

# Laboratório de Controle de Sistemas

**Profa. Grace S. Deaecto**

Faculdade de Engenharia Mecânica / UNICAMP  
13083-860, Campinas, SP, Brasil.  
[grace@fem.unicamp.br](mailto:grace@fem.unicamp.br)

Primeiro Semestre de 2023

## 1 Experimento 3

- Objetivo
- Conceitos fundamentais
- Pré-roteiro
- Roteiro

## Experimento 3

Controle de plantas eletrônicas utilizando um controlador PID digital





# Regime transitório

- Entretanto, na maioria das aplicações práticas, não basta somente garantir estabilidade. O sistema deve ter um tempo de estabilização adequado e não apresentar picos muito elevados. Neste sentido, adotando a **aproximação de polos dominantes**, podemos escrever  $F(s)$  como

$$F(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} + Q(s)$$

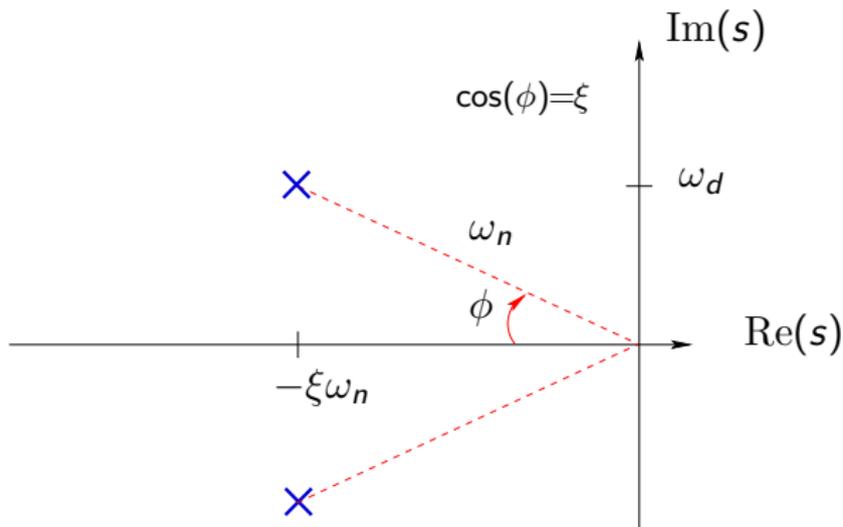
sendo que os dois polos mais próximos do eixo imaginário, chamados polos dominantes, são para  $(0 < \xi < 1)$  iguais a

$$s = -\xi\omega_n \pm j\omega_d, \quad \omega_d = \omega_n \sqrt{1 - \xi^2}$$

e os polos de  $Q(s)$  são tais que  $\text{Re}(s) \ll -\xi\omega_n$ , de forma que sua influência, no domínio do tempo, rapidamente desaparece.

# Regime transitório

- O parâmetro  $\xi$  é chamado de **coeficiente de amortecimento**,  $\omega_n$  é a **frequência natural não amortecida** e  $\omega_d$  é a **frequência natural amortecida**.



# Regime transitório

- Para a entrada degrau unitário  $r(t) = 1, t \geq 0$  a transformada de Laplace inversa fornece

$$y(t) \approx 1 - e^{-\xi\omega_n t} \left( \cos \omega_d t + \frac{\xi\omega_n}{\omega_d} \text{sen}(\omega_d t) \right)$$

- Seu **valor de pico máximo** é

$$\max_{t \geq 0} y(t) = 1 + \underbrace{e^{-\xi\pi/\sqrt{1-\xi^2}}}_{\psi(\xi)}$$



**A sobrelevação  $\Psi(\xi)$  é uma função decrescente de  $\xi$  !**

# Regime transitório

- Por outro lado, tendo em vista que  $|\varepsilon(t)|$  tem uma envoltória exponencial, o tempo necessário para que o sinal  $y(t)$  permaneça dentro desta envoltória, ou seja, o erro satisfaça  $|\varepsilon(t)| < \epsilon \approx 0$  para todo  $t \geq t_\epsilon$  é dado aproximadamente por

$$t_\epsilon = -\frac{\ln(\epsilon)}{\xi\omega_n}$$

e é chamado **tempo de estabilização**.



**O tempo de estabilização depende da distância entre os polos dominantes e o eixo imaginário !**

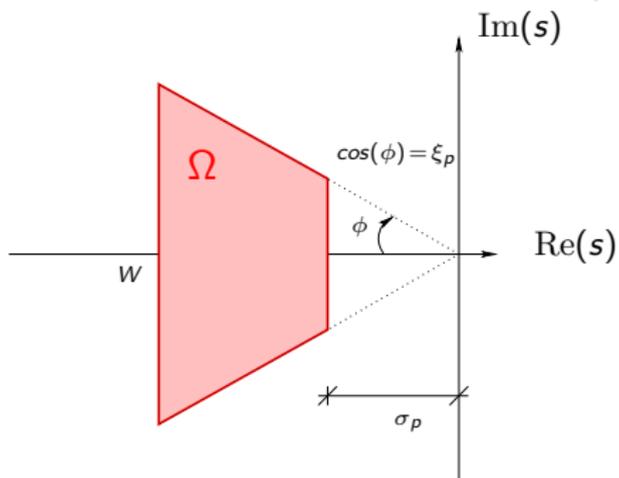
# Regime transitório

- De acordo com a frase anterior, um possível raciocínio é projetar  $C(s)$  para obter o menor tempo de estabilização possível fazendo  $\omega_n$  muito grande.
- Infelizmente, este raciocínio não pode ser adotado. **Existe um limite para  $\omega_n$** . Este parâmetro é aproximadamente igual à largura de faixa do sistema e, se muito grande, permite que ruídos de alta frequência influenciem na sua resposta.



**Deve haver um equilíbrio entre tempo de estabilização mínimo e rejeição de ruídos**

- Vamos agora determinar a **região do plano complexo  $\Omega$**  onde os polos dominantes devem estar localizados para satisfazer os requisitos de projeto. Note que :
  - Sobrelevação menor do que um valor  $\Psi(\xi_p)$  significa considerar polos com coeficiente de amortecimento  $\xi > \xi_p$ .
  - Tempo de estabilização menor do que um valor  $t_p$  significa considerar polos com  $\xi\omega_n > -\ln(\epsilon)/t_p = \sigma_p$ .
  - Para eliminar a influência de ruídos de alta frequência  $\omega_n < W$ .



## Regime permanente

- Além destes critérios é importante que em um sistema assintoticamente estável a saída  $y(t)$  siga o sinal de referência  $r(t)$  após o regime transitório. Em outras palavras, é importante que  $\lim_{t \rightarrow \infty} \varepsilon(t)$  seja nulo ou tenha um valor aceitável. A função de transferência do erro  $\varepsilon(t)$  é a seguinte

$$\frac{\hat{\varepsilon}(s)}{\hat{r}(s)} = \frac{1}{1 + C(s)G(s)}$$

Vamos adotar as seguintes entradas para o sinal de referência

$$\hat{r}(s) = \frac{1}{s^k}, \quad k = 1, 2, \dots$$

de forma que para  $k = 1$  a entrada é um degrau,  $k = 2$  uma rampa unitária e assim sucessivamente.

# Regime permanente

- Dizemos que um sistema dinâmico a tempo contínuo é do tipo  $k$  se ele possui  $k > 0$  polos na origem.
- Podemos calcular o valor do erro em regime permanente utilizando o teorema do valor final. Desta forma, temos

$$\lim_{t \rightarrow \infty} \varepsilon(t) = \lim_{s \rightarrow 0} \left( \frac{s}{1 + C(s)G(s)} \right) \left( \frac{1}{s^k} \right)$$

- A equação acima deixa claro que o sistema em malha fechada com realimentação unitária possui **erro em regime permanente nulo para uma entrada  $\hat{r}(s) = 1/s^k$ ,  $k = 1, 2, \dots$ , se e somente se a função  $C(s)G(s)$  for no mínimo do tipo  $k$ .**

## Regime permanente

- Assim, se  $C(s)G(s)$  tem **um polo na origem**, então o sistema em malha fechada, sempre que assintoticamente estável, possui **erro de regime nulo para a entrada degrau**.
- Da mesma maneira, se a função  $C(s)G(s)$  **tem dois polos na origem** então o mesmo raciocínio é válido para **entrada rampa** e assim sucessivamente.

É claro que desejamos satisfazer os requisitos de projeto, no que se refere ao desempenho em regime transitório e em regime permanente, utilizando o **controlador mais simples possível**. Entretanto, a medida que as exigências de projeto aumentam, o controlador tende a ser mais sofisticado.

# Projeto de controladores

- O controlador mais simples é o controlador proporcional

$$C(s) = \kappa > 0$$

- O mais utilizado em aplicações práticas é o controlador PID (Proporcional-Integral-Derivativo)

$$C(s) = \kappa_p + \frac{\kappa_i}{s} + \kappa_d s$$

com  $\kappa_p \geq 0$ ,  $\kappa_i \geq 0$  e  $\kappa_d \geq 0$ .

Note que o controlador PID é não realizável e, portanto, para fins de implementação consideramos

$$C(s) = \kappa_p + \frac{\kappa_i}{s} + \kappa_d \frac{s}{\tau s + 1}$$

para  $\tau > 0$  arbitrariamente pequeno.

# Projeto de controladores

- Um método empírico muito utilizado em situações práticas para sintonizar os parâmetros de um PID

$$C(s) = \kappa_p \left( 1 + \frac{1}{T_i s} + T_d \frac{s}{\tau s + 1} \right)$$

com  $T_i = \kappa_p / \kappa_i$  e  $T_d = \kappa_d / \kappa_p$  é o método de **Ziegler Nichols**.

- Ele consiste em determinar com a equação característica  $1 + \kappa G(s) = 0$  o valor de  $\kappa = \kappa_{osc} > 0$  (no limiar de estabilidade) de forma que o sistema passe a oscilar com período de oscilação  $T_{osc} = 2\pi / \omega_{osc}$ .

- Os coeficientes dos controladores são dados na tabela seguinte

Controlador	$\kappa_p$	$T_i$	$T_d$
P	$0.50\kappa_{osc}$		
PI	$0.45\kappa_{osc}$	$T_{osc}/1.2$	
PID	$0.60\kappa_{osc}$	$T_{osc}/2.0$	$T_{osc}/8.0$

- Observação** : Uma maneira de obter um resultado possivelmente melhor é determinando  $T_i$  e  $T_d$  através da tabela de Ziegler Nichols e  $\kappa_p$  pelo método do lugar das raízes.

# Pré-roteiro

- Antes de começar o projeto é importante lembrar que **o esforço de controle  $u(t)$  deve estar entre os limites de tensão de  $\pm 10$  [volts]**. Considere neste experimento :
  - A utilização do comando “**sisotool**” do Matlab para realizar o projeto via lugar das raízes.
  - A função de transferência da planta  $G(s)$  já identificada no Experimento 2.
  - Um sinal de onda quadrada de 0.05 [Hz] e amplitude variando de 0 a 1 [volt] na entrada de referência  $r(t)$ , que simula degraus repetidos.
  - Aproximação de polos dominantes para a obtenção dos critérios de desempenho no regime transitório.
  - A envoltória  $\epsilon = 0.02$  o que implica  $-\ln(\epsilon = 0.02) \approx 4$  importante para calcular o tempo de estabilização.
  - **Para o projeto dos controladores PID, utilize  $\tau = 0.01$ .**

# Pré-roteiro

## Projeto do controlador proporcional

- Utilizando o método do lugar das raízes, projete um controlador proporcional  $C(s) = \kappa > 0$  de forma a obter tempo de estabilização mínimo. Para este controlador, determine :
  - Tempo de estabilização
  - Coeficiente de amortecimento e sobrelevação.
  - Erro de regime para entrada degrau.
  - Erro de regime para entrada rampa.

# Pré-roteiro

- Utilizando o Matlab obtenha  $y(t)$  e  $u(t)$  considerando, primeiramente, que a entrada de referência  $r(t)$  é a onda quadrada de 0.05 [Hz], variando de 0 a 1 [volt] de amplitude, como especificado anteriormente.
- Posteriormente, considere a entrada  $r_{ramp}(t)$  gerada por uma onda triangular de 0.025 [Hz] e amplitude  $\pm 1$  [volt] de forma a simular rampas repetidas na entrada de referência, e obtenha os mesmos sinais.





# Pré-roteiro

## Projeto do controlador PID - Sisotool

- Projete um controlador PID utilizando o “sisotool” do Matlab que forneça um desempenho melhor do que aquele obtido utilizando o método de Ziegler Nichols (tente obter um desempenho próximo daquele fornecido pelo controlador proporcional no que se refere ao tempo de estabilização).
- Para este controlador, determine : tempo de estabilização, coeficiente de amortecimento, sobrelevação e erros de regime para entrada degrau e rampa, bem como as mesmas simulações descritas no projeto anterior.
- Compare os projetos realizados no que se refere aos requisitos de desempenho no regime transitório e permanente.

# Roteiro

## Materiais :

- Planta eletrônica.
- Placa DAQ NI-USB 6001
- Cabo de rede

## Procedimento :

- As conexões entre a planta e a placa de aquisição são idênticas às realizadas no Experimento 2.
- O programa Labview pode se basear naquele desenvolvido no Experimento 1, acrescentando o bloco do controlador e levando em conta que o sistema deve respeitar o esquema de malha fechada apresentado na figura da página 4.

# Roteiro

- Gere o sinal de referência  $r(t)$  (onda quadrada de 1 [volt] e 0.05 [Hz]).
- Obtenha o erro  $\varepsilon = r(t) - y(t)$  e aplique-o na entrada do controlador  $C(s)$ .
- Implemente o controlador fora do loop de execução. Para a sua implementação, utilize o bloco “CD Construct PID Model” disponível na biblioteca de controle do LabVIEW. Coloque sua saída no bloco de conversão de contínuo para discreto “CD Convert Continuous to Discrete.vi” utilizando um período de amostragem de  $T = 0.002$  [s]. Em seguida, dentro do loop, conecte o resultado em “CD Discrete Transfer Function.vi” e dentro do bloco altere a opção “Parameter Source” para “Terminal”. A saída deste bloco é  $u(t)$  que deve ser aplicada à planta através de um bloco DAQ.

# Roteiro

- No bloco “CD Convert Continuous to Discrete.vi” coloque as entradas de de controle  $k_p$ ,  $k_i$ ,  $k_d$  e a constante de tempo de alta frequência  $T_F$ .
- Acrescente um bloco de saturação “In Range and Coerce Function” para que a saída do controlador não ultrapasse os limites permitidos  $\pm 10$  [volts].
- Utilize um outro bloco DAQ para a aquisição de  $y(t)$  e realmente a saída para gerar o erro  $\varepsilon(t)$ .
- Realize o experimento para os controladores proporcional e PIDs utilizando os parâmetros calculados no pré-laboratório.
- Para o sinal de onda quadrada e, para todos os controladores, armazene os sinais de referência  $r(t)$ , o esforço de controle  $u(t)$  e a saída  $y(t)$  para posterior análise no Matlab.



- A figura a seguir mostra um esquema da tela de execução

