



# **PMR3404 Controle I**

## **Aula 1**

Introdução aos Sistemas de Controle de Tempo Discreto  
Controle Digital

---

Newton Maruyama  
17 de agosto de 2021

PMR-EPUSP

# Conteúdo

1. Cronologia histórica
2. Sistemas de controle digital
3. Sistemas de aquisição de dados
4. Condições para que o sistema de controle digital se aproxime do equivalente de tempo contínuo
5. Exemplo: Controle digital para um sistema massa-mola
6. Reconstrução de sinais

## Cronología histórica

---

## Uma cronologia histórica - 1

- A idéia da discretização do tempo como técnica de solução numérica está na origem da própria concepção do cálculo integral e diferencial
- Tecnologicamente o interesse por amostragem de sinais analógicos se inicia com a idéia de multiplexação de canais em sistemas telegráficos (1870) e telefônicos (1910).
- Vários desenvolvimentos matemáticos de análise de sinais foram desenvolvidos no *Bell Labs* na primeira metade do século 20 por pesquisadores como Nyquist e Shannon.
- as idéias básicas do que hoje é conhecido como Transformada Z já era conhecida por Laplace e foi re-introduzida por W. Hurewicz em 1947 além de Ragazzini, Zadeh e Juri.

## Uma cronologia histórica - 2

- Controlador PID mecânico - Sperry (1911).
- Controlador PID pneumático - Taylor Instruments (1939).
- Controlador PID semicondutor - Bailey Meter Co. (1959).
- Controlador PID digital - Taylor Instruments (1964)

## Sistemas de controle de tempo real: mísseis, foguetes, aviões

- Os sistemas de controle digital se são implementados em uma infra-estrutura de hardware e software adequados para cada aplicação.
- Os sistemas críticos são de particular interesse.
- Sistemas de tempo real devem prever o tempo de resposta máxima para um programa reagir.



## Acidente - lançador de satélites Ariane 5, 1996

- Primeira missão
- Custo de desenvolvimento durante 10 anos US\$7.0 Bilhões
- Carga: 4 satélites da Agência Espacial Européia (US\$ 370 Milhões)



- Sistema de controle de atitude reaproveitado do Ariane 4,
- Intervalo de acelerações maiores no Ariane 5,
- Conversão de tipo de ponto flutuante (64 bits) para inteiro (16 bits),
- Falha na verificação do software.



## **Sistemas de controle digital**

---

## Sistemas dinâmicos de tempo discreto

- Os sistemas dinâmicos de tempo discreto são descritos através de equações de diferenças:

$$u(k) = u(k - 1) + 1.5e(k) - 0.5e(k - 1)$$

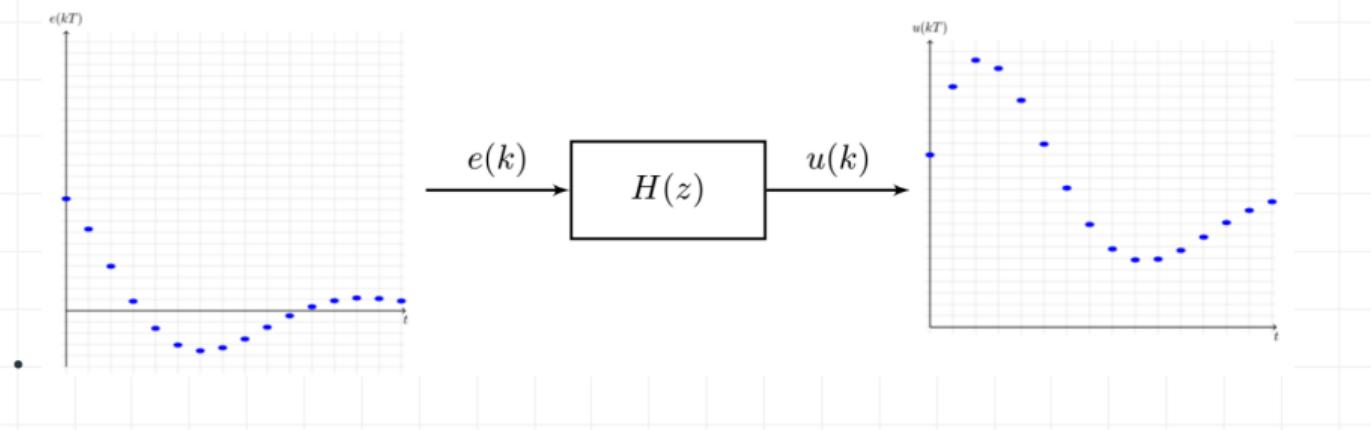
- $e(k) =$

$\{1, 0.72, 0.39, 0.08, -0.15, -0.30, -0.35, -0.33, -0.25, -0.14, -0.04, 0.03, 0.09, 0.11, 0.11, 0.09\}$

$k = 0, \dots, 15$

- $u(k) =$

$\{1.50, 2.09, 2.32, 2.25, 1.97, 1.59, 1.21, 0.89, 0.68, 0.59, 0.59, 0.67, 0.78, 0.91, 1.02, 1.09\}$



## Tempo contínuo

- Equação diferencial:

$$a_0y^{(n)} + a_1y^{(n-1)} + \cdots + a_{n-1}y^{(1)} + a_ny = b_0u^{(m)} + b_1x^{m-1} + \cdots + b_{(m-1)}x^{(1)} + b_mx \quad (n \geq m)$$

- Função de transferência variável complexa s:

$$\frac{Y(s)}{U(s)} = \frac{b_0s^m + b_1s^{m-1} + \cdots + b_{m-1}s + b_m}{a_0s^n + a_1s^{n-1} + \cdots + a_{n-1}s + a_n}$$

Transformação  
de Laplace

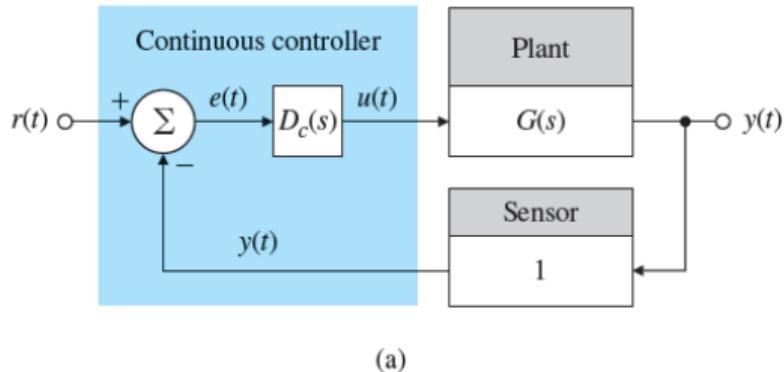
- Espaço de estados:

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t)$$

## Sistema de controle de tempo contínuo

- A figura abaixo representa um sistema de controle em malha fechada em tempo contínuo:



**Figura 1:** Fonte: (Franklin, Powell & Emami-Naeini,2019)

- Controle contínuo e controle de tempo contínuo são utilizados como sinônimos.

## Tempo discreto

- Equação de diferenças:

$$y(k+n) + a_{n-1}y(k+n-1) + \dots + a_0y(k) = b_nu(k+n) + \dots + b_0u(k)$$

- Função de transferência variável complexa z:

$$\frac{Y(z)}{U(z)} = \frac{b_n + z^{-1}b_{n-1} + \dots + z^{-n}b_0}{1 + z^{-1}a_{n-1} + \dots + z^{-n}a_0}$$

Transf.  
z

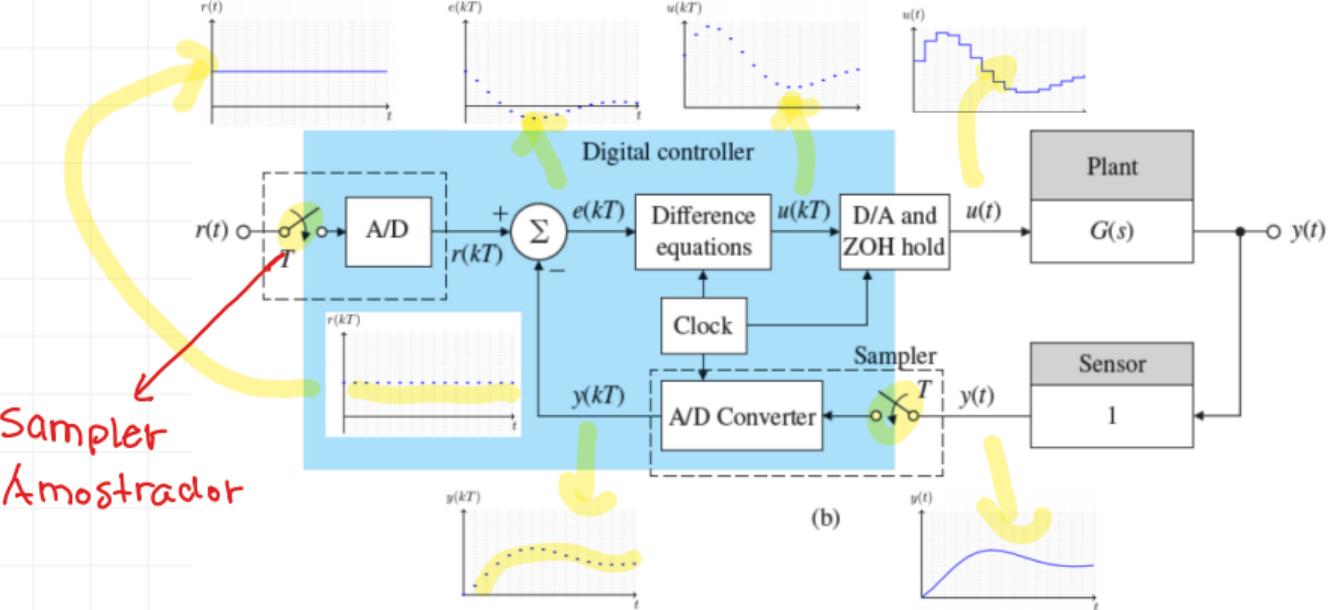
- Espaço de estados:

$$x(k+1) = Ax(k) + Bu(k)$$

$$y(k) = Cx(k)$$

# Sistema de controle de tempo discreto

- Sistema de controle e sinais associados:



**Figura 2:** Adaptação de (Franklin, Powell & Emami-Naeini, 2019)

- Controle discreto, controle de tempo discreto e controle digital são utilizados como sinônimos.

## Controlador digital

- **Amostrador e Conversor A/D:** uma amostragem de um sinal de tensão é realizada e posteriormente esse valor de tensão é transformado em uma representação interna do computador;
- **Equações de diferenças:** o algoritmo do controlador é executado pela CPU utilizando uma representação por equações de diferenças;
- **Relógio:** as operações são realizadas através de interrupções da CPU de maneira sincronizada a cada intervalo de tempo  $T$  (período de amostragem) utilizando como referência um clock de tempo real;
- **Conversor D/A e Segurador de Ordem-Zero<sup>1</sup>:** o sinal de controle  $u(kT)$  é convertido para um valor de tensão contínuo que é mantido constante no intervalo de tempo  $[kT, (k + 1)T[$

---

<sup>1</sup>Retentor de Ordem-Zero

## Algumas observações

1. O sistema de controle de tempo discreto é na verdade um **sistema híbrido** pois contém variáveis de tempo contínuo e de tempo discreto.
2. Todos os valores das variáveis de tempo contínuo ou de tempo discreto são números reais.
3. O sinal de controle  $u(t)$  possui um atraso na média igual a  $T/2$  devido à atuação do segurador de ordem-zero.
4. O sinal de saída  $u(t)$  possui uma característica tipo variação em "escada".
5. A figura abaixo ilustra essa questão. Note que uma curva hipotética contínua que passa sobre os pontos discretos  $u(kT)$ . Essa curva seria o sinal de controle ideal.

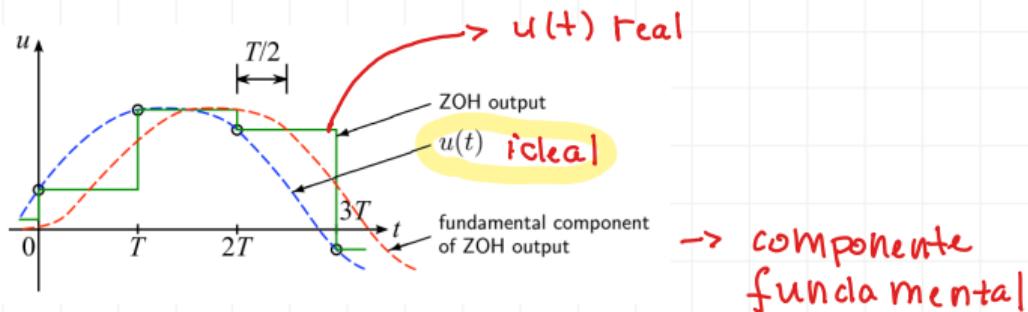
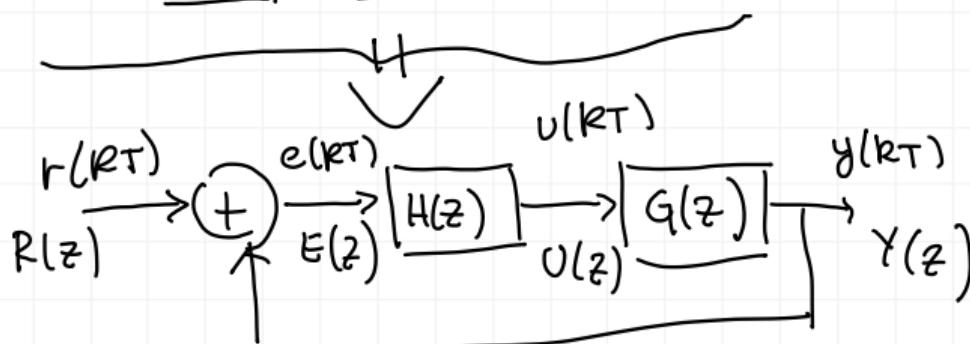
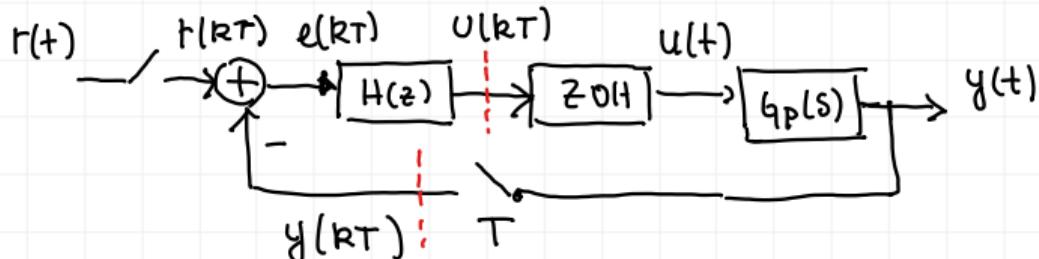


Figura 3: Fonte: Lecure Notes Mark Cannon

## Sistema de controle digital - transformação para variáveis discretas



CONVERSÃO para Modelo  
discreto  
para projeto e  
análise

## **Sistemas de aquisição de dados**

---

# Sistema de aquisição de dados

- Alto custo:

1. National Instruments PCIe6321: 16 canais de entradas analógicas (8 no modo diferencial), resolução de 16 bits, 2 canais de saídas analógicas, 8 pinos de E/S digital, taxa de amostragem máxima 250kS/s (na utilização de 1 canal),



2. National Instruments CompactRio: Sistema modular



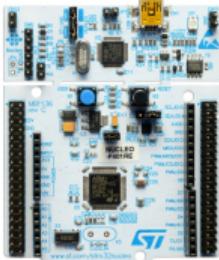
# Sistemas de Aquisição de dados

- Baixo custo:

1. Arduino Uno: 6 entradas analógicas, resolução de 10 bits, taxa de amostragem típica 9600Samples/s (depende do clock), saídas PWM analógicas.

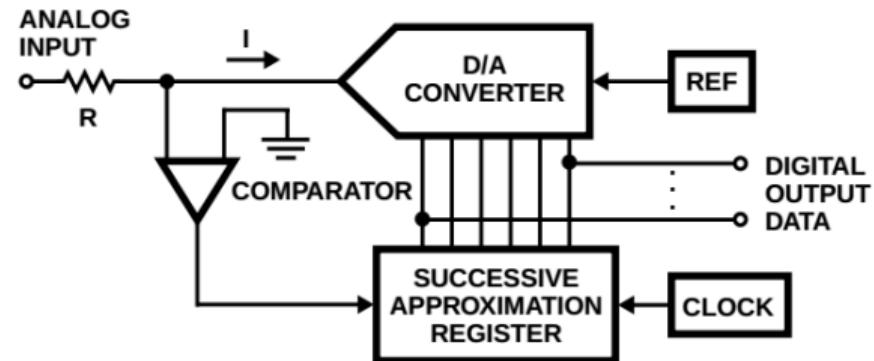


2. ST STM32 Nucleo: 16 entradas analógicas, 2.4MS/s.



# Conversor Analógico-Digital

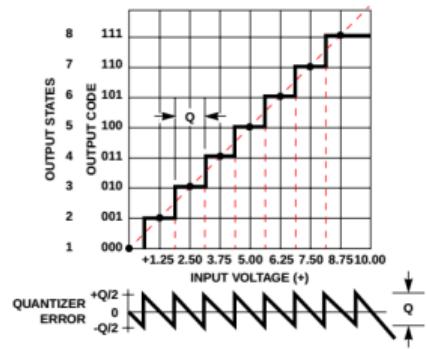
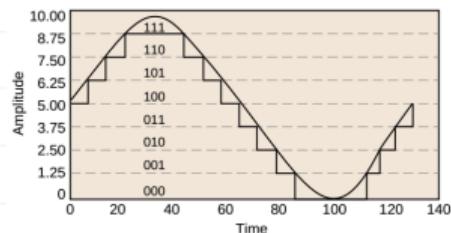
- Um possível diagrama esquemático para um circuito A/D - método de aproximações sucessivas:



- Um contador conectado a um circuito D/A gera valores de tensão crescentes que são comparados com o valor de entrada até atingir o valor de tensão correta.

# Quantização

- **Resolução:** corresponde ao número de bits utilizado no circuito.
- **Intervalo de quantização:** define a faixa de tensão elétrica (valores mínimo e máximo).
- Largura de 1 bit, **resolução de tensão**  $Q$ : define a faixa de tensão representada pela variação de 1 bit.
- Figura: sinal senoidal, resolução de 3 bits, intervalo de quantização de  $[0, +10V]$ .
- A resolução de tensão é dada por:  $Q = \frac{10}{2^3} = 1.25V$ .



- Note que valores intermediários são arredondados durante a conversão o que acarreta no chamado **erro de quantização**. Note que o erro de quantização oscila entre  $[-Q/2, +Q/2]$ .

## Quantização

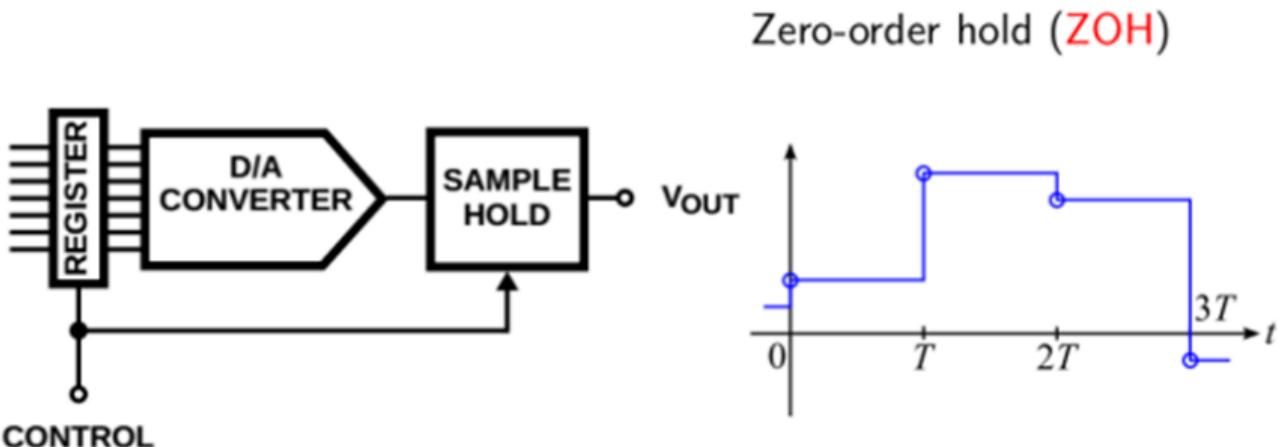
- Há algumas décadas atrás os conversores A/D mais comuns possuíam uma resolução de apenas 8 bits, resoluções maiores possuíam custos muito elevados.
- Atualmente a maioria dos sistemas possui 10bits ou 12 bits mas há sistemas de 24bits.
- o sistema DAQ AT-MIO-16X da National Instruments utiliza 16 bits e possui intervalos de quantização selecionáveis entre [0,+10V] e [-10V,+10V], além de ganhos selecionáveis de 1, 2, 5, 10, 20, 50 a 100. Com uma tensão no intervalo de [0,+10V] e um ganho de 100, a resolução de tensão é dada por:

$$\frac{10}{100 \times 2^{16}} = 1.5\mu V$$

Portanto, a resolução teórica para 1 bit é dada pelo valor  $1.5\mu V$ .

# Conversor Digital-Analógico

- Um possível diagrama esquemático para um circuito D/A:



- Os sistemas DAQ industriais estão sempre associados a um circuito do tipo Zero-order hold (ZOH).

**Condições para que o sistema de  
controle digital se aproxime do  
equivalente de tempo contínuo**

---

## As condições ideais

- Não é difícil chegar a conclusão que um sistema de controle digital idealmente deve se comportar exatamente como seu correspondente sistema de controle de tempo contínuo.
- Dois parâmetros definem o quão próximo se encontram os dois sistemas:
  1. Período de amostragem  $T$ : idealmente  $T \rightarrow 0$ .
  2. Resolução de tensão  $Q$ : idealmente  $Q \rightarrow 0$ .

## Na prática

- $$\frac{x(k) - x(k-1)}{T} \rightarrow \dot{x}(t)$$
- A frequencia angular de amostragem  $\omega_s = \frac{2\pi}{T}$  deve ser 50 vezes maior que a **largura de banda**  $B_w$  do sistema de controle em malha fechada:

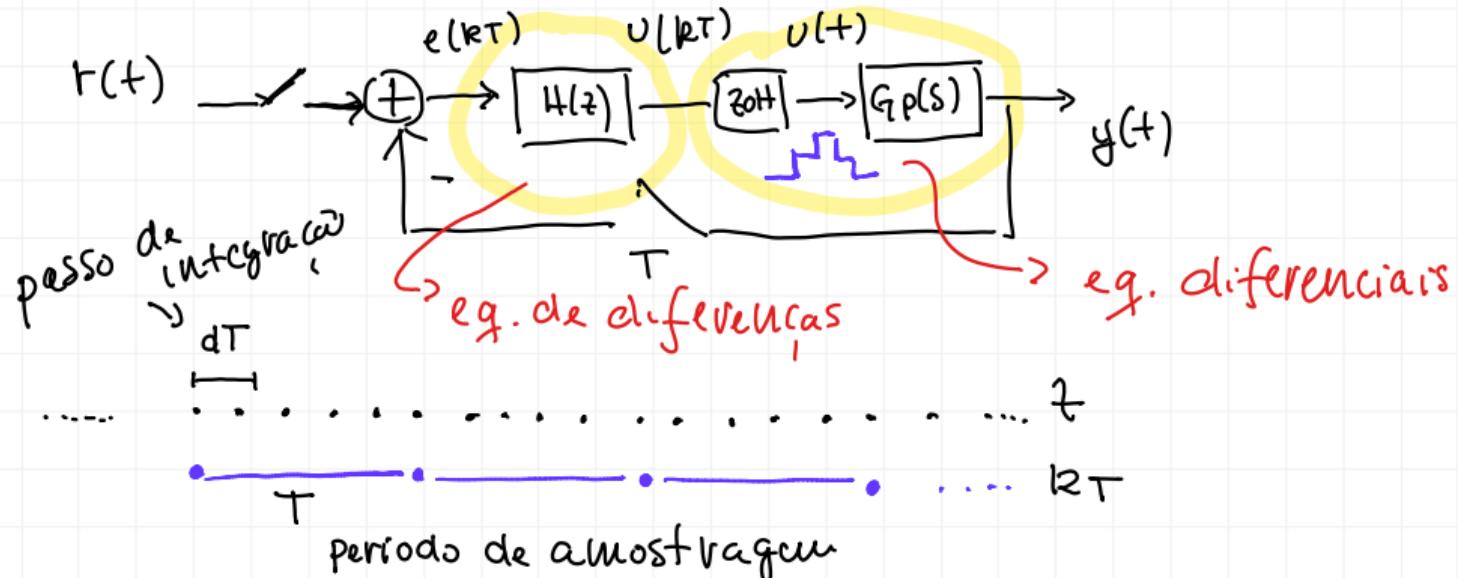
$$\omega_s \geq 50B_w$$

- A resolução do DAQ  $nbits$  deve ser:

$$nbits >= 16.$$

A resolução de tensão  $Q$  para  $nbits = 16$  é ( $2^{16} = 65536$  níveis de tensão) é suficientemente pequena para qualquer faixa de tensão.

# Simulação de um sistema de controle digital



- **Tempo discreto:** a simulação do controlador é modelado através de equações de diferenças considerando o intervalo de amostragem  $T$ .
- **Tempo contínuo:** a planta deve ser simulada através de métodos de integração numéricos com passo de integração que garanta precisão. Vários métodos possíveis: passo único (single step), múltiplos passos, passo variável, etc.

## **Exemplo: Controle digital para um sistema massa-mola**

---

## Exemplo: Sistema massa-mola

- Seja um sistema massa-mola descrito pela seguinte equação:

$$M\ddot{x}(t) + Kx(t) = F(t)$$

onde  $M = 1\text{Kg}$  e  $K = 10\text{N/m}$ .

- A Transformada de Laplace é dada por:

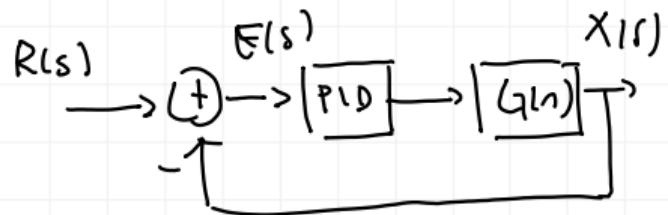
$$\frac{X(s)}{F(s)} = G(s) = \frac{1}{Ms^2 + K}$$

- É proposto a utilização de um controlador PID:

$$H(s) = K_p + K_d s + \frac{K_i}{s}$$

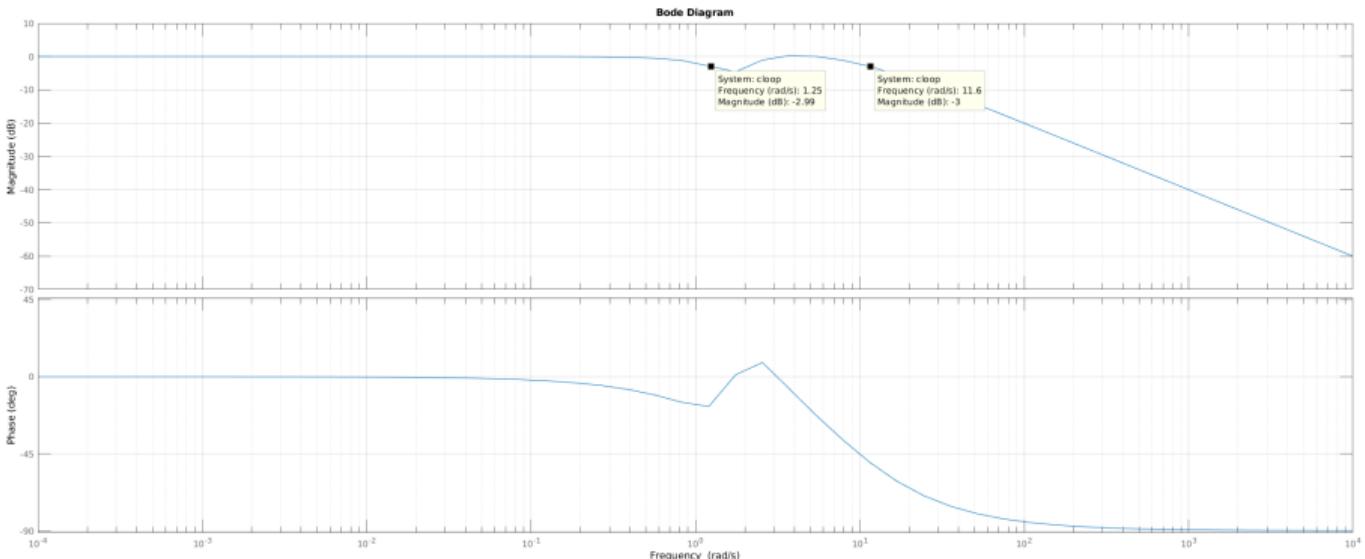
onde  $K_p = 10$ ,  $K_d = 10$  e  $K_i = 30$ .

- O diagrama de Bode do sistema de controle em malha fechada proposto  $G(s)H(s)/(1 + G(s)H(s))$  é ilustrado na figura a seguir.



inicialmente a análise  
do sistema em  
malha fechada  
domínio S

## Exemplo ...



- Note que existem dois pontos no gráfico de módulo com valor  $-3\text{dB}$ :  
 $\omega_1 = 1.25\text{rad/s}$  e  $\omega_2 = 11.6\text{rad/s}$ . Obviamente devemos escolher  $\omega_c = 11.6\text{rad/s}$  pois engloba um conjunto de frequências maior.

## Exemplo ...

- Considerando  $\omega_c = 11.6 \text{ rad/s}$  então:

$$\omega_c = 2\pi f_c \Rightarrow$$

$$f_c = \frac{11.6}{2\pi} = 1.85 \text{ Hz}$$

- Logo, para que um sistema de controle digital tenha comportamento como o de um sistema de controle de tempo contínuo devemos observar:

$$f_s = \frac{1}{T} > 50f_c = 92.5 \text{ Hz}$$

## Exemplo: software de simulação - controle\_discreto.py

- O software simula um sistema de controle digital, i.e., o controlador é simulado através de equações de diferenças e a planta utiliza um método de integração numérico utilizando a função `odeint()` do pacote Scipy.
- O array de tempo para integração numérica é definido como:

```
# number of time points
n = 10001

# time points
t = np.linspace(0,10,n)
```

- Dessa forma, o array `t` tem dimensão 10001 e é definido dentro do domínio  $[0, 10\text{seg}]$ , começando em  $t[0]=0$  até  $t[10001]=10$ .
- o intervalo de tempo  $dt=t[2]-t[1]$  define o passo de integração da planta.
- É definido uma grandeza que representa quantos múltiplos de  $dt$  cabem dentro do período de amostragem do controlador discreto:

```
fs = 10 #frequencia amostragem Hz
mult = int((1/dt)/fs)
```

## Exemplo: laço principal

- o laço principal da simulação do sistema é apresentado abaixo:

```
for i in range(1,n):
    # span for next time step
    tspan = [t[i-1],t[i]]

    erro = ref-x1[i-1]
    if (contador%mult)==0:
        acum_int = acum_int + (erro+errolast)/2*(1/fs)
        F = Kp*erro + Kd*(-x1[i-1]+xlast)/(1/fs) + Ki*acum_int
        errolast = erro
        xlast = x1[i-1]

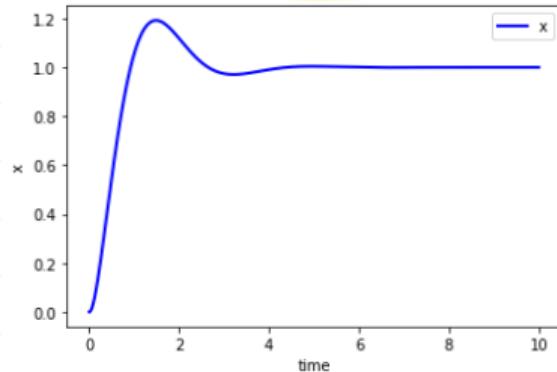
        # solve for next step

        z = odeint(model,z0,tspan,args=(F,))
        # store solution for plotting
        x1[i] = z[1][0]
        x2[i] = z[1][1]
        # next initial condition
        z0 = z[1]
        contador = contador + 1
        vetF[i]=F
```

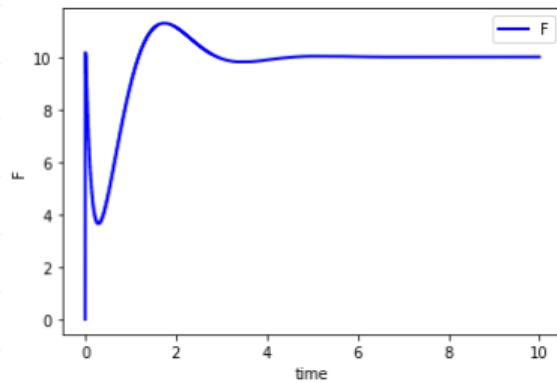
- O controlador discreto só é executado se a variável contador é múltiplo inteiro da variável mult.

## Exemplo ...

- Simulação com  $f_s = 92.5\text{Hz}$ :



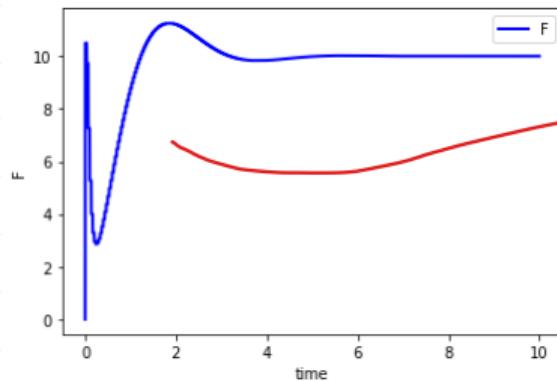
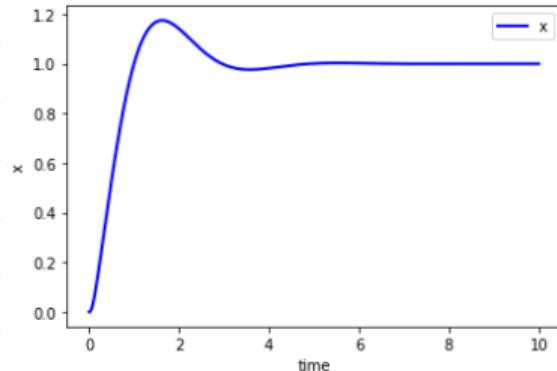
Saída do sistema  $x(t)$   
posição da massa



Sinal de atuação força  $f(t)$   
saída do controlador

## Exemplo ...

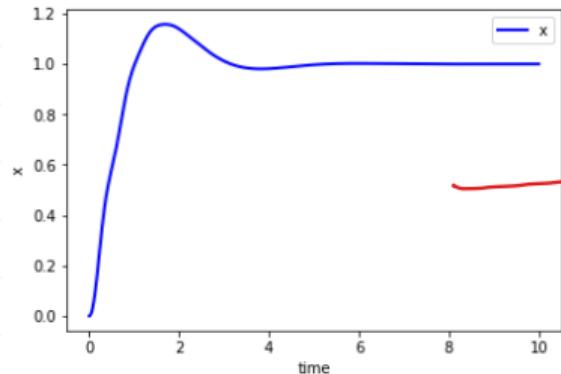
- Simulação com  $f_s = 30\text{Hz}$ :



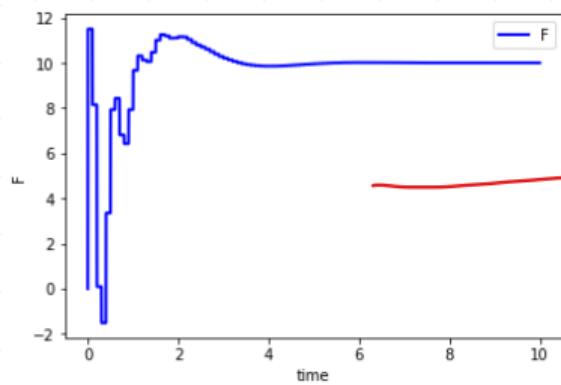
→ pode ser observado  
aspectos de discretização  
no tempo

## Exemplo ...

- Simulação com  $f_s = 10\text{Hz}$ :



→ alteração do formato de  $x(t)$   
quando comparado aos dois  
casos anteriores



→ discretização  
bastante perceptível

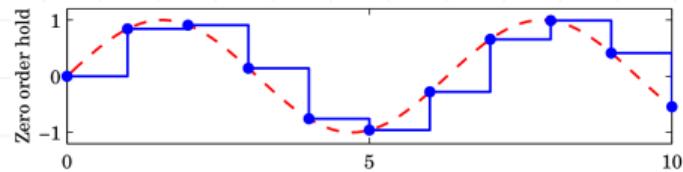
## **Reconstrução de sinais**

---

# Reconstrução de sinais

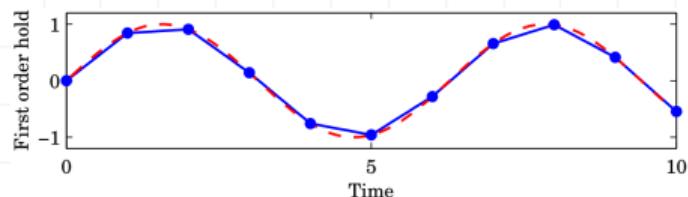
- Zero order hold:

$$f(t) = f(kT), \quad kT \leq t < kT + T$$



- First order hold:

$$f(t) = f(kT) + \frac{t - kT}{h} (f(kT + T) - f(kT)), \quad kT \leq t < kT + T$$



- Reconstrutor ideal, Reconstrutor de Shannon:

reconstrução perfeita

## Reconstrução de sinais

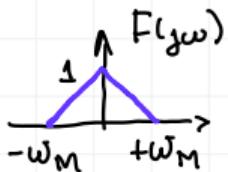
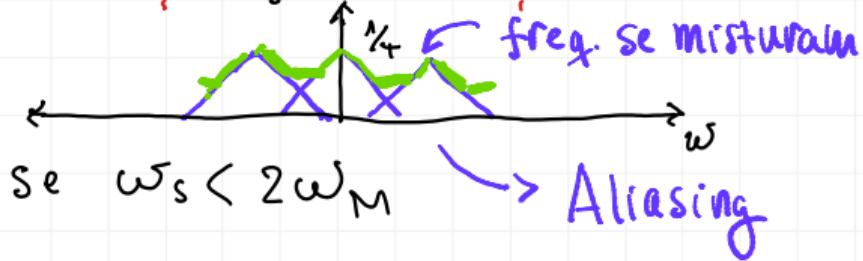
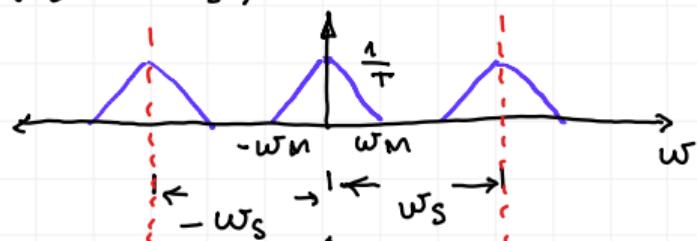
### Amostragem de sinais



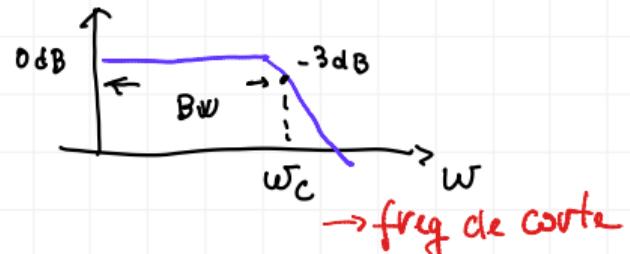
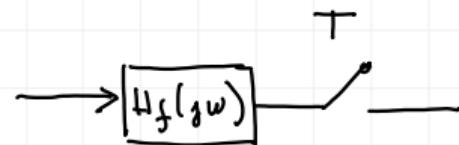
$f(t)$  sinal de banda limitada  $\omega_M$

amostras  $f(kT)$   $k=0,1,2,\dots$

Se  $\omega_s > 2\omega_M$



### Filtros Anti-aliasing

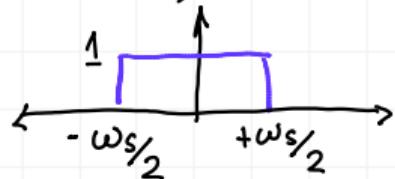


### Filtro Passa-baixas

Analógico p/ garantir que  
sinais acima de  $\omega_c$   
não tenham potência  
significativa

## Reconstrução de sinais

Filtro ideal, Reconstructor de Shannon



$$H(j\omega) = \begin{cases} 1 & -\omega_s/2 < \omega < +\omega_s/2 \\ 0 & \text{fora do intervalo acima} \end{cases}$$

Transformada de Fourier

Inversa

$$h(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} H(j\omega) e^{j\omega t} d\omega$$

multiplicamos por T

$$h(t) = \frac{1}{2\pi} T \int_{-\omega_s/2}^{+\omega_s/2} e^{j\omega t} d\omega$$

$$\begin{aligned} h(t) &= \int_{-\omega_s/2}^{+\omega_s/2} e^{j\omega t} d\omega \\ &= \frac{T}{2\pi j t} \left( e^{\frac{1}{2}j\omega_s t} - e^{-\frac{1}{2}j\omega_s t} \right) \end{aligned}$$

utilizando a fórmula de Euler:

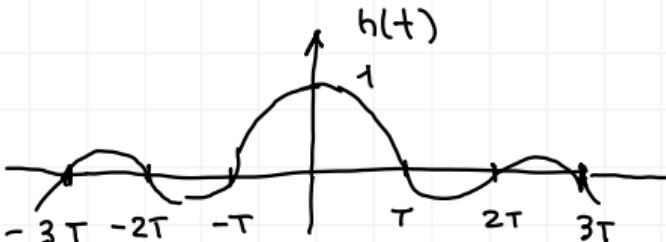
$$h(t) = \frac{\sin\left(\frac{\omega_s}{2} \cdot t\right)}{\frac{\omega_s}{2} \cdot t}$$

*Qdo discretizado  
o espectro fica  
dividido por T*

*"função sinc(x)"*  
. função simétrica e não singular  
. para  $t=0$  o valor da função  
é definida pelo limite à  
esquerda e à direita

## Reconstrução de sinais

$$h(t) = \frac{\sin\left(\frac{\omega_s}{2} \cdot t\right)}{\frac{\omega_s}{2} \cdot t}$$



Reconstrução: pontos amostrados

$$f_r(t) = \sum_{k=-\infty}^{+\infty} f(kT) \frac{\sin\left(\frac{\omega_s}{2}(t-kT)\right)}{\frac{\omega_s}{2}(t-kT)}$$

Observações:

- função não causal: depende de infinitos pontos a frente e a trás
- inviável para implementação em tempo real

- Na prática é possível considerar apenas alguns pontos já que a função sinc( $\tau$ ) decresce rapidamente conforme se distancia da origem.

## Reconstrução de sinais

## Reconstrução de sinais