1 Theoretical aspects

- Discretizing continuous time systems; discretizing methods;
- The response of discrete time systems;

2 Aims

Bad sampling: Aliasing and folding effects.
Discretizing in Matlab continuous time transfer functions.
The equivalent discrete transfer function for specific connections (*H(z)*).
Simulate the response of discrete time systems: the impulse response, the step response, the response to sine input signal.

3 Sampling continuous time signals

3.1 *Establishing the sampling period*

Sampling theorem (Shannon's Theorem) : $f_s > 2f_{\max}$

(f_s: sampling frequency, $f_{\max}$ The maximum frequency in the signal spectrum)

When Shannon is not respected will appear false frequencies inducing *aliasing and folding effects*

In the next graphic can be seen the aliasing effect:

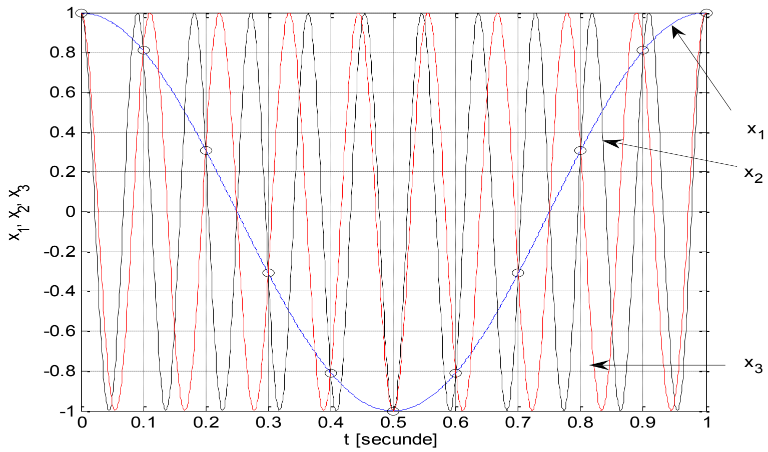
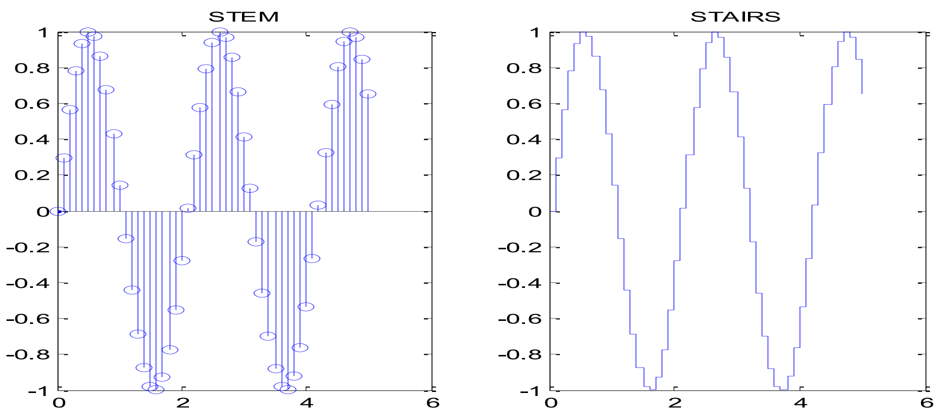


Figure 1 Aliasing and Folding effect

3.2 *Plotting discrete time signals oin Matlab*

To graphically represent sampled signals the suggestion is to use the **`stem`** function in Matlab.
To put in evidence the sampled and reconstructed signal, use the Matlab function **`stairs;`**



3.3 *Problem*

Read from Figure 1 the period of the three signal plotted and make Matlab script in order to obtain the same representation in Figure 1.

4 Discretizing the transfer function in Matlab

There are two functions available to discretize transfer functions in Matlab:

a) **c2dm** (continuous to discrete method, the old version)

The function syntax is as follows:

```
[nz,dz]=c2dm(nc,dc,Te,'metoda')
```

- **nz** and **dz** (output parameters) the nominator and denominator of the discrete transfer function;
- **nc** and **dc** are the nominator and denominator of the continuous transfer function;
- **Te** is the sampling period
- **'metoda'** represents the discretizing method with the following values
  - **'zoh'** Convert to discrete time assuming a zero order hold on the inputs.
  - **'foh'** Convert to discrete time assuming a first order hold on the inputs.
  - **'tustin'** Convert to discrete time using the bilinear (Tustin) approximation to the derivative.
  - **'prewarp'** Convert to discrete time using the bilinear (Tustin) approximation with frequency prewarping. Specify the critical frequency with an additional argument, i.e c2dm(A,B,C,D,Ts,'prewarp',Wc)
  - **'matched'** Convert the SISO system to discrete time using the matched pole-zero method.

b) **c2d** (continuous to discrete)

Este varianta modificată a funcţiei **c2dm**, pentru a putea fi apelată cu obiect de tip funcţie de transfer.
Sintaxa funcţiei este următoarea:

```
sist_discret=c2d(sist_continuu,Te,'metoda')
```

- in addition to the methods presented at the previous function, in this case is available also
  - **'imp'** (based on impulse response invariance)

4.1 *Problem*

Fill in the next table as in the example of the cell [1,4]:

The Matlab script to obtain the discrete time model::

```
[nz,dz]=c2dm(2,[1 5 6],0.01,'zoh')
nz=
1.0e-004 *
        0    0.9835    0.9672
dz =
   1.0000   -1.9506    0.9512
roots(nz)
ans =
  -0.9835
roots(dz)
ans =
   0.9802
   0.9704
```

Discretizing method	$H(s)=\frac{3}{5s+2}$		$H(s)=\frac{2}{(s+2)(s+3)}$		$H(s)=\frac{2}{s^2+0.5s+2}$	
	Te=1	Te=0.1	Te=0.6	Te=0.01	Te=1	Te=0.25
'zoh'				$H(z)=\frac{10^{-4}(0.9835z+0.9672)}{z^2-1.9506z+0.9512}$ $\dot{z}_1=-0.9835$ $\hat{z}_1=0.9802, \hat{z}_2=0.9704$		
'foh'						
'tustin'						
'matched'						

4.2 *Questions*

- Establish in what domain are situated the zeros and the poles of the discrete time transfer functions (in the table) ;
- Indicate if the order of the system changes after the discretizing process
- What happens with the values of the poles as the sampling period increases? (analyze this aspect for each discretizing method, and indicate where the values of the poles are located)
- Find the discretizing methods that introduces zeros in the discrete transfer function and when?

5 Discrete time system response in Matlab

To obtain the graphical representation of the response in Matlab, it is preferable to use the c2d function to obtain the discrete time transfer function.
Example of script in Matlab

```
wn=2;nc=wn^2;dc=[1 wn wn^2]; sist_c=tf(nc,dc);te=pi/(4*wn);
sist_d=c2d(sist_c,te,'tustin');
subplot(221);impulse(sist_d);
subplot(222);step(sist_d);
subplot(223);t=0:te:10/wn;lsim(sist_d,t,t); title('Ramp response')
subplot(224);t=0:te:60;lsim(sist_d,sin(t),t); title('Response to sin(t)')
```