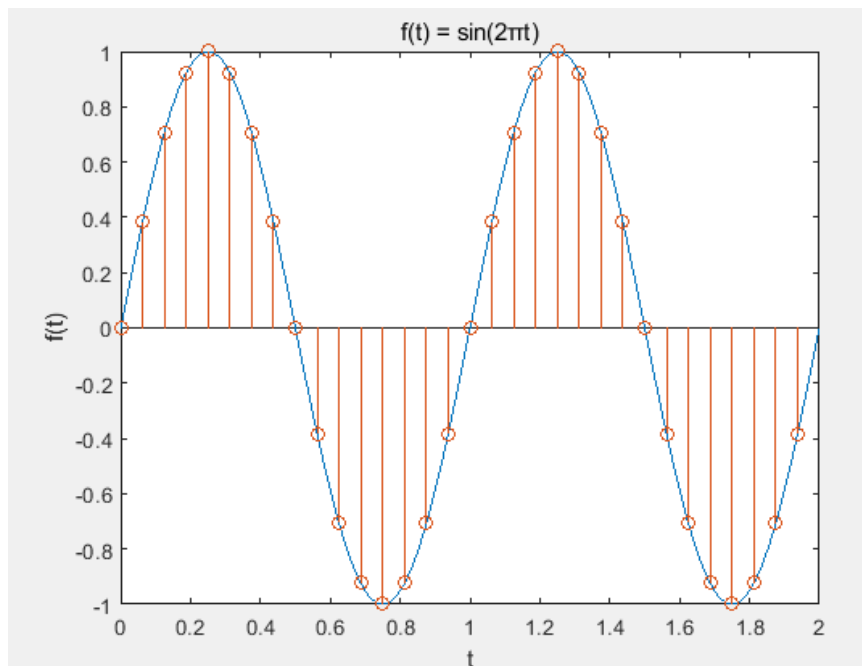
由 M2.6, 可先确定初始函数为  $y = \sin(2 \cdot \pi \cdot t)$ , 画出其图像及其采样图像, 注意根据题目的要求不应设置采样的频率过高, 这里我们设置采样点数为 16, 代码如下:

```

M2_7.m
1 - f=1;Um=1;nt=2;%输入信号频率、振幅和显示周期数
2 - N = 16; %采样点数
3 - T = 1/f;% 信号周期
4 - dt = T/N; %采样时间间隔
5 - n = 0 : nt*N - 1; %建立离散信号的时间序列
6 - tn = n*dt;
7 - y1 = Um*sin(2*f*pi*t); %声明sin函数
8 - y_1 = Um*sin(2*f*pi*tn); %对原模拟信号进行采样, 采样点数为8
9 - t = 0:0.01:2;
10 - plot(t,y1); %绘制函数
11 - hold; %锁定绘图

```

实现效果如下：



然后添加两个函数：

$$y_2 = \sin( (2\pi t + 0) + 2\pi t ) = \sin(4\pi t);$$

$$y_3 = \sin( (2\pi t + \pi/2) + 0 ) = \sin(2\pi t + \pi/2);$$

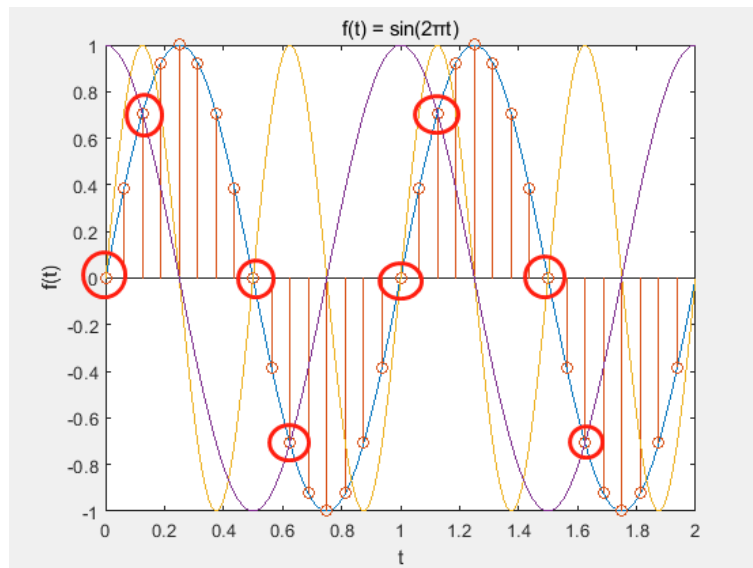
绘制；两个函数的图像：

```

20 - y2 = Um*sin(4*f*pi*t); %声明y2函数
21 - plot(t,y2);
22
23 - y3 = Um*sin(2*f*pi*t + pi/2); %声明y3函数
24 - plot(t,y3);

```

实现效果如下：



上图圈出来的部分即为三个函数采样点重合的地方，证明以上式子成立，当采样点的个数过少时，不同的函数图像可能获得同样的采样信号。

完整代码如下：

```
M2_7.m  x  +
1 - f=1;Um=1;nt=2;%输入信号频率、振幅和显示周期数
2 - N = 16;%采样点数
3 - T = 1/f;% 信号周期
4 - dt = T/N;%采样时间间隔
5 - n = 0 : nt*N - 1;%建立离散信号的时间序列
6 - tn = n*dt;
7 - y1 = Um*sin(2*f*pi*t);%声明sin函数
8 - y_1 = Um*sin(2*f*pi*tn);%对原模拟信号进行采样,采样点数为8
9 - t = 0:0.01:2;
10 - plot(t,y1);%绘制函数
11 - hold;%锁定绘图
12
13 - axis([0,nt*T,min(y1),max(y1)]);%限定横坐标和纵坐标的显示范围
14 - title('f(t) = sin(2πt)');%加标题
15 - xlabel('t');%横坐标符号
16 - ylabel('f(t)');%纵坐标符号
17 - stem(tn,y_1);
18
19
20 - y2 = Um*sin(4*f*pi*t);%声明y2函数
21 - plot(t,y2);
22
23 - y3 = Um*sin(2*f*pi*t + pi/2);%声明y3函数
24 - plot(t,y3);
```