```
Processamento Digital de Sinais
         Amostragem de sinais de tempo contínuo
         Grupo formado por:
             Danilo Raposo
             Janderson Barboza
             Rodrigo Cunha
         Os tópicos que são abordados serão os seguintes:

    Fundamentação dos sinais de tempo contínuo

           • Fundamentação do processo de amostragem
           • Teorema de Nyquist
           · Tipos de amostragem
              Amostragem com Impulsos
              Amostragem Natural

    Amostragem sample and hold

    Considerações finais

    Referências

In [1]: #Importando as bibliotecas que serão necessárias para realização do processo de exemplificaç
         import matplotlib.pyplot as plt
         import seaborn as sns
          import pandas as pd
         import numpy as np
         import numpy as np
         import scipy.signal as sng
          import matplotlib.animation as animation
         from IPython.display import Image
         from IPython.core.display import HTML
          #Importando os valores que serão utilizados de forma recorrente na apresentação desse trabal
         pi = 2 * np.pi
                                                         #Definição do valor de pi
          Uma nota importate, como muitas linhas desse código utilizam variáveis com a mesma nomenclat
         ura, recomenda-se que
         seja executado as linhas do código de forma sequêncial, do contrário é bem provável que seja
         obtido representações
         não esperadas que por sua vez são errôneas diante do que foi objetivado.
         Fundamentação dos sinais de tempo contínuo
                 Qualquer sinal que é gerado e obtido através da natureza transmite informação que,
             contendo dados relevantes ou não, pode ser representado a partir de conjuntos matemáti
             cos, definindo características únicas do mesmo, tais como sua amplitude, frequência ou
             até mesmo a extensão do seu sinal quando representado no tempo. Porém, por se tratar d
             e um sinal que tem como principal aspecto a quantidade infinita de valores em um único
             instante de tempo observado, o mesmo apresenta uma susceptibilidade tamanha a fatores
              adtivos que comprometeriam a utilização da informação uma vez transmitida, tomando co
             mo exemplo a prórpia voz trafegando de um ponto a outro. Certamente vale mencionar que
             o ruído não é o único fator que pode acarretar na alteração do sinal transmitido, poré
             m a variável noise será tida como representante de quaisquer alterações no sinal para
              facilitar a compreensão.
                 Uma forma de contornar tal problema é fazendo o processo de digitalização do sina
             l, que consiste em transformar tanto os níveis de amplitude como o período do sinal, u
             ma vez infinitos, em valores finitos, facilitando a visualização de eventuais erros du
             rante o processo de transmissão da informação e certamente também quanto ao processo d
             e correção da mesma no receptor.
In [2]: tmin = 0
          tmax = 100
          t = np.linspace(tmin, tmax, 100)
                                                              #Gerando a margem gráfica
         f = 1/tmax
                                                              #Definição do ruído
         noise = np.random.normal(0,1,100)
         x = 5*np.cos(pi*t)
                                                              #Representação do sinal sem ruído
         xn = (5*np.cos(pi*t)) + noise
                                                              #Representação do ruído mencionado
          'plot'
          plt.figure(figsize=(10,4))
          plt.ylabel('Amplitude')
         plt.xlabel('Tempo')
         plt.title('Sinal cosseinoidal com f = {} Hz [Fig 1.0]'.format(f))
         plt.plot(t, x, 'c', label="Sem ruído")
         plt.plot(t, xn, 'pink', label="Com ruído")
         plt.legend();
                                 Sinal cosseinoidal com f = 0.01 Hz [Fig 1.0]
            -4
                    Sem ruído
            -6
                    Com ruído
                                                                                 100
                              20
                                           40
                                                        60
                                                                    80
                                                Tempo
         Fundamentação do processo de amostragem
                 O processo de amostragem, por sua vez, é o primeiro passo que deve ser tomado para
             alcançar o objetivo de tornar o sinal menos susceptível ao ruído inerente ao processo
              de comunicação, visando, certamente, a digitalização do sinal.
                 É possível visualizar a seguir a notação mais comum diante da representação de um
              sinal discreto no tempo, sendo ela uma sequência de números x. Sendo o valor no insta
             nte x[n] o valor da função x(t) no instante nT, sendo n um número inteiro.
                x = x[n] = x(nTs), -\infty < n < \infty [1.0]
                 Conceituando um pouco melhor o que foi mencionado, digitalização de um sinal possa
             de fato ser realizado, o sinal primordiamente deverá ser convoluído com um trem de imp
             ulsos, e a partir do uso de suas propriedades, será obtido um sinal que tem uma aplitu
             de correspondente as amplitudes do sinal original, porém com um período n discreto.
                 Abaixo será apresentado o impulso unitário e também o trem de pulsos (que nada mai
             s é do que a soma sequencial de impulsos unitários deslocados em n)
                 Impulso unitário
                 Área do impulso unitário
                \int \delta(t) = 1 [2.1]
                 Representação gráfica do impulso unitário deslocado no tempo:
In [3]: n1 = np.arange(-5,6)
                                                                               #Gerando a margem gráfica
          impulse = sng.unit_impulse(11, 'mid')
                                                                               #Criando o impulso unitári
         o (inexistente na prática)
          'plot'
          plt.figure(figsize=(10,4))
          plt.stem(n1, impulse, 'c', markerfmt = ' ',use_line_collection=True)
         plt.ylabel('Area')
         plt.xlabel('Tempo')
          plt.title('Gráfico representativo do impulso unitário [Fig 2.0]');
                               Gráfico representativo do impulso unitário [Fig 2.0]
            1.0
            0.8
            0.6
            0.4
            0.2
            0.0
                                    -2
                                                Tempo
                 Trem de impulsos
                 Representação gráfica do trem de impulsos:
In [4]: n1 = np.arange(-5,6)
                                                                               #Gerando a margem gráfica
         impulse = 0
                                                                               #Representação do Trem de
          pulsos
          for i in range(11):
              impulse = impulse + sng.unit_impulse(11, i)
          'plot'
          plt.figure(figsize=(10,4))
          plt.stem(n1, impulse, 'c', markerfmt = ' ',use_line_collection=True)
          plt.ylabel('Area')
          plt.xlabel('Tempo')
          plt.title('Gráfico representativo do Trem de impulsos [Fig 3.0]');
                              Gráfico representativo do Trem de impulsos [Fig 3.0]
            1.0
            0.8
            0.6
            0.4
            0.2
            0.0
                                                  Ó
                                                Tempo
                 Com a visualização do trem de pulsos que será responsável por amostrar o sinal, a
              fórmula matemática que devemos nos ater frente ao processo de amostragem toma como de
             finição o seguinte:
                xs(t) = \sum
                         x(nTs)\delta(t-nTs) [4.0]
                 Para que possa ficar um pouco mais claro a aplicação do trem de impulsos, abaixo s
             erá apresentado um exemplo utilizando o sinal que foi visto anteriormente.
In [5]: tmin = 0
          tmax = 30
          t1 = np.linspace(tmin, tmax, 30)
         n = np.arange(30)
                                                              #Definição de n pontos discretos
         x1 = np.cos(pi*t1)
                                                              #Representação do sinal sem ruído
                                                              #Processo direto de amostragem (sem convol
         nt = 1/30
         ução de forma direta)
         xs = np.cos(pi * n * nt)
          plt.figure(figsize=(10,4))
         plt.ylabel('Amplitude')
         plt.xlabel('Tempo')
          plt.plot(t1, x1, 'pink', alpha = 0.8, label='Sinal Analógico')
         plt.stem(n , xs, 'c', markerfmt=' ', use_line_collection=True, label="sinal amostrado")
         plt.title('Gráfico representativo do sinal cossenoidal amostrado [Fig 4.0]')
         plt.legend();
                            Gráfico representativo do sinal cossenoidal amostrado [Fig 4.0]
             1.00
             0.75
             0.50
             0.25
             0.00
            -0.25
            -0.50
            -0.75
                     Sinal Analógico
                     sinal amostrado
            -1.00
                                        10
                                                   15
                                                              20
                                                                        25
                                                                                   30
                                                  Tempo
In [6]: impulse = 0
                                                                    #Representação do Trem de pulsos
         for i in range(30):
              impulse = impulse + sng.unit_impulse(30, i)
          conv = x1 * impulse
                                                                   #Convolução
          'plot'
          plt.figure(figsize=(10,4))
         plt.ylabel('Amplitude')
         plt.xlabel('Tempo')
         plt.stem(n, conv, 'c', markerfmt=' ', use_line_collection=True)
         plt.title('Gráfico representativo do sinal cossenoidal amostrado II [Fig 4.1]');
                           Gráfico representativo do sinal cossenoidal amostrado II [Fig 4.1]
             1.00
             0.75
             0.50
             0.25
          Amplitude
             0.00
            -0.25
            -0.50
            -0.75
            -1.00
                                         10
                                                               20
                                                                          25
                                                                                     30
                                                    15
                                                  Tempo
                 Conforme pode ter sido deduzido a partir da apresentação acima feita, o processo d
             e amostragem varia conforme a quantidade de amostras que é retirada do sinal que preci
             sa pretende-se amostrar, e essa variável está sendo representada pelo Ts na equação 4.
             0.
         Teorema de Nyquist
                 A frequência de amostragem (ou período de amostragem, ambos simbolizam a mesma ope
             ração) ressalta consigo um tema bastante importante, tema esse titulado como Teorema d
             e amostragem. Tal proposição delimita que um sinal limitado em banda sem componentes e
             spectrais além de FM, pode ser determinado exclusivamente por valores amostrados em in
             tervalos uniformes, sendo tal teorema representado da seguinte forma:
                Ts \le 1/fm [5.0]
                 Caso a frequência de amostragem não seja maior ou igual a frequência do sinal a se
             r amostrado, serão apresentados ao processo de amostragem uma série de problemas que p
             rejudicarão tanto a visualização do sinal (graficamente falando) como durante o proces
             so de reconstrução do mesmo a partir de um receptor. A amostragem cosiste na multiplic
             ação do sinal pelo impulso unitário, se mudarmos a representação do sinal para o domín
             io da frequência é possível visualizar o quão agressivamente o sinal é prejudicado qua
             nto não considerado o teorema da amostragem.
                 A seguir, será demonstrado graficamente como que a alteração das frequências de am
             ostragem podem fazer com que o sinal seja drasticamente modificado.
In [7]: #Definição de variáveis para teste do tópico comentado acima
          f1 = 200
                                                       #Frequência do sinal analógico
          #Sinal a 400 amostras por segundo
         fs1 = 400
                                                        # Frequência de amostragem do sinal (fs)
                                                        # Duração do período
         T1 = 0.03
                                                                                              (t)
         t1 = np.arange(fs1*T1)/fs1
                                                        # Gerando a margem gráfica
          #Sinal a 900 amostras por segundo
                                                        # Frequência de amostragem do sinal (fs)
         fs2 = 900
         T2 = 0.03
                                                        # Duração do período
                                                                                              (t)
         t2 = np.arange(fs2*T2)/fs2
                                                        # Gerando a margem gráfica
         #Sinal a 3000 amostras por segundo
         fs3 = 3000
                                                        # Frequência de amostragem do sinal (fs)
         T3 = 0.03
                                                        # Duração do período
          t3 = np.arange(fs3*T3)/fs3
                                                        # Gerando a margem gráfica
In [8]: y1 = np.cos(pi*f1*t1)
          'plot'
         plt.figure(figsize=(12,5))
         plt.plot(t1, y1, 'c')
          plt.xlabel('Tempo')
         plt.ylabel('Amplitude)')
         plt.title('Cosseno de {} Hz amostrado em uma taxa de {} amostras por segundo. [Fig 5.0]'.for
         mat(f1, fs1));
                           Cosseno de 200 Hz amostrado em uma taxa de 400 amostras por segundo. [Fig 5.0]
             1.00
             0.75
             0.50
             0.25
          Amplitude)
             0.00
            -0.25
            -0.50
            -0.75
            -1.00
                   0.000
                                 0.005
                                               0.010
                                                             0.015
                                                                                        0.025
                                                                           0.020
                                                         Tempo
In [9]: y2 = np.cos(pi*f1*t2)
          'plot'
          plt.figure(figsize=(12,5))
         plt.plot(t2, y2, 'c')
         plt.xlabel('Tempo')
         plt.ylabel('Amplitude)')
         plt.title('Cosseno de {} Hz amostrado em uma taxa de {} amostras por segundo. [Fig 5.1]'.for
         mat(f1, fs2));
                           Cosseno de 200 Hz amostrado em uma taxa de 900 amostras por segundo. [Fig 5.1]
             1.00
             0.75
             0.50
             0.25
             0.00
            -0.25
            -0.50
            -0.75
            -1.00
                   0.000
                                0.005
                                             0.010
                                                           0.015
                                                                        0.020
                                                                                     0.025
                                                                                                  0.030
                                                         Tempo
In [10]: y3 = np.cos(pi*f1*t3)
          'plot'
          plt.figure(figsize=(12,5))
          plt.plot(t3, y3, 'c')
          plt.xlabel('Tempo')
         plt.ylabel('Amplitude)')
         plt.title('Cosseno de {} Hz amostrado em uma taxa de {} amostras por segundo. [Fig 5.2]'.for
         mat(f1, fs3));
                          Cosseno de 200 Hz amostrado em uma taxa de 3000 amostras por segundo. [Fig 5.2]
             1.00
             0.75
             0.50
             0.25
          Amplitude)
             0.00
            -0.25
            -0.50
            -0.75
            -1.00
                   0.000
                                0.005
                                             0.010
                                                                       0.020
                                                                                    0.025
                                                                                                0.030
                                                         0.015
                                                         Tempo
         Tipos de amostragem
                 Explicado o processo de amostragem por impulsos e certamente também deixando claro
             a necessidade de se prestar a devida atenção quanto ao período de amostragem do sinal,
             adentramos agora no que seria a delimitação dos tipos de amostragem, serão elas a Amos
             tragem com Impulsos, Amostragem Natural e a Amostragem sample and hold. A amostragem p
             or impulsos foi a fundamentada anteriormente, que é um método ideal e por isso tem sua
             utilidade apenas teórica, já a Amostragem Natural e Sample and hold tem aplicações prá
             ticas e reais, e por isso as mesmas serão estudadas com um pouco mais de cuidado.
         Amostragem Natural
                 A amostragem natural - modelo prático baseado na teoria da amostragem com impulsos
             - utiliza um trem de pulsos retangulares, cujas parcelas do sinal analógico são amostr
             adas nesses pulsos em um tempo T de largura em sua duração, assim a largura das amostr
             as vão tomar o formato do sinal analógico. Quanto menor o intervalo da largura, mais c
             hances de o sinal amostrado se aproximar da amostragem com impulsos.
                 Esse processo de amostragem é aplicado em modulações de sistemas Pulse Amplitude M
             odulation (PAM), cuja saída é a convolução do sinal modulante com o sinal do trem de p
             ulsos.
                 O sistema PAM recebe como exemplo uma função senoidal denotada como x(t), cujo sin
             al apresenta uma grandeza infinita em tempo e amplitude, necessitando-o convolucionar.
             Todo sinal analógico nesse tipo de modulação é preciso deslocar a amplitude positivame
             nte.
                 O sinal que será tomado como exemplo para as explicações a seguir será plotado aba
             ixo.
In [11]: tmin=0; tmax=1;
          t = np.linspace(tmin, tmax, 1000, endpoint=False)
         dslc = 4.5
                                                                         #Deslocamento do sinal x(t)+4.5
         xtpam = 3.5*np.cos(pi*t)+dslc
          plt.figure(figsize=(10,4))
          plt.plot(t,xtpam,'c')
         plt.xlabel('Tempo')
         plt.ylabel('Amplitude')
         plt.title('Sinal analógico [Fig 6.0]')
          plt.show();
                                        Sinal analógico [Fig 6.0]
            3
            2
                                          0.4
                                                                   0.8
                0.0
                             0.2
                                                      0.6
                                                                                1.0
                                               Tempo
                 A função abaixo é exemplo do trem de pulsos. Com o período e a largura de pulso co
             nstantes, ela é denotada conforme abaixo e é o sinal resultante da expressão listada é
             responsável por convolucionar o sinal analógico o qual deseja-se amostrar. Um modelo m
             ais próximo do real é calculado quanto se considera o trem de pulsos de amplitude igua
             l ao inverso de sua largura de banda.
               x_p(t) = \frac{1}{T} \sum_{n=-\infty}^{\infty} c_n e^{j2\pi n f_s t} [6.0]
                 As variáveis fs (frequência de amostragem) e cn (coeficientes da série exponencial
             de Fourier) fazem parte de xp(t), denotadas como:
               f_s = \frac{1}{T_s} [6.1];
c_n = \frac{1}{T_s} sinc(\frac{nT}{T_s}) \text{ ou } c_n = f_s sinc(\frac{nT}{T_s}) [6.2].
In [12]: fs = 1/0.2
                                                                        #Frequência de amostragem
          ts = 1/fs
                                                                        #Período de amostragem
         xpt = (sng.square(2*np.pi*fs*t)+1)/2
                                                                        #Trem de pulsos
          'plot'
          plt.figure(figsize=(10,4))
         plt.plot(t,xpt,'c')
         plt.xlabel('Tempo')
          plt.ylabel('Amplitude')
          plt.title('Trem de pulsos (f_s =  Hzs =  ss) [Fig 6.1]'.format(f_s, ts))
          plt.show();
                                Trem de pulsos (f_s = 5.0Hz e T_s = 0.2s) [Fig 6.1]
            1.0
            0.8
          9.0 Amplitude
            0.2
            0.0
                                                                                 1.0
                 0.0
                                                        0.6
                                                                    0.8
                                                Tempo
                 Amostragem adquirida pela convolução das funções, denotada como
                x_s(t) = x(t) * x_p(t) [7.0]
In [13]: xst = xtpam*xpt
          'plot'
          plt.figure(figsize=(10,4))
         plt.plot(t,xst,'c',label='Sinal amostrado')
         plt.plot(t,xtpam,'r:',label='Sinal analógico',alpha=0.5)
         plt.legend(loc='best')
         plt.xlabel('Tempo')
         plt.ylabel('Amplitude')
         plt.title('Convolução em PAM [Fig 6.2]')
         plt.show();
                                      Convolução em PAM [Fig 6.2]
                                             Sinal amostrado
                                             Sinal analógico
            7
            6
            2
            1
                             0.2
                                         0.4
                                                                   0.8
                                                                                1.0
                0.0
                                                      0.6
                                               Tempo
                 Em sistemas não PAM, o sinal analógico não precisa ter amplitude deslocada, portan
             to:
In [14]: xtnpam = xtpam-dslc
                                                                         #Deslocamento x(t)-4.5
          xst2 = xtnpam*xpt
          'plot'
          plt.figure(figsize=(10,4))
          plt.plot(t,xst2,'c',label='Sinal amostrado')
          plt.plot(t,xtnpam,'r:',label='Sinal analógico',alpha=0.5)
          plt.legend(loc='best')
          plt.xlabel('Tempo')
          plt.ylabel('Amplitude')
          plt.title('Convolução sem deslocamento [Fig 6.2]')
         plt.show();
                                   Convolução sem deslocamento [Fig 6.2]
             3
             2
             0
            -1
            -2
                   Sinal amostrado
            -3
                   Sinal analógico
                 0.0
                                           0.4
                                                                                 1.0
                                                Tempo
         Amostragem Sample and Hold
                 Circuitos Sample and Hold, também conhecidos como amostragem e retenção geralmente
             são utilizados com conversores de sinais analógicos para digital. O intuito deste proc
             esso é segurar o sinal de entrada analógico e segurar o sinal amostrado.
                 Alguns circuitos Sample-and-Hold recebem pulso externo, mas ele também pode possu
             ir seu próprio pulso interno também conhecido por alguns como clock. Além disso os cir
             cuitos Sample-and-Hold também pode possuem seu próprio gerador interno de ruído branc
             o que pode ser usado na mixação com outro sinal de entrada produzindo efeitos de som.
              Uma curiosidade é que muitas empresas de som, fabricantes utilizam desse recurso já i
             mplementado como ferramenta.
                 Para que o processo ocorra o sinal recebido deve estar estático, enquanto a chave
              do circuito estiver fechada o sinal de entrada é amostrado e o valor do mesmo é armaz
             enado no capacitor do circuito, caso contrário, enquanto a chave estiver aberta a conv
             ersão analógica/digital é efetuada. Toda vez que uma novo sinal estático chega, a chav
             e do conversor analógico/digital alterna entre aberto e fechada, estando assim em sicr
             onia com o mesmo.
                 Quando o circuito Sample-and-Hold recebe um pulso, o módulo irá guardar na memóri
             a, o nível de voltagem até que o outro pulso chegue para uma nova requisição, caso o s
             inal de seja constituído constituído por séries de voltagens randômicas que pode ser l
             igadas ao controle de entrada de um oscilador, no qual, irá produzir uma série randôm
             ica de notas, caso a saída for uma onda em formato de escada a melodia produzida será
              arpejada.
                 As simulações seguir mostram a entrada e a saída de como um sinal se comporta dian
             te de um sistema S/H:
In [15]: tmin = 0
          tmax = 1
          t = np.linspace(tmin, tmax, 50)
         y = np.cos(t*pi)
          'plot'
          plt.figure(figsize=(10,4))
         plt.ylabel('Amplitude')
         plt.xlabel('Tempo')
         plt.plot(t, y,color = 'c',label = 'Sinal analógico')
         plt.title('Sinal analógico [Fig 7.0]');
                                           Sinal analógico [Fig 7.0]
             0.75
             0.50
             0.25
             0.00
            -0.25
            -0.50
            -0.75
            -1.00
                   0.0
                                0.2
                                                         0.6
                                                                      0.8
                                                                                   1.0
                                             0.4
                                                  Tempo
In [16]: 'plot'
          plt.figure(figsize=(10,4))
          plt.stem(t, y, 'c-', markerfmt=' ', label = "amostragem do sinal analógico", use_line_collec
         tion = True)
         plt.plot(t, y, color = 'pink', label = 'Sinal analógico')
         plt.ylabel('Amplitude')
         plt.xlabel('Tempo')
         plt.legend()
         plt.title('Sinal analógico amostrado [Fig 7.1]');
                                       Sinal analógico amostrado [Fig 7.1]
             1.00
             0.75
             0.50
             0.25
          Amplitude
             0.00
            -0.25
            -0.50
            -0.75
                     Sinal analógico
                   amostragem do sinal analógico
            -1.00
                                                                                   1.0
                   0.0
                                0.2
                                             0.4
                                                         0.6
                                                                      0.8
                                                  Tempo
                 E como bem sabemos, a amostragem é representada pela convolução do trem de pulsos,
             x(t)x\delta(t) por um pulso retangular p(t) de amplitude unitária e largura Ts , assim temo
             s que:
                x_s(t) = p(t) * x(t)\delta(t) [7.0]
                            \delta(t-nTs)] [7.1]
```

Para melhor compreensão das fórmulas aqui descritas será feito, bem como foi feito

anteriormente, a exemplificação de cada uma delas como uma espécie de simulação.

plt.stem(t,y,'pink', label='Amostras do sinal', markerfmt=',', use_line_collection = True)

0.6

Após ter todo esse caminho percorrido diante do assunto de amostragem de sinais an alógicos, por sua vez de tempo e amplitude contínuos, foi possível compreender a impor tância do processo de amostragem nas comunicações digitais e certamente também a neces sidade de se ater a todos os parâmetros matemáticos que a fundamentam, seja a teoria p or trás do impulso unitário (trem de impulsos no geral) ou até mesmo o que nomeamos aq

Também foi possível verificar os tipos de amostragem e a forma com que o sinal (co

Existe, no entanto, uma série de outros pontos que sucedem o processo de amostrage m para que de fato um sinal analógico seja transformado em um sinal digital. O bloco c onversor analógico/digital, também conhecido como conversor A/D, é composto por vários blocos menores que constituem o que chamamos de PCM, que é um processo de modulação qu

Lista-se aqui como pesquisas futuras uma análise quanto ao processo de quantizaçã o, bloco posterior ao de amostragem no conversor A/D, que é responsável por transforma r o sinal de tempo discreto e amplitude contínua, portanto ainda não digital, presente na saída do amostrador e transformá-lo em um sinal com o tempo e sua amplitude discret

1.0

Sinal amostrado com Sample and Hold [Fig 7.2]

Tempo

ui como teorema de amostragem, esse fundamentado por Harry Nyquist.

ssenoidal por padrão) analógico teve uma saída PAM variada em virtude disso.

HAYKIN, S; MOHER, M. Introdução aos Sistemas de comunicação, 2a Ed., Bookman, Porto Alegre, 2008.

Audio Processing in Python Part I: Sampling, Nyquist, and the Fast Fourier Transform. Disponível em:

Teoria e Prática de Processamento de Sinais e Imagens. Disponível em:

https://fei.edu.br/~isanches/verao/curso_de_verao_2017.html. Acesso em: 04/05/2021.

Getting into Digital Signal Processing Part 2: Sampling & Aliasing. Disponível em: https://www.rs-

online.com/designspark/getting-into-digital-signal-processing-sampling-aliasing. Acesso em: 05/05/2021.

LATHI, B. P.; ZHI DING. Modern Digital and Analog Communication Systems, 4a Ed., OXFORD UNIVERSITY PRESS, New

https://makersportal.com/blog/2018/9/13/audio-processing-in-python-part-i-sampling-and-the-fast-fourier-transform. Acesso

Sample and Hold Circuit. Disponível em: https://www.electronicshub.org/sample-and-hold-circuit/. Acesso em: 06/05/2021.

plt.step(t, y,label='Retenção do sinal analógico',color = 'cyan')

plt.title('Sinal amostrado com Sample and Hold [Fig 7.2]');

In [17]: 'plot'

plt.figure(figsize=(10,4))

Retenção do sinal analógico

e não está no escopo desse documento.

Amostras do sinal

Considerações finais

plt.ylabel('Amplitude')
plt.xlabel('Tempo')

plt.legend();

0.75 0.50 0.25

0.00 -0.25 -0.50 -0.75

-1.00

izadas.

Referências

York, 2010.

em: 03/05/2021.

Amplitude