

III – Amostragem

Processamento Digital de Sinais

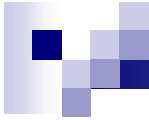




Sumário

- Amostragem
- Teorema da Amostragem
- Aliasing
- Interpolação





PROCESSAMENTO DIGITAL DE SINAL





Objectivo deste capítulo

Entender o teorema da amostragem:

Quando um sinal é amostrado a uma taxa superior a duas vezes a frequência mais alta do espectro do sinal, então é possível reconstruir exatamente o sinal original a partir das amostras.

Ex: CDs ónde a música é guardada no forma digital e o leitor de CDs reconstroi o sinal que se ouve (sinal contínuo) na forma analógica.

O processo de reconstrução é basicamente **interpolação** (ligar as bolinhas do stem por uma linha de plot)





Os computadores não lidam com sinais contínuos diretamente: manipulam-nos simbolicamente ou numericamente, mas sempre na forma discreta.

Um sinal discreto é uma sucessão de índice n que só por si não tem informação sobre o tempo.

Estes índices estão relacionados com instantes de tempo através do período de amostragem T_s do sinal:

$$x[n] = x(nT_s), \quad n \in \mathbb{Z}$$

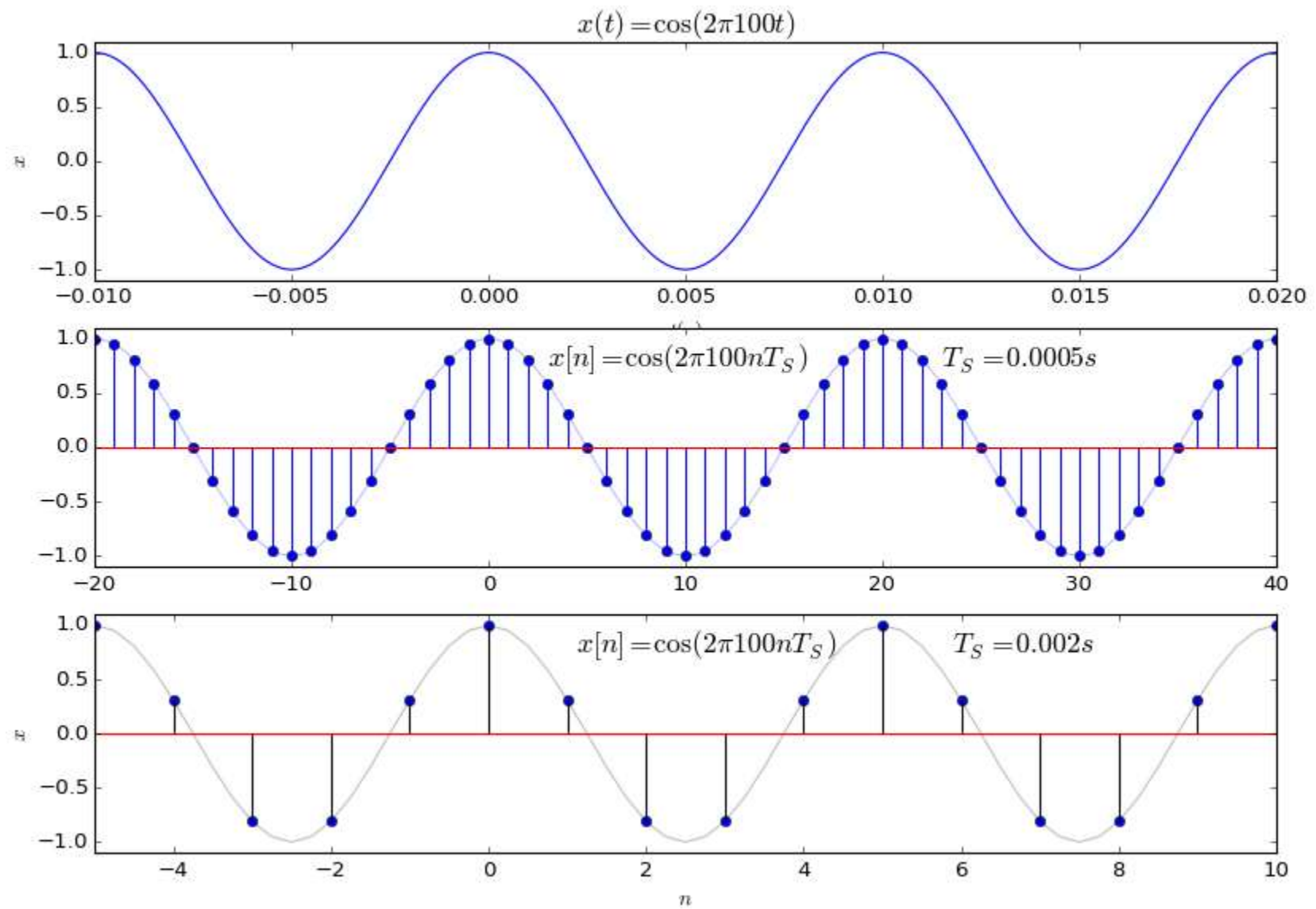
Seja o sinal contínuo:

$$x(t) = A \cos(2\pi ft + \phi)$$

Se amostramos este sinal à taxa $F_s = \frac{1}{T_s}$ obtém-se:

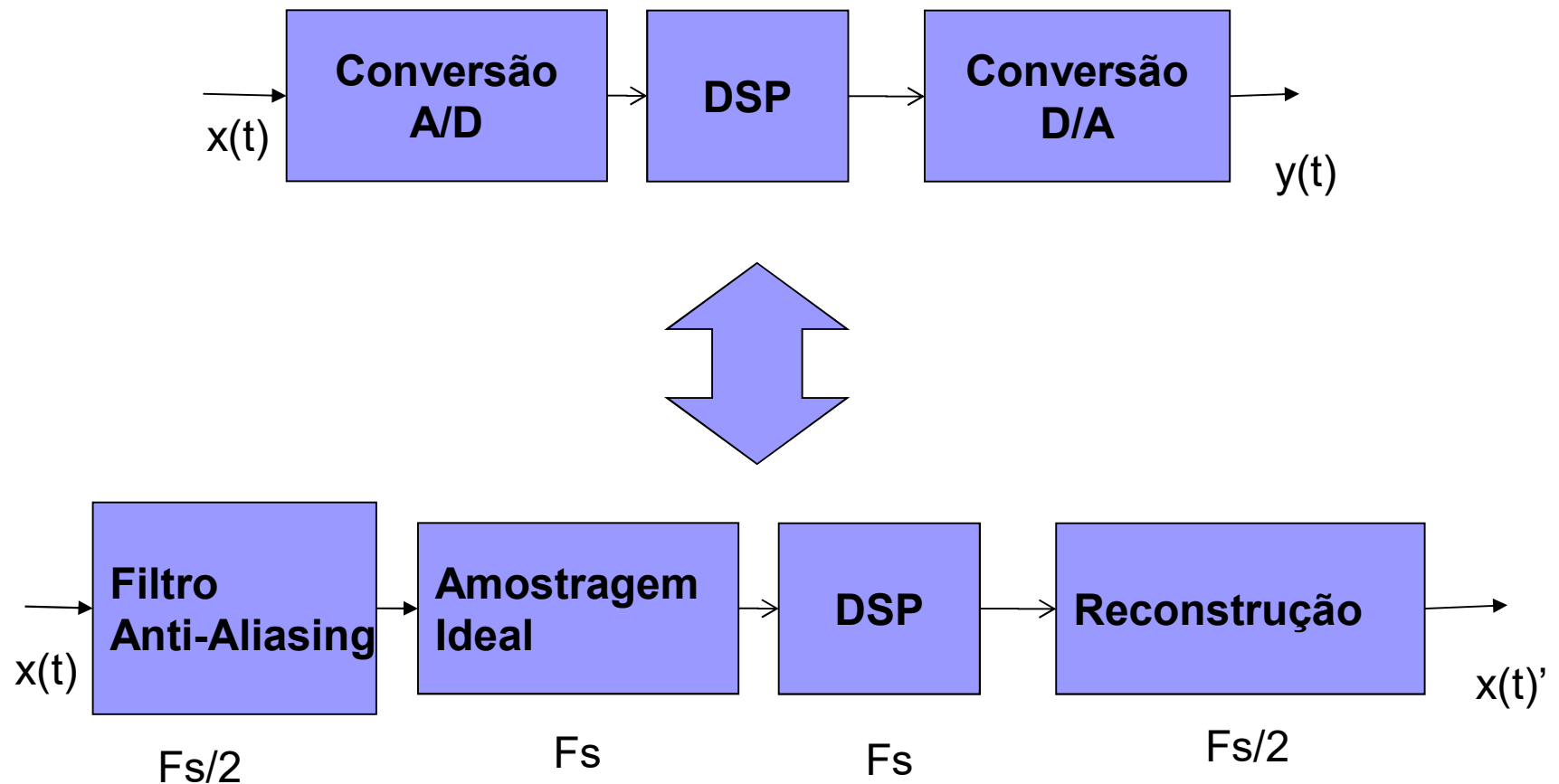
$$x[n] = x(nT_s) = A \cos(2\pi fnT_s + \phi)$$







Processamento Digital de Sinal





Conversão Analógico/Digital (A/D)

Os dispositivos digitais armazenam a informação usando bits.

Como se transformam sinais em números?

Converter um sinal contínuo no tempo e amplitude para um conjunto de números representados por conjunto limitado de bits.

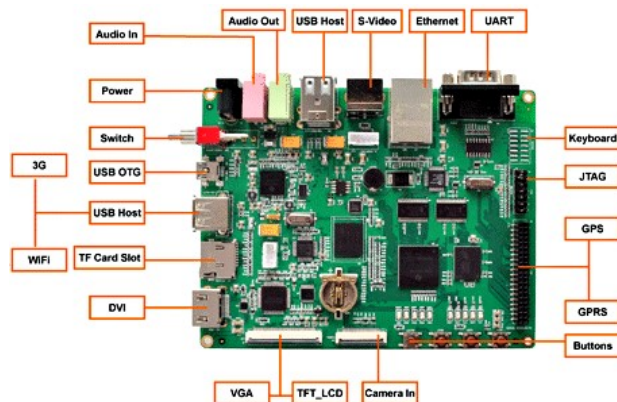
Resposta: **Digitalização**

Amostragem (contínuo \rightarrow discreto no tempo);
Quantificação (contínuo \rightarrow discreto na amplitude).



DSP – Digital Signal Processing

- Processamento/transformação das amostras no domínio digital



 beagleboard.org





Conversão Digital/Analógico (D/A)

- Realiza-se convertendo as amostras na forma digital em valores analógicos, a partir de um processo denominado de **reconstrução ou interpolação**

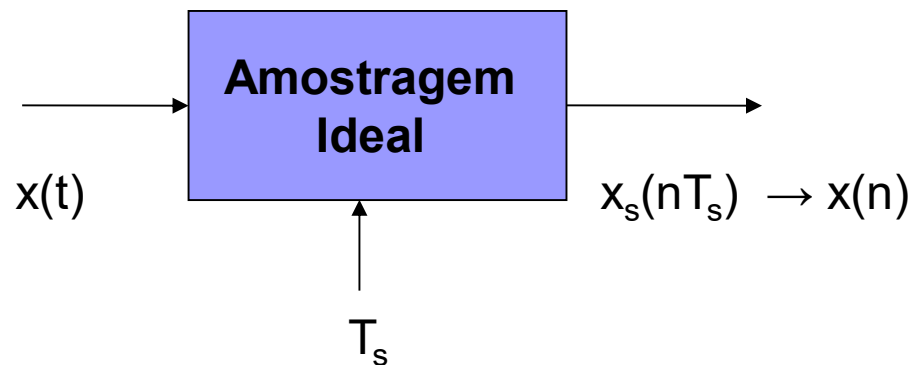




AMOSTRAGEM



Amostragem



- Conversão de um sinal contínuo num sinal discreto

- Período de Amostragem: T_s
- Frequência de Amostragem: F_s

$$T_s = 1/F_s$$



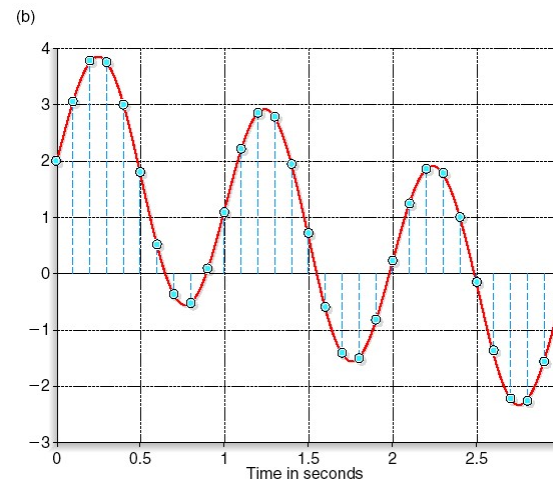
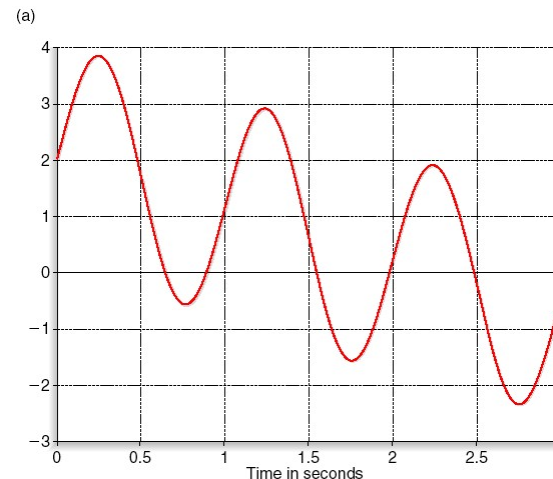
Amostragem

- **Amostragem:** retirar valores do sinal em intervalos de tempo equi-espçados
- **Formula:**

$$s[n] = s(nT_s)$$

$$s(t) = \{\text{sinal original}\}$$

$$s[n] = \{\text{sinal amostrado}\}$$
- $T_s = \{\text{período de amostragem}\}$
- $f_s = 1/T_s$ define a frequência ou ritmo de amostragem.



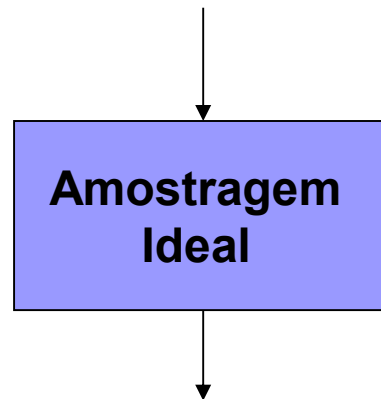
Time (s)	Sample Values
0.0	2.0000
0.1	3.1674
0.2	3.8694
0.3	3.8289
0.4	3.0465
0.5	1.8006
0.6	0.5412
0.7	-0.2814
0.8	-0.3885
0.9	0.2217
1.0	1.2732
1.1	2.3188
1.2	2.9112
1.3	2.7748
1.4	1.9113
1.5	0.6002
1.6	-0.7078
1.7	-1.5621
1.8	-1.6835
1.9	-1.0707
2.0	0.0000
2.1	1.0807
2.2	1.7233
2.3	1.6508
2.4	0.8637
2.5	-0.3601
2.6	-1.5718
2.7	-2.3223
2.8	-2.3346
2.9	-1.6092
3.0	-0.4244





Amostragem de uma Sinusóide

$$x(t) = A \cos(\omega t + \phi)$$



$$x[n] = x(nT_s) = A \cos(\omega nT_s + \phi) = A \cos(\hat{\omega}n + \phi)$$

■ Frequência Normalizada

$$\hat{\omega} = \omega T_s = \frac{\omega}{F_s}$$





Frequência Normalizada

$$\hat{\omega} = \omega T_s = \frac{\omega}{F_s}$$

Expressa em radianos

$$0 < \hat{\omega} < 2\pi$$

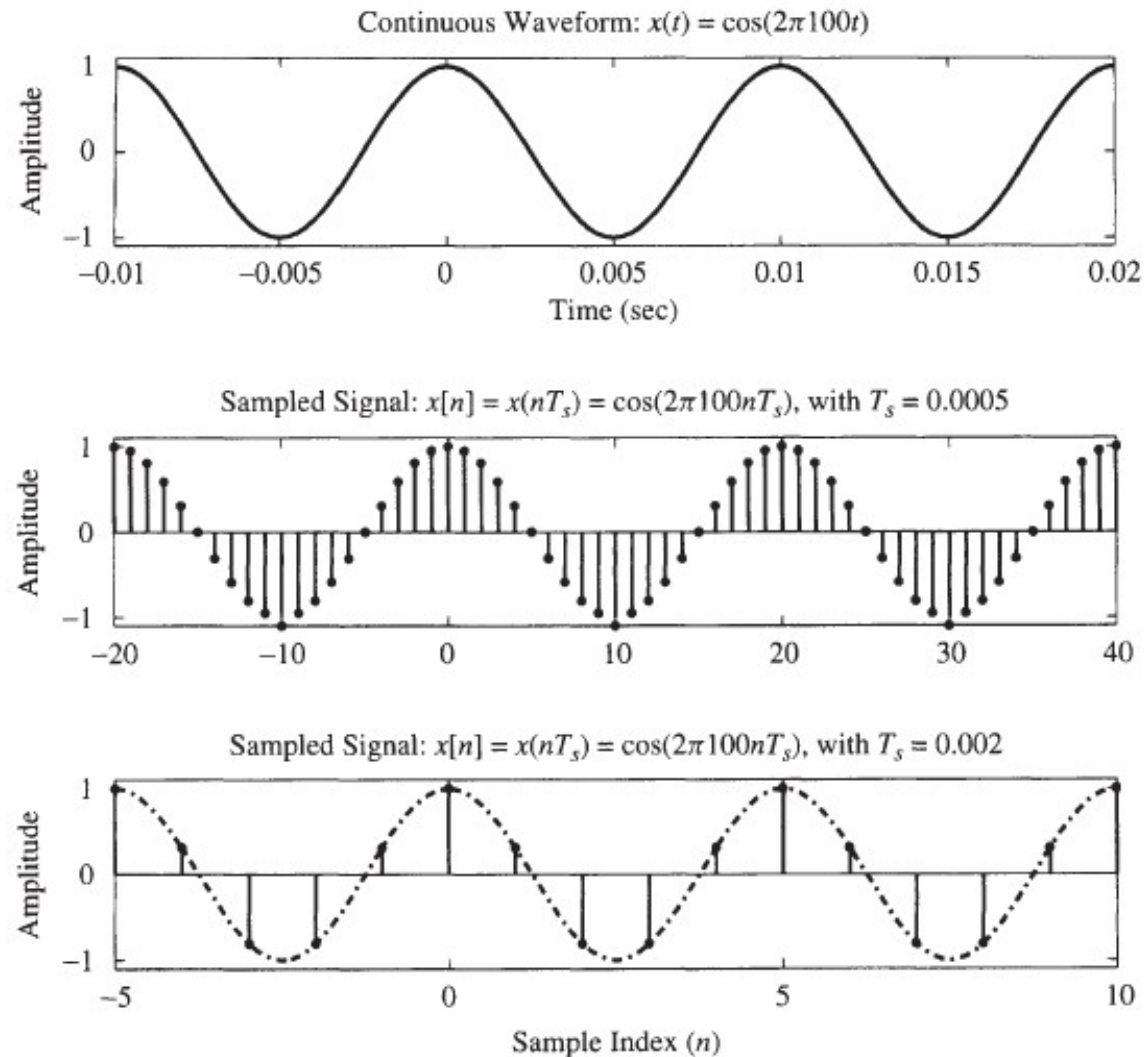
$$\hat{f} = f T_s = \frac{f}{F_s}$$

Adimensional

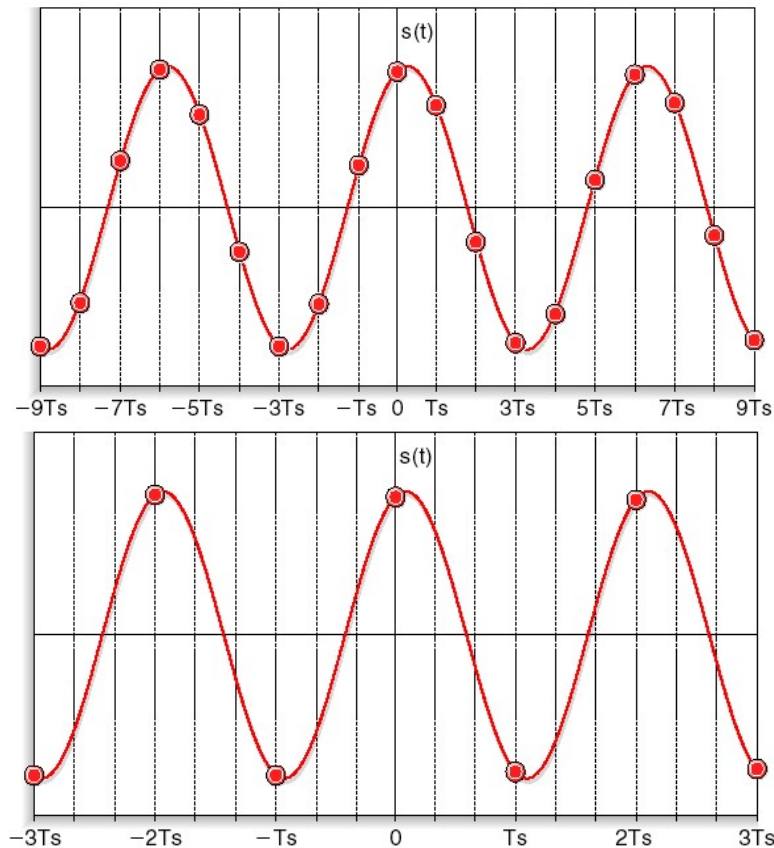
$$0 < \hat{f} < 1$$



Amostragem de uma Sinusóide



Amostragem - Compromisso



- Um valor pequeno de T_s gera demasiadas amostras
- Um valor grande de T_s destrói o sinal original
- Como se *calcula* T_s ?
- Resposta: O Ritmo de *Nyquist*.



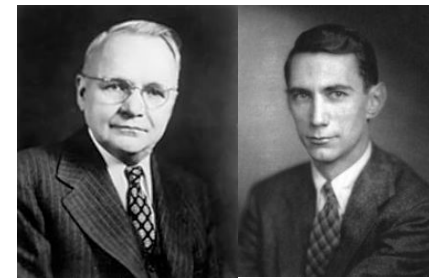
Teorema da Amostragem

Seja $x(t)$ um sinal contínuo com frequências f não superiores a f_{\max} . Então esse sinal poderá ser reconstruído a partir das suas amostras $x[n] = x(nT_s)$, se estas forem retiradas com um ritmo $F_s = 1/T_s$ maior que $2 \times f_{\max}$

■ Frequência de Nyquist:

$$F_{\text{nyquist}} = 2 \times f_{\max}$$

$$F_s > F_{\text{nyquist}}$$

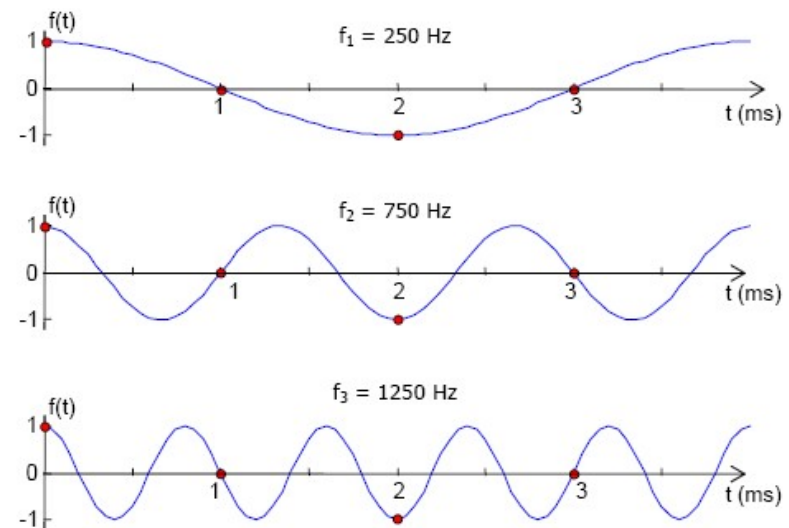


Nyquist-Shannon
(1889-1976)/(1916-2001)



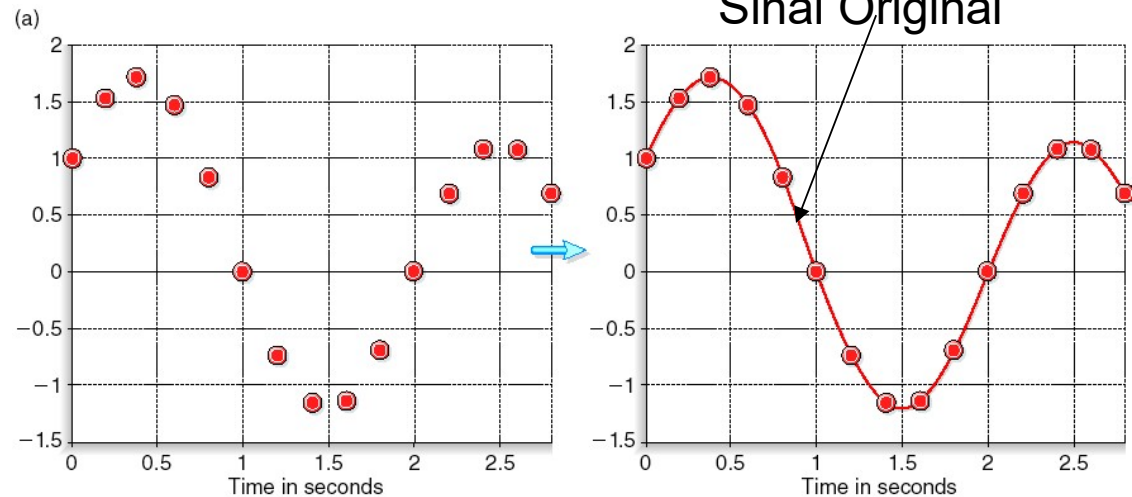
Amostragem

- A mesma sequência de amostras foi obtida em três sinusóides diferentes.
- Poderá servir para recuperar essas três sinusóides?
- Claro que não!!

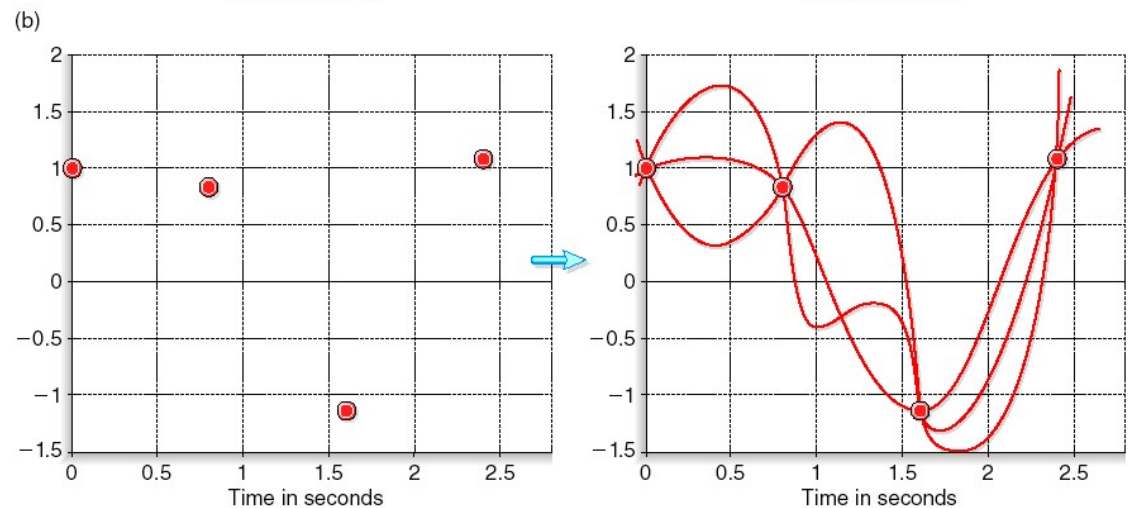


Amostragem Superior ou Inferior ao Ritmo de *Nyquist*

- $f_s > 2 f_{\max}$



- $f_s < 2 f_{\max}$



Ritmos de Amostragem para alguns Sinais Importantes

Signal Type	Signal Bandwidth	Minimum Sampling Rate	Rate Actually Used
Telephone quality speech	3.5 kHz	7000 samples/s	8000 samples/s
Music	20 kHz	40,000 samples/s	44,100 samples/s
FM radio	200 kHz	400,000 samples/s	400,000 samples/s
Standard definition television	6 MHz	12,000,000 samples/s	14,400,000 samples/s

- Os projectistas utilizam estes ritmos de amostragem para projectar leitores de CD ou DVD, telemóveis, rádios para automóveis e satélites de TV.



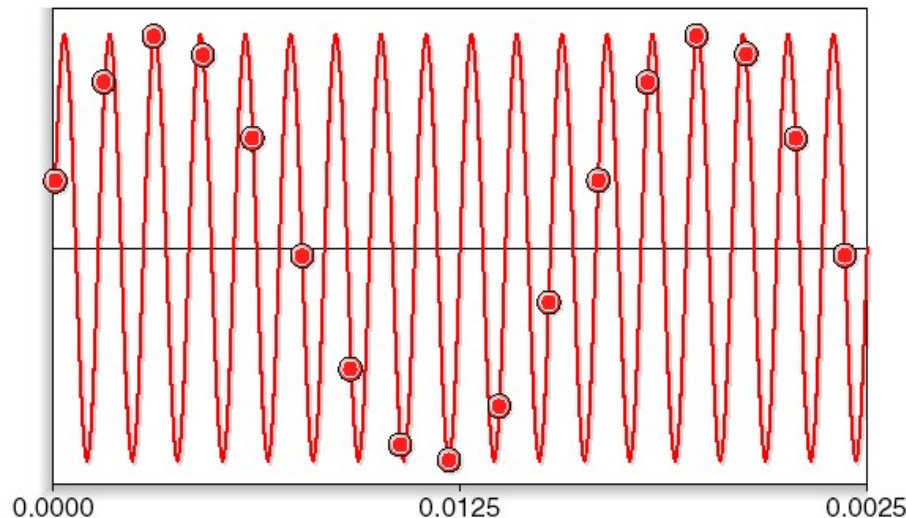


ALIASING



Amostragem Inferior ao Ritmo de Nyquist Produz *Aliasing*

- Exemplo: Uma sinusóide com 720Hz amostrada a um ritmo de 660Hz...



...gera uma sinusóide que parece ter 60Hz!

- Como explicar este fenómeno?



Aliasing

- As sinusoides $x(t)$ e $y(t)$ produzem a mesma sequência de amostras quando amostradas à frequência F_s

$$x(t) = A \cos(2\pi f_0 t + \phi)$$
$$y(t) = A \cos(2\pi (f_0 + m f_s) t + \phi)$$

$$x[n] = x(nT_s) = A \cos(2\pi f_0 nT_s + \phi)$$

$$\begin{aligned} y[n] &= y(nT_s) = A \cos(2\pi (f_0 + m f_s) nT_s + \phi) = \\ &= A \cos(2\pi f_0 nT_s + 2\pi n \cancel{T_s} m \cancel{f_s} + \phi) = \\ &= A \cos(2\pi f_0 nT_s + \phi) = x[n] \end{aligned}$$





Folding

- Outra fonte de aliasing advém das frequências negativas do espectro

$$w(t) = A \cos(2\pi(-f_0 + mf_s)t - \phi)$$

$$\begin{aligned} w[n] = w(nT_s) &= A \cos(2\pi(-f_0 + mf_s)nT_s - \phi) = \\ &= A \cos(2\pi f_0 nT_s + \phi) = x[n] \end{aligned}$$



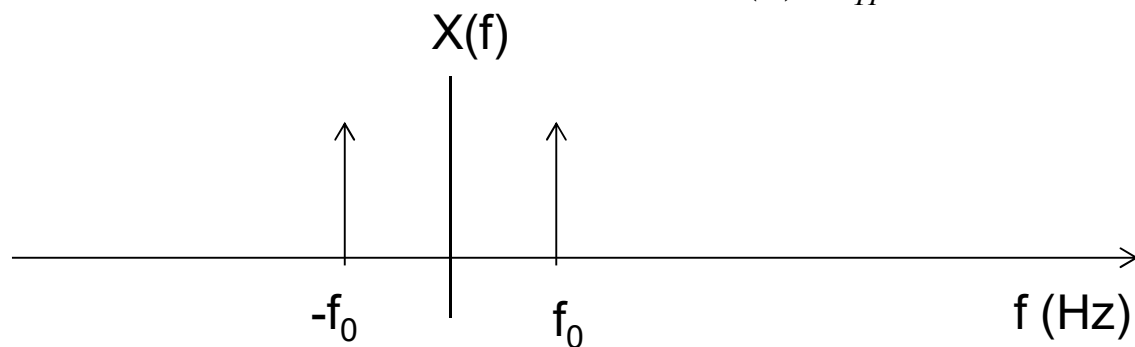
Interpretação Espectral



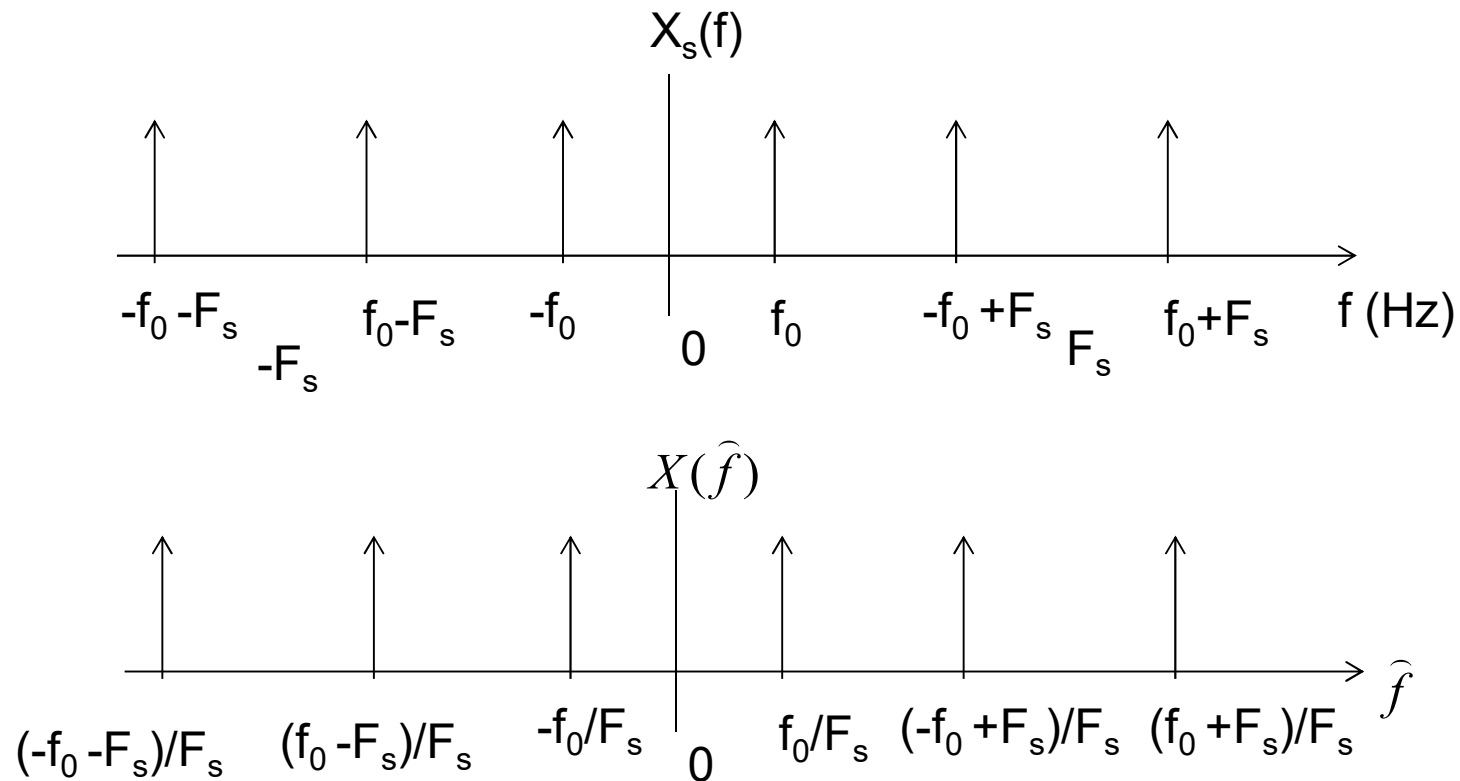
$$x(t) \xleftrightarrow{TF} X(f)$$

$$x(n) \xleftrightarrow{TF} X(\hat{\omega})$$

$$x(n) \xleftrightarrow{TF} X(\hat{f})$$

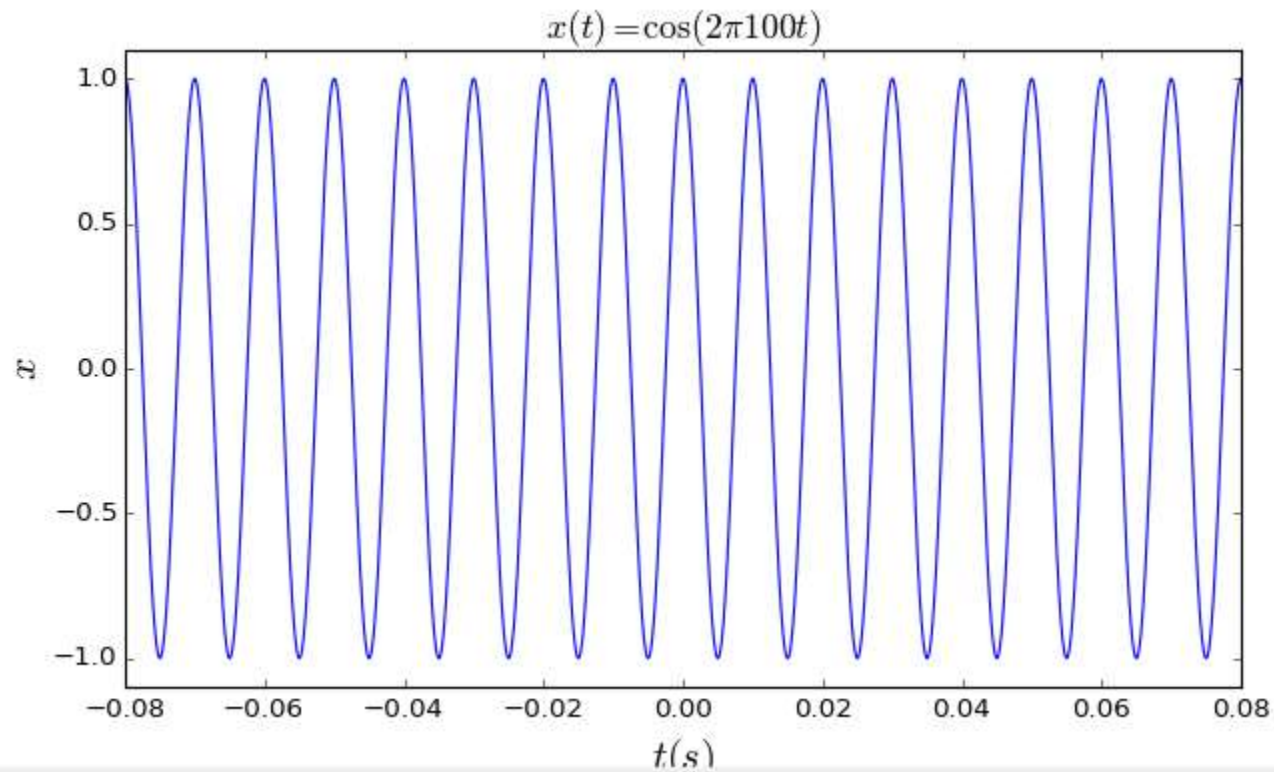


Interpretação Espectral





Sinal analógico original

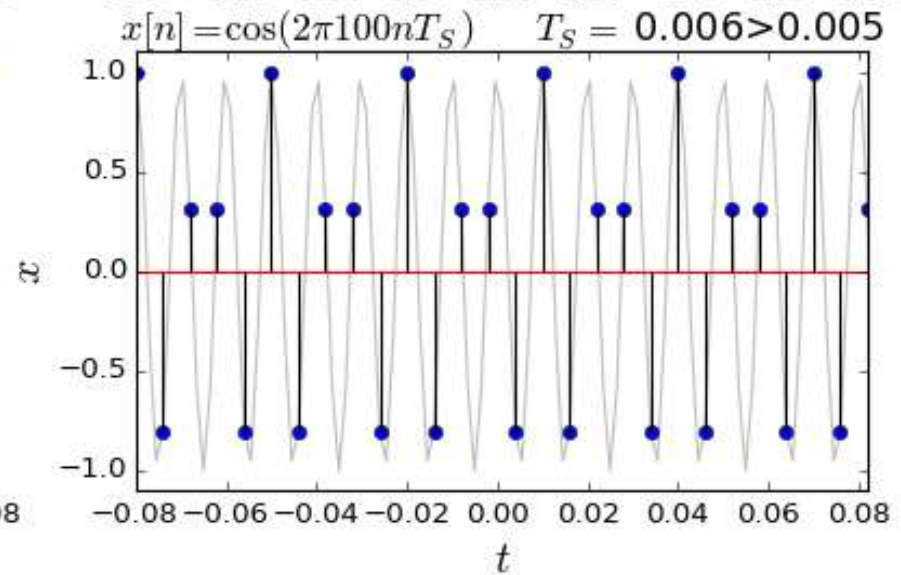
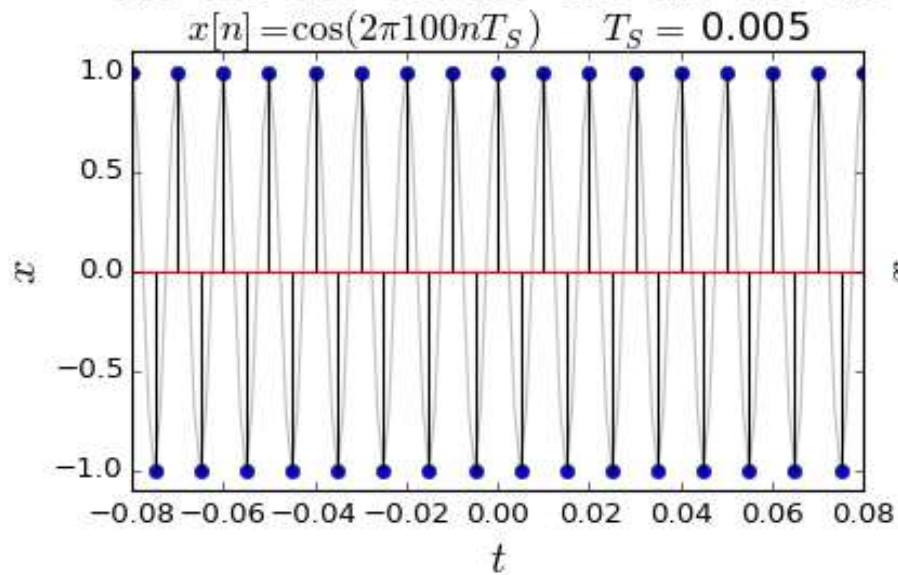
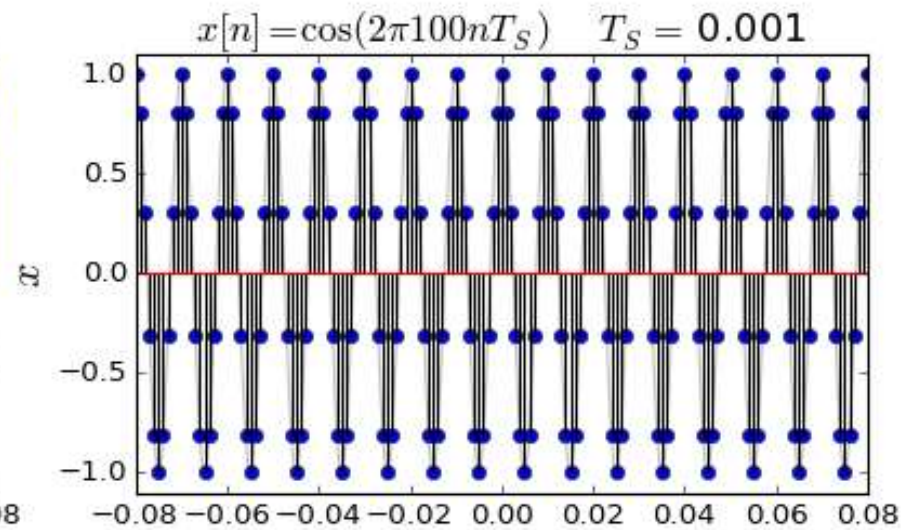
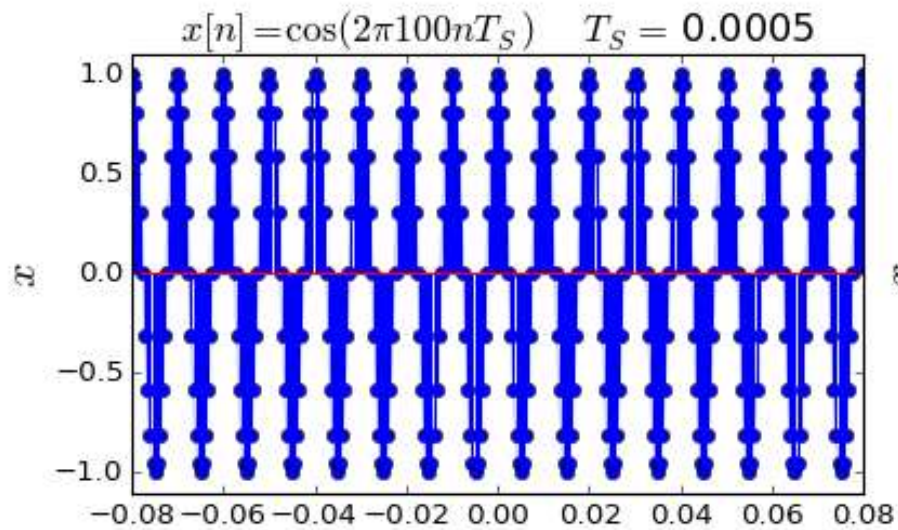


teorema amostragem

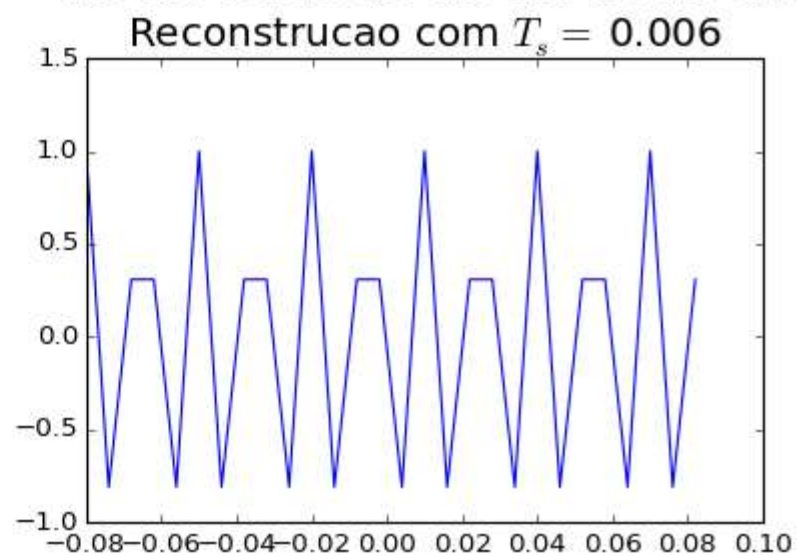
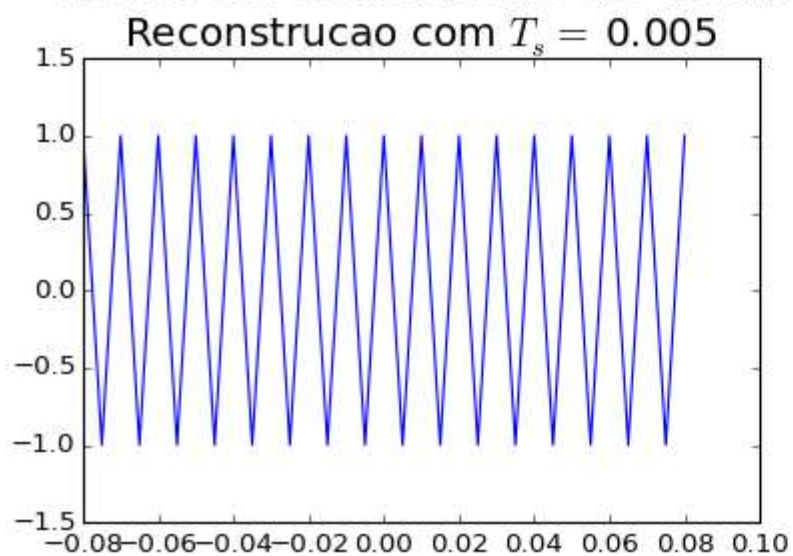
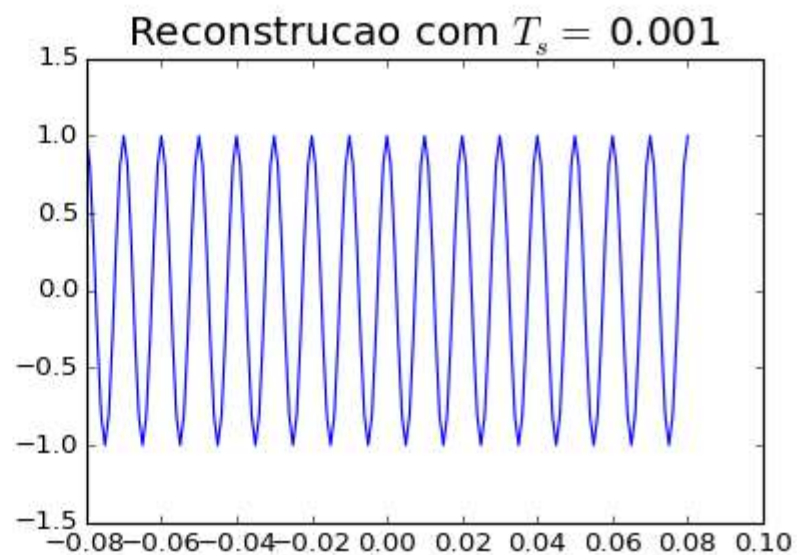
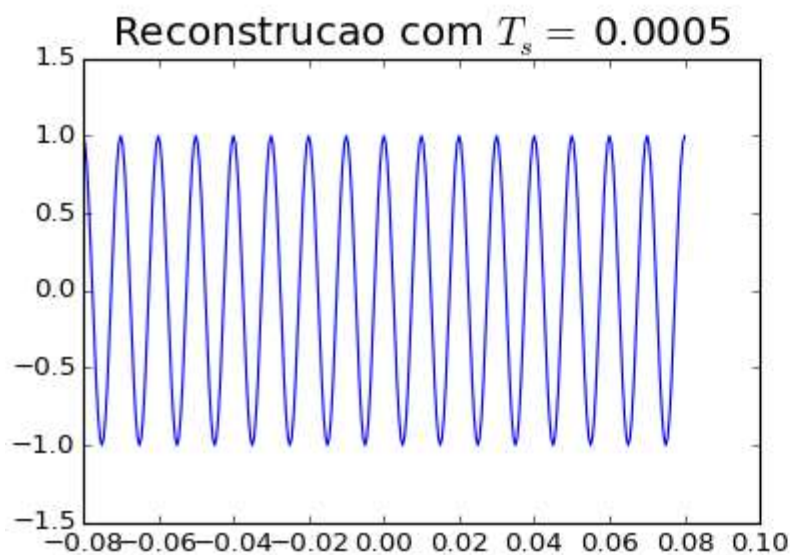
$$T = 0.01s \quad \xrightarrow{\quad} \quad f_s > 2f \Rightarrow f_s > 200 \quad \Rightarrow \quad T_s < 0.005s$$

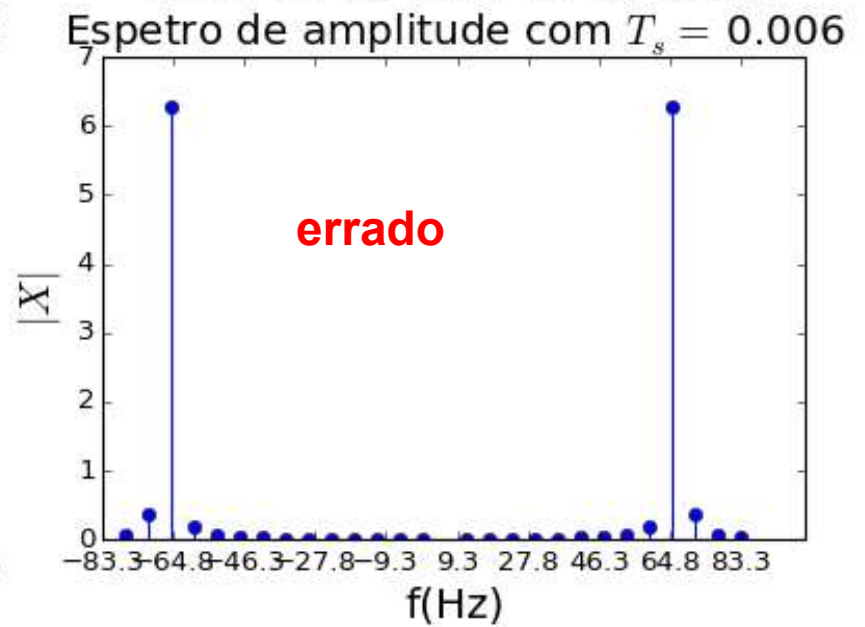
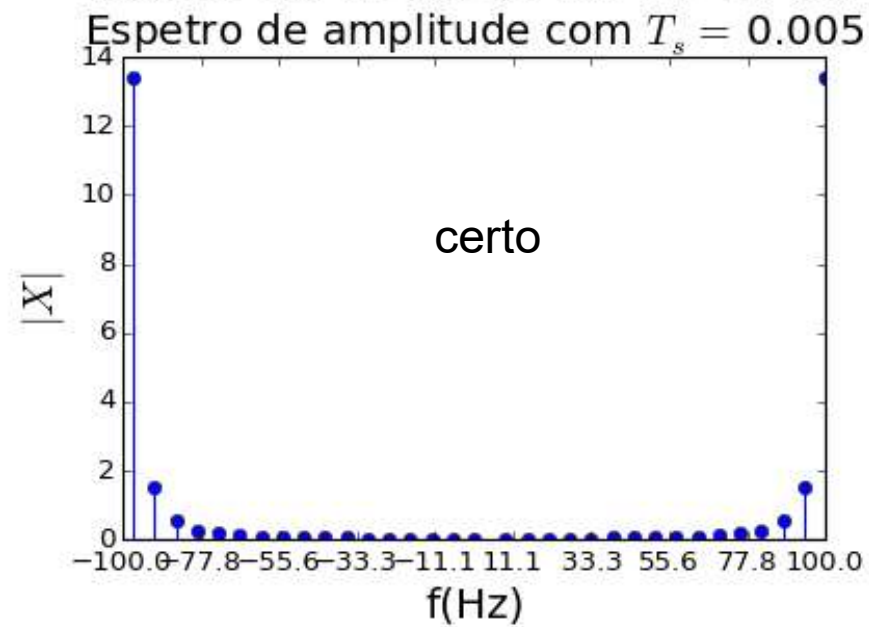
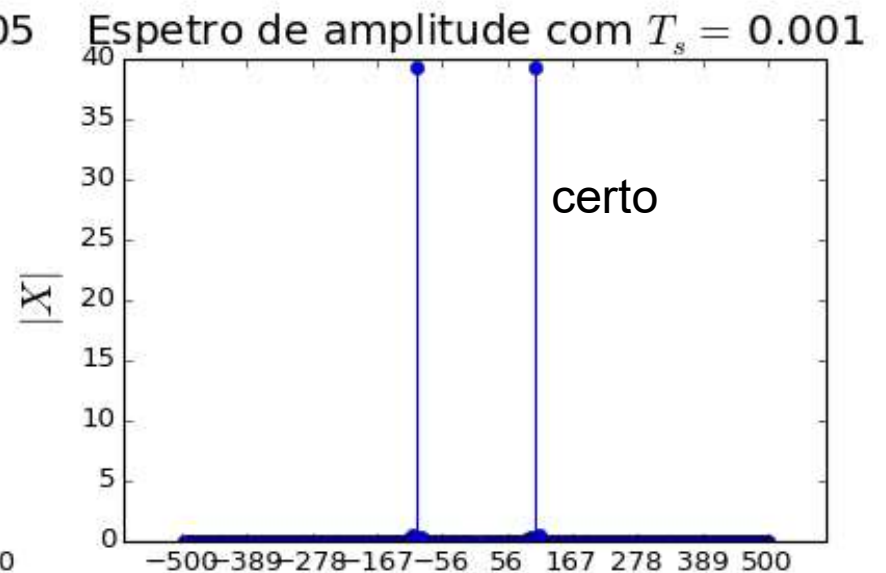
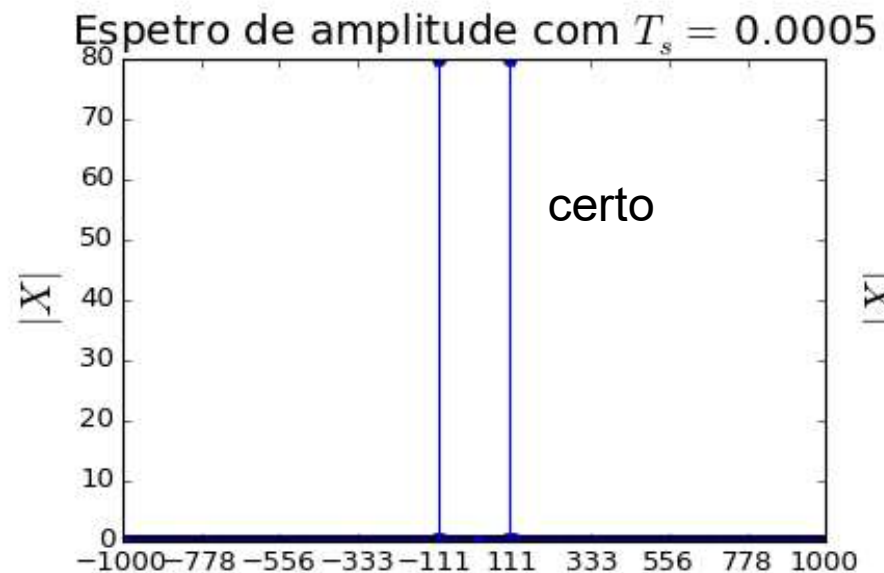


Amostragem



FFT + IFFT → Reconstrução







```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Mar 16 18:59:56 2016
4
5 @author: Isabel Rodrigues
6 """
7
8 # -*- coding: latin-1 -*-
9 import matplotlib.pyplot as plt
10 import numpy as np
11
12 Ts1=0.0005
13 Ts2=0.001
14 Ts3=0.005
15 Ts4=0.006
16
17 tI=[-0.08,0.08]
18 t=np.linspace(tI[0],tI[1],1e4)
19
20 n1=np.arange(tI[0]/Ts1,tI[1]/Ts1+1,1)
21 n2=np.arange(tI[0]/Ts2,tI[1]/Ts2+1,1)
22 n3=np.arange(tI[0]/Ts3,tI[1]/Ts3+1,1)
23 n4=np.arange(tI[0]/Ts4,tI[1]/Ts4+1,1)
24
25 nn1=np.arange(tI[0]/Ts1,tI[1]/Ts1+1,0.
26 nn2=np.arange(tI[0]/Ts2,tI[1]/Ts2+1,0.
27 nn3=np.arange(tI[0]/Ts3,tI[1]/Ts3+1,0.
28 nn4=np.arange(tI[0]/Ts4,tI[1]/Ts4+1,0.
29
30
31 x_t=np.cos(2*np.pi*100*t)
32
33 x_n1=np.cos(2.*np.pi*100*n1*Ts1)
34 x_n2=np.cos(2.*np.pi*100*n2*Ts2)
35 x_n3=np.cos(2.*np.pi*100*n3*Ts3)
36 x_n4=np.cos(2.*np.pi*100*n4*Ts4)
37
38 x_nn1=np.cos(2.*np.pi*100*nn1*Ts1)
39 x_nn2=np.cos(2.*np.pi*100*nn2*Ts2)
40 x_nn3=np.cos(2.*np.pi*100*nn3*Ts3)
41 x_nn4=np.cos(2.*np.pi*100*nn4*Ts4)
42
43 #plot
44 plt.close('all')
45 plt.figure(facecolor='w',figsize=(9,5))
46 plt.title(r'$x(t)=\cos(2 \pi 100t)$', fontsize=16)
47 plt.plot(t,x_t)
48 plt.xlabel('$t(s)$', fontsize=18);
49 plt.ylabel('$x$', fontsize=18);
50 plt.axis([t.min(),t.max(),-1.1,1.1])
51
52
53 plt.figure(facecolor='w',figsize=(12,7))
54
55 plt.subplot(221)
56 plt.title(r'$x[n]=\cos(2 \pi 100nT_S)$ $T_S=$ '+np.str(Ts1), fontsize=16)
57 plt.plot(nn1*Ts1,x_nn1,'b',alpha=0.25)
58 plt.stem(n1*Ts1,x_n1,'b',linewidth=1.5)
59 plt.axis([n1.min()*Ts1,n1.max()*Ts1,-1.1,1.1])
60 plt.ylabel('$x$', fontsize=18);
61 #s1=r'$T_S=$ '+np.str(Ts1)
62 #plt.title(s1, fontsize=18)
63
64 plt.subplot(222)
65 plt.title(r'$x[n]=\cos(2 \pi 100nT_S)$ $T_S=$ '+np.str(Ts2), fontsize=16)
66 plt.plot(nn2*Ts2,x_nn2,'k',alpha=0.25)
67 plt.stem(n2*Ts2,x_n2,'k',linewidth=1.5)
68 plt.axis([n2.min()*Ts2,n2.max()*Ts2,-1.1,1.1])
69 plt.ylabel('$x$', fontsize=18);
70 #s2=r'$T_S=$ '+np.str(Ts2)
71 #plt.title(s2)
72
```



```

73 plt.subplot(223)
74 plt.title(r'$x[n]=\cos(2 \pi 100nT_S)$ $T_S=$ '+np.str(Ts3), fontsize=16)
75 plt.plot(nn3*Ts3,x_nn3,'k',alpha=0.25)
76 plt.stem(n3*Ts3,x_n3,'k',linewidth=1.5)
77 plt.axis([n3.min()*Ts3,n3.max()*Ts3,-1.1,1.1])
78 plt.ylabel('$x$', fontsize=18);
79 plt.xlabel('$t$', fontsize=18);
80 #s3=r'$T_s=$ '+np.str(Ts3)
81 #plt.title(s3)
82
83 plt.subplot(224)
84 plt.title(r'$x[n]=\cos(2 \pi 100nT_S)$ $T_S=$ '+np.str(Ts4)+'>0.005', fontsize=16)
85 plt.plot(nn4*Ts4,x_nn4,'k',alpha=0.25)
86 plt.stem(n4*Ts4,x_n4,'k',linewidth=1.5)
87 plt.axis([n4.min()*Ts4,n4.max()*Ts4,-1.1,1.1])
88 plt.xlabel('$t$', fontsize=18);
89 plt.ylabel('$x$', fontsize=18);
90
91 plt.figure(facecolor='w',figsize=(12,8))
92
93 Y1=np.fft.fft(x_n1);
94 z1=np.fft.ifft(Y1)
95 plt.subplot(221)
96 plt.plot(n1*Ts1,z1)
97 s1='Reconstrucao com '+r'$T_s=$ '+np.str(Ts1)
98 plt.title(s1, fontsize=18)
99
100 Y2=np.fft.fft(x_n2);
101 z2=np.fft.ifft(Y2)
102 plt.subplot(222)
103 plt.plot(n2*Ts2,z2)
104 s2='Reconstrucao com '+r'$T_s=$ '+np.str(Ts2)
105 plt.title(s2, fontsize=18)
106
107 Y3=np.fft.fft(x_n3);
108 z3=np.fft.ifft(Y3)

```



```

109 plt.subplot(223)
110 plt.plot(n3*Ts3,z3)
111 s3='Reconstrucao com '+r'$T_s=$ '+np.str(Ts3)
112 plt.title(s3, fontsize=18)
113
114 Y4=np.fft.fft(x_n4);
115 z4=np.fft.ifft(Y4)
116 plt.subplot(224)
117 plt.plot(n4*Ts4,z4)
118 s4='Reconstrucao com '+r'$T_s=$ '+np.str(Ts4)
119 plt.title(s4, fontsize=18)
120
121
122
123 plt.figure(facecolor='w',figsize=(12,8))
124
125 Pyy1 = Y1* np.conj(Y1)/np.size(Y1)
126 #Phase_y1=np.arctan2(np.imag(Y1),np.real(Y1))
127 #f1 = 1./Ts1*np.arange(1,np.size(Y1)/2+1)/np.size(Y1)
128 fd = 1./Ts1*np.arange(0.,-np.size(Y1)/2.,-1.)/np.size(Y1)
129 f1 =np.array(np.concatenate([fd, 1./Ts1*np.arange(np.size(Y1)/2.,1,-1.)/np.size(Y1)],axis=0))
130 plt.subplot(221)
131 plt.stem(f1,Pyy1[0:np.size(f1)])
132 plt.xticks(np.linspace(-1./(2*Ts1),1./(2*Ts1),10), rotation=0)
133 s1='Espectro de amplitude com '+r'$T_s=$ '+np.str(Ts1)
134 plt.title(s1, fontsize=18)
135 #plt.xlabel('f(Hz)', fontsize=18)
136 plt.ylabel('$|X|$', fontsize=18)
137 #intervalos no x = 1/2
138
139 Pyy2 = Y2* np.conj(Y2)/np.size(Y2)
140 #Phase_y2=np.arctan2(np.imag(Y2),np.real(Y2))
141 fd = 1./Ts2*np.arange(0.,-np.size(Y2)/2.,-1.)/np.size(Y2)
142 f2 =np.array(np.concatenate([fd, 1./Ts2*np.arange(np.size(Y2)/2.,1,-1.)/np.size(Y2)],axis=0))
143 plt.subplot(222)
144 plt.stem(f2,Pyy2[0:np.size(f2)])

```



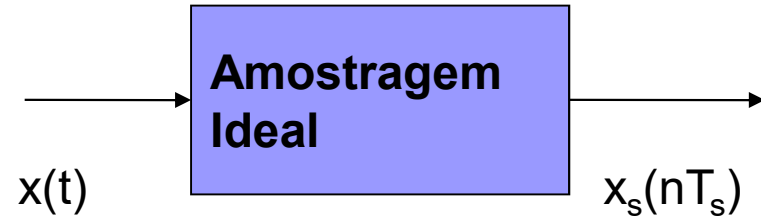
```

145 plt.xticks(np.linspace(-1./(2*Ts2),1./(2*Ts2),10), rotation=0)
146 s2='Espectro de amplitude com '+r'$T_s=$ '+np.str(Ts2)
147 plt.title(s2, fontsize=18)
148 plt.ylabel('$|X|$', fontsize=18)
149 #plt.xlabel('f(Hz)', fontsize=18)
150
151 Pyy3 = Y3* np.conj(Y3)/np.size(Y3)
152 #Phase_y3=np.arctan2(np.imag(Y3),np.real(Y3))
153 fd = 1./Ts3*np.arange(0.,-np.size(Y3)/2.,-1.)/np.size(Y3)
154 f3 =np.array(np.concatenate([fd, 1./Ts3*np.arange(np.size(Y3)/2.,1,-1.)/np.size(Y3)],axis=0))
155 plt.subplot(223)
156 plt.stem(f3,Pyy3[0:np.size(f3)])
157 plt.xticks(np.linspace(-1./(2*Ts3),1./(2*Ts3),10), rotation=0)
158 s3='Espectro de amplitude com '+r'$T_s=$ '+np.str(Ts3)
159 plt.title(s3, fontsize=18)
160 plt.ylabel('$|X|$', fontsize=18)
161 plt.xlabel('f(Hz)', fontsize=18)
162
163 Pyy4 = Y4* np.conj(Y4)/np.size(Y4)
164 #Phase_y3=np.arctan2(np.imag(Y3),np.real(Y3))
165 fd = 1./Ts4*np.arange(0.,-np.size(Y4)/2.,-1.)/np.size(Y4)
166 f4 =np.array(np.concatenate([fd, 1./Ts4*np.arange(np.size(Y4)/2.,1,-1.)/np.size(Y4)],axis=0))
167 plt.subplot(224)
168 plt.stem(f4,Pyy4[0:np.size(f4)])
169 plt.xticks(np.linspace(-1./(2*Ts4),1./(2*Ts4),10), rotation=0)
170 s4='Espectro de amplitude com '+r'$T_s=$ '+np.str(Ts4)
171 plt.title(s4, fontsize=18)
172 plt.ylabel('$|X|$', fontsize=18)
173 plt.xlabel('f(Hz)', fontsize=18)
174

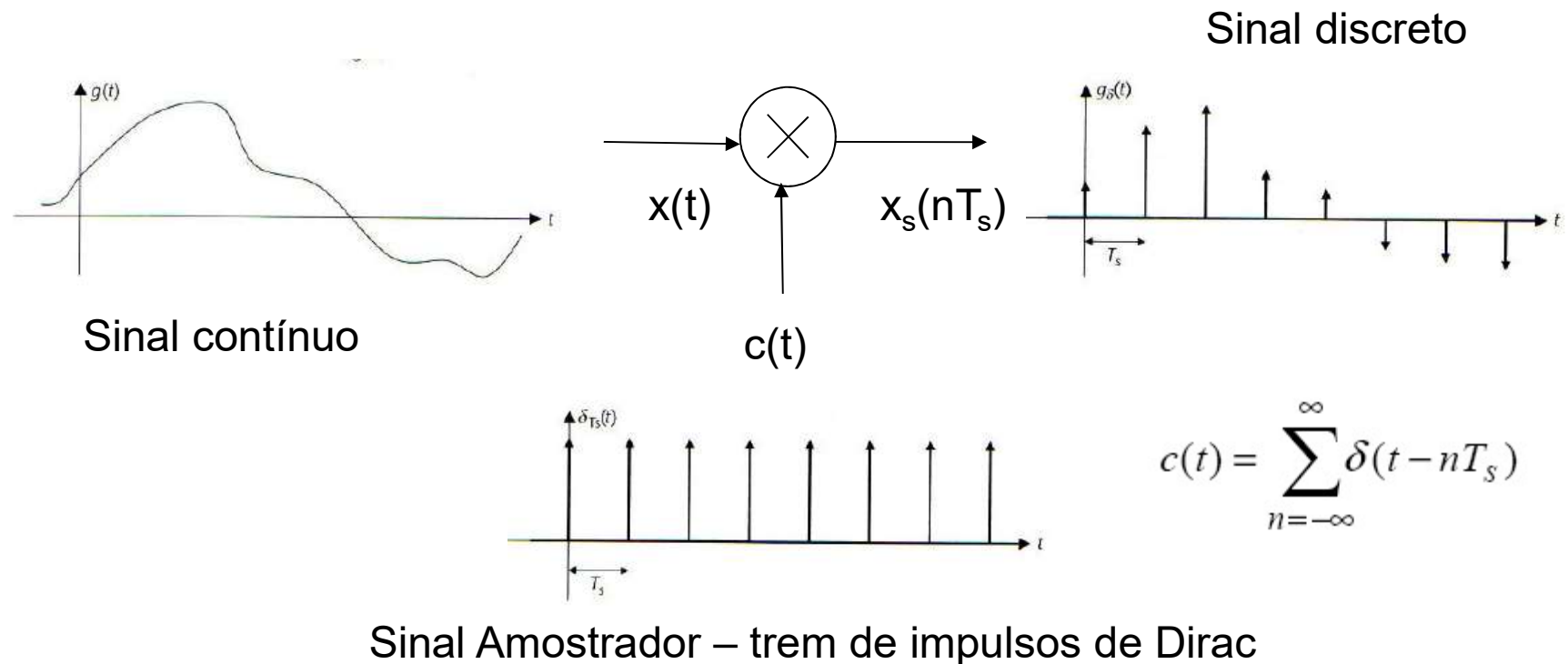
```



Amostragem Ideal



- Pode ser vista como a multiplicação por um trem de impulsos de Dirac:



Amostragem Ideal

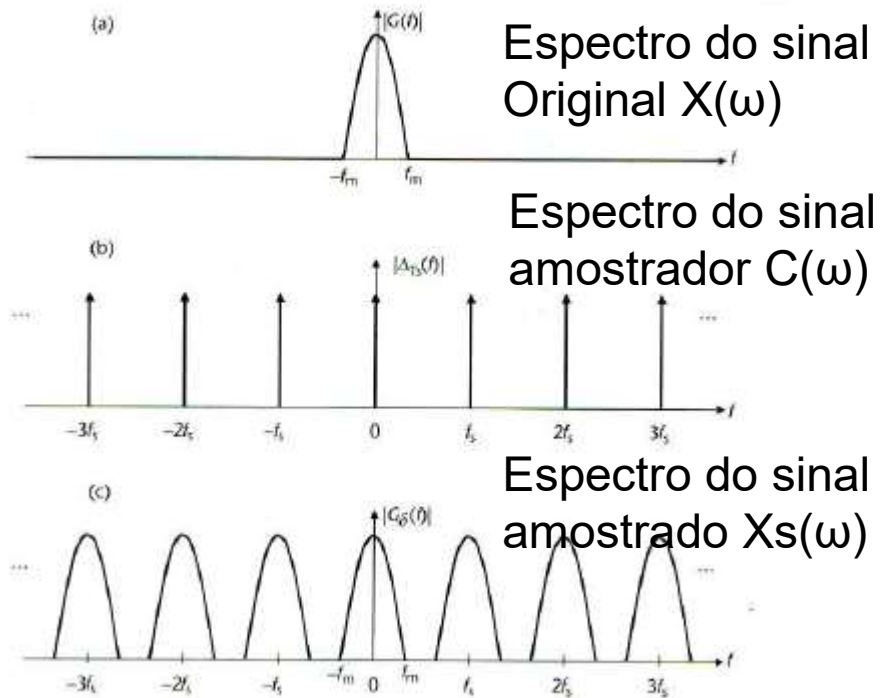
$$x_s(nT_s) = x(t)c(t) \xleftrightarrow{TF} X_s(\omega) = X(\omega) * C(\omega)$$

Produto no domínio do tempo =
Convolução no domínio da Frequência

- Espectro do sinal amostrado $X_s(\omega)$ constituído por:

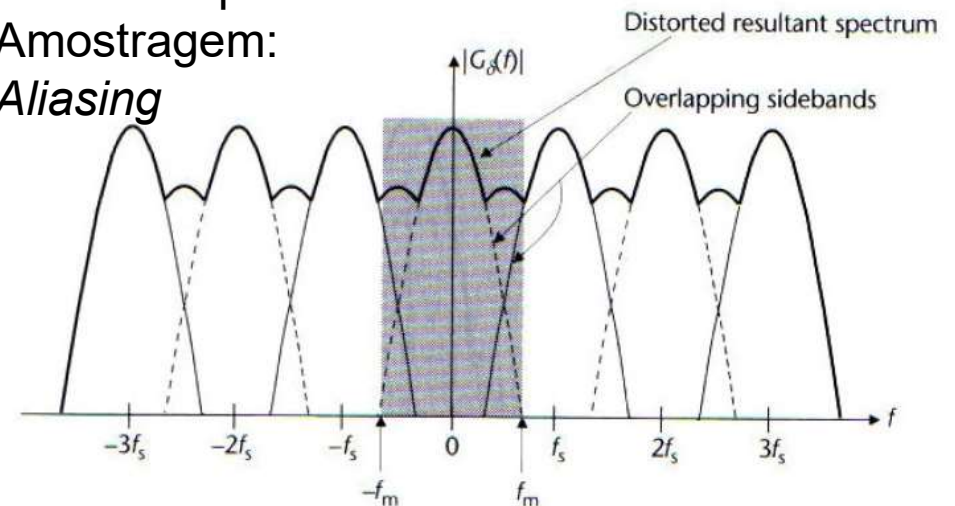
- Réplicas do espectro original $X(\omega)$ centradas em múltiplos de F_s

- Quando $F_s < F_{\text{nyquist}}$ ocorre Aliasing

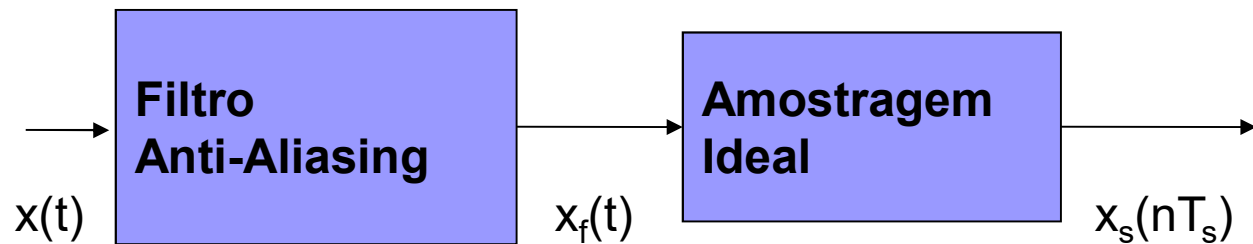


Espectro do sinal amostrado $X_s(\omega)$ quando não é cumprido o Teorema da Amostragem:

Aliasing



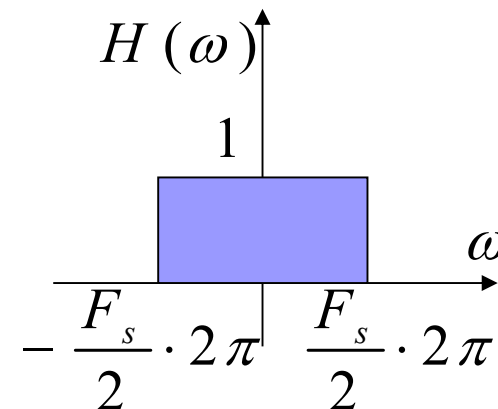
Amostragem – Filtro Anti-Aliasing



- Para garantir que a amostragem é bem sucedida é usado um filtro Anti-Aliasing

■ Filtro Anti-Aliasing

- Filtro passa baixo Ideal

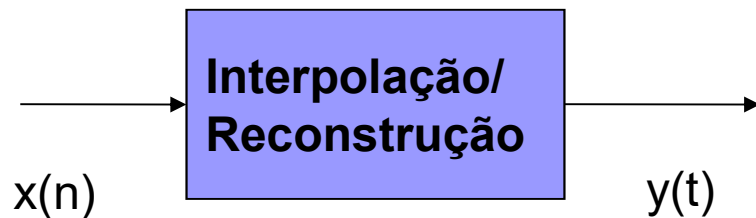




INTERPOLAÇÃO

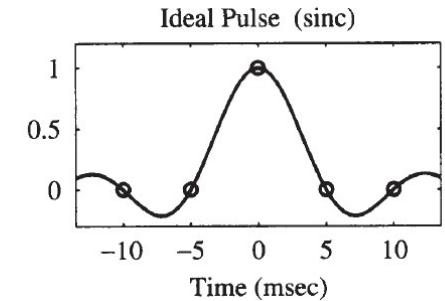
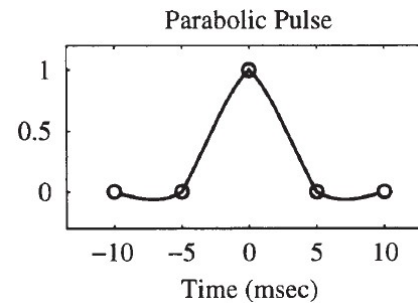
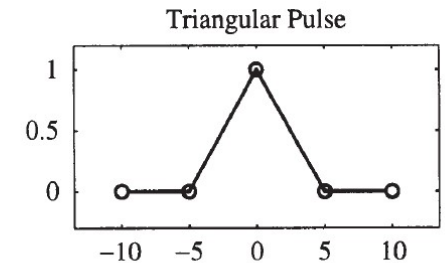
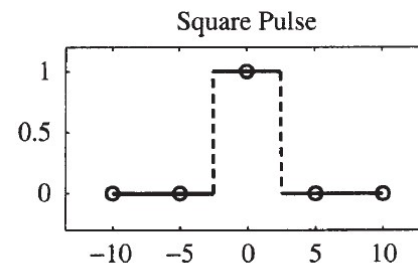


Amostragem – Reconstrução/ Interpolação

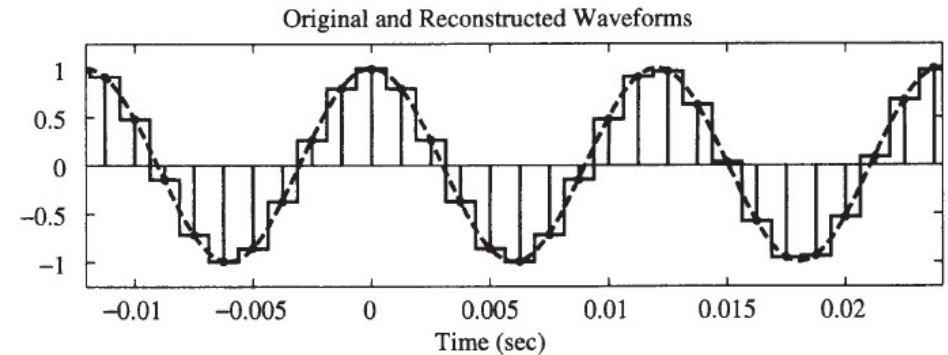
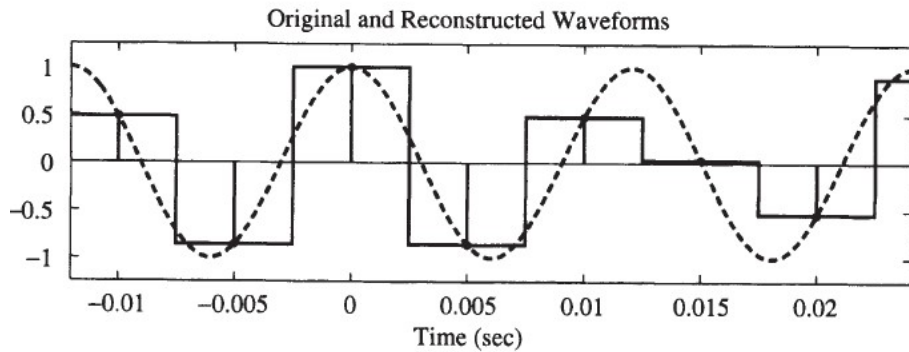
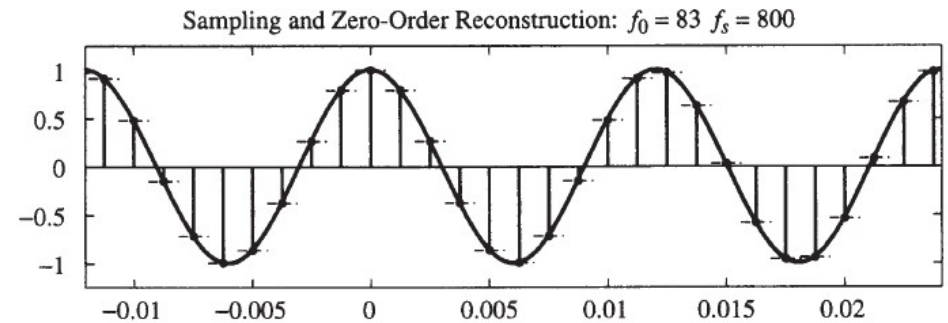
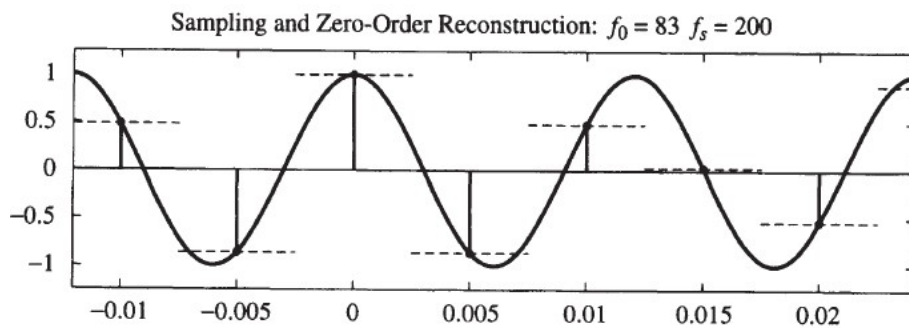
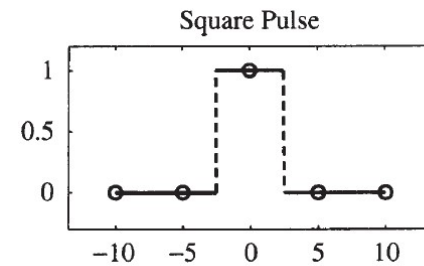


$$y(t) = \sum_{n=-\infty}^{+\infty} y[n] p(t - nTs)$$

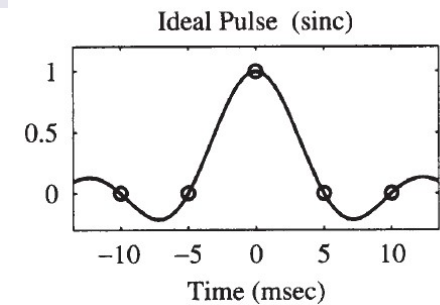
■ Exemplos de funções interpoladoras



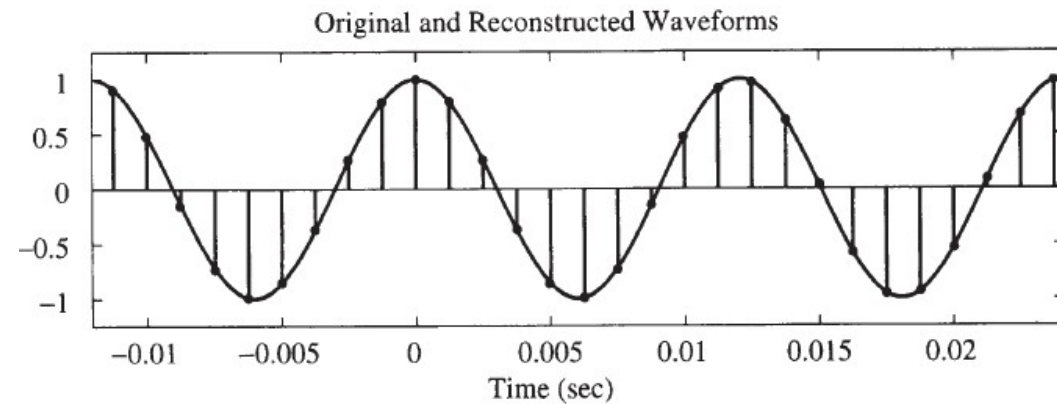
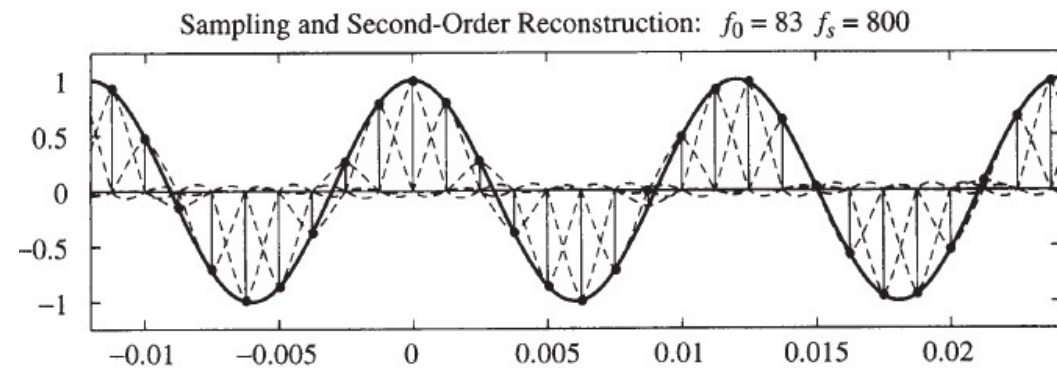
Zero-Order Hold



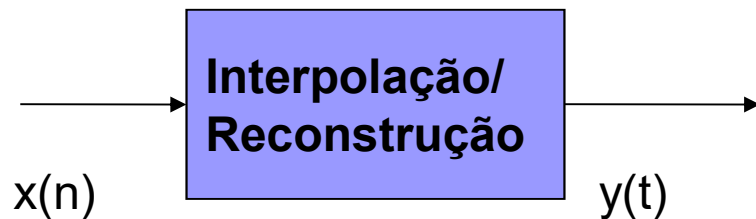
Reconstrução Ideal



$$p(t) = \text{sinc}\left(\frac{t}{T_s}\right) = \frac{\sin\left(\frac{\pi t}{T_s}\right)}{\frac{\pi t}{T_s}}$$



Amostragem – Reconstrução/ Interpolação

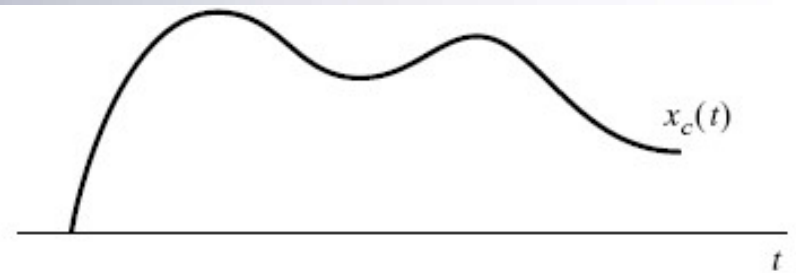


- Exemplo de função interpoladora:

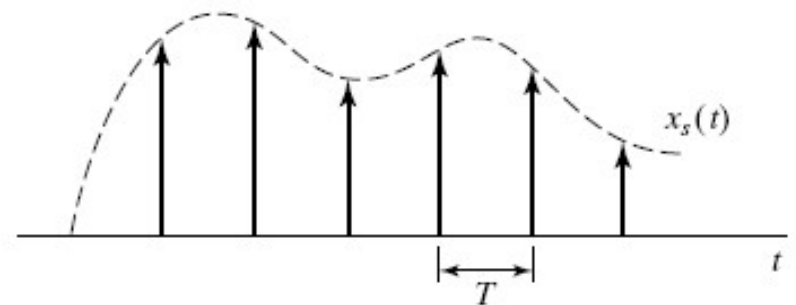
$$\Phi(t) = \text{sinc}(t)$$

$$y(t) = \sum_{n=-\infty}^{\infty} x(n)\phi(t - n)$$

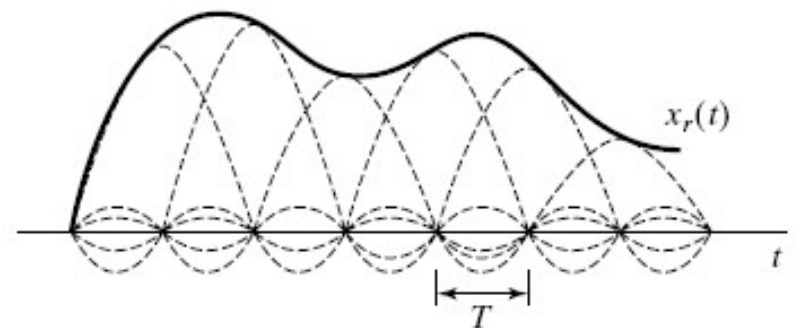
- $y(t) = x(t)$



(a)



(b)



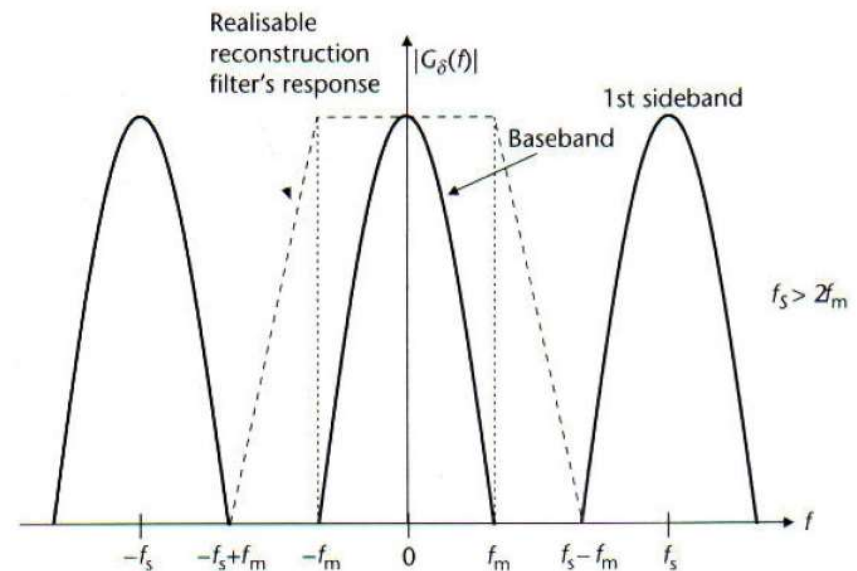
(c)



Amostragem – processo completo



- Processo de obter o sinal original a partir das amostras chama-se Reconstrução (ou Interpolação)
- Reconstrução realiza-se a partir de um filtro Passa Baixo Ideal



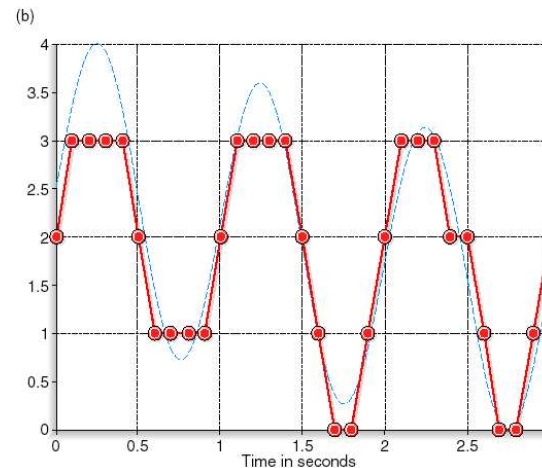
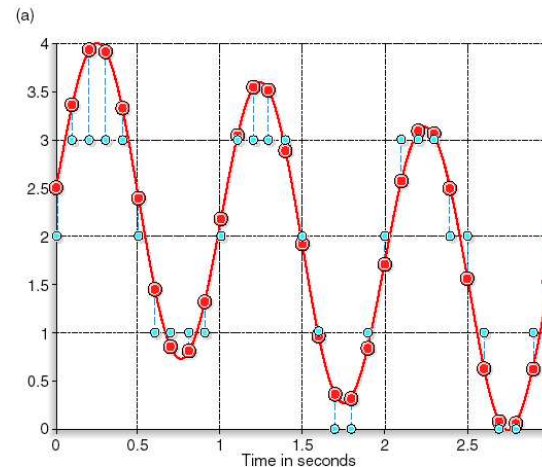


QUANTIZAÇÃO



Armazenar Amostra com Bits: Quantificação

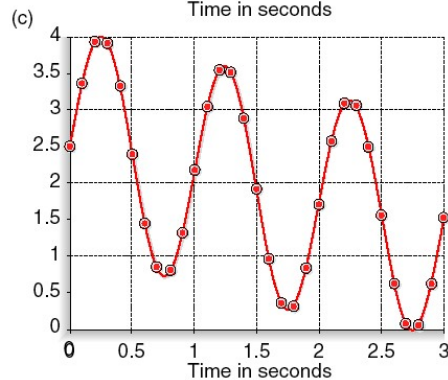
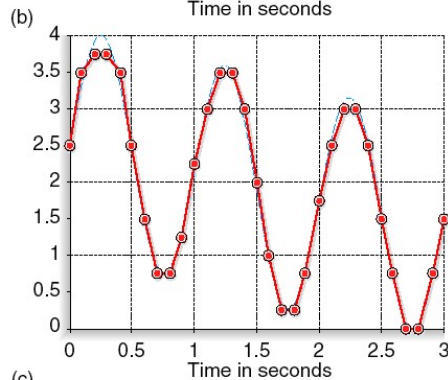
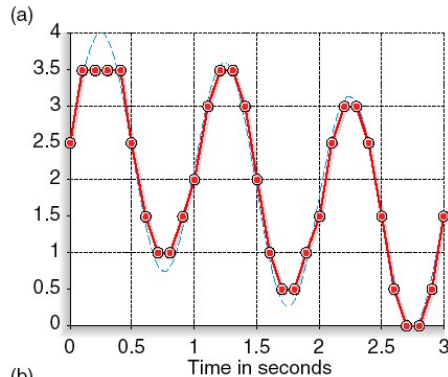
- Os dispositivos digitais utilizam um número limitado de bits para armazenar cada amostra de um sinal.
- Os erros causados pela quantificação é visto ou escutado como *ruído*.



Exemplo:
2 bits por
amostra



Mais Bits Significa Menos Erros



- Exemplo:
 - Topo: 3 bits por amostra
 - Centro: 4 bits por amostra
 - Fundo: 16 bits por amostra
- Mais bits significa mais precisão, mas mais memória e esforço

