	Projeto #1	Turma: C213 L1/L2	Data: 24/09/24
	C213 – Sistemas Embarcados		
Docente: Samuel Baraldi Mafra		PED: Igor Gonçalves de Souza	
Objetivo: Apresentar a proposta para o primeiro Projeto Prático da disciplina.			
Integrante 1:			
Integrante 2:			
Integrante 3:			

Projeto Prático de Sistemas Embarcados

Identificação de Processos e Sintonia de Controladores PID

Controle de Processos Remotos com Atraso de Propagação

1 Controle de Processos - contextos e importância

O controle de processos é uma disciplina fundamental na engenharia, com aplicações desde a indústria química até sistemas de transporte e automação predial. A capacidade de monitorar e ajustar variáveis de um sistema garante eficiência operacional e otimização de resultados.

A modelagem destes processos define representações matemáticas e permite uma compreensão mais profunda do seu comportamento, facilitando a análise de suas dinâmicas. Modelos precisos são essenciais para a simulação e projeto de controladores, garantindo a possibilidade de diferentes estratégias de controle antes de implementá-las na prática.

2 Modelagem Matemática de Processos

O modelo matemático de um processo é uma Função de Transferência no domínio de Laplace que relaciona os sinais de saída $V_o(s)$ ou $PV(s)$ e de entrada $V_i(s)$ ou $SP(s)$ e descreve características do sistema prevendo parâmetros como estabilidade e erro.

A resposta de um processo a um determinado estímulo de entrada pode ser dividida em **resposta transitória** - estado inicial, na qual o sinal varia rapidamente no tempo e tende à um estado final chamado de **resposta estacionária** ou **permanente**, regime no qual o comportamento da saída varia pouco com o tempo. A amplitude em regime permanente - Valor Final, é calculada pelo Teorema do Valor Final como em 1, em que $H(s)$ representa o modelo matemático do sistema.

$$VF = \lim_{s \rightarrow 0} s \cdot SP(s) \cdot H(s) \quad (1)$$

Nos instantes iniciais de resposta, a curva apresenta variações mais rápidas que tendem a amortecer no tempo. O **Tempo de Subida** - t_r , infere o intervalo necessário para que a amplitude varie de 10% a 90% do Valor Final, enquanto o **Tempo de Acomodação** - t_s , é o intervalo para o regime estacionário.

3 Resposta Típica de Sistemas de Primeira Ordem

Circuitos RC, sistemas térmicos, hidráulicos ou semelhantes representam um Sistema de Primeira Ordem, com Função de Transferência na forma canônica dada pela Equação 2, em

que k é o **Ganho Estático** em malha aberta e τ , a **Constante de Tempo**.

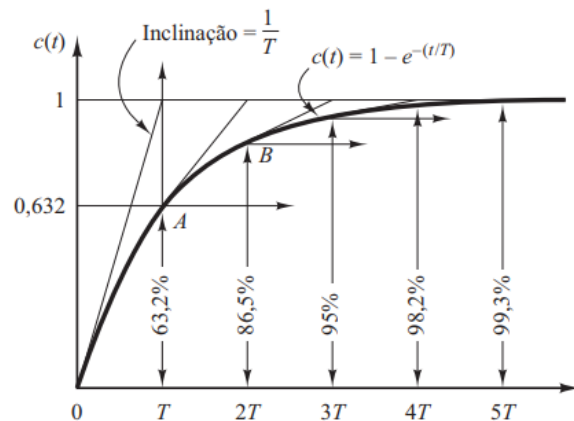
$$H(s) = \frac{V_o(s)}{V_i(s)} = \frac{PV(s)}{SP(s)} = \frac{k}{\tau s + 1} \quad (2)$$

A classe `tf` do MATLAB e o método `control.tf()` do Python modelam uma Função de Transferência:

```
>> sys = tf(k, [tau 1]) ou
      sys = control.tf(k, [tau, 1]).
```

O gráfico da Figura 1 apresenta o comportamento típico de um Sistema de Primeira Ordem para uma entrada do tipo Degrau. Após uma constante de tempo, em $T = \tau$, a resposta do sistema varia de 0 a 63.21% do Valor Final. Após duas constantes de tempo, em $T = 2\tau$, atinge 86.47%. Segundo o **Critério dos 2%**, um Sistema de Controle atinge o regime permanente em 98% do seu Valor Final. Assim, para um Sistema de Primeira Ordem, $t_s \cong 4\tau$.

Figura 1: Resposta Típica de Sistemas de Primeira Ordem.



4 Resposta Típica de Sistemas de Segunda Ordem

Circuitos RLC e a grande maioria dos Algoritmos de Controle representam um Sistema de Segunda Ordem, com Função de Transferência na forma canônica dada pela Equação 3, em que ω_n é a **Frequência de Oscilação** e ξ , o **Fator de Amortecimento**.

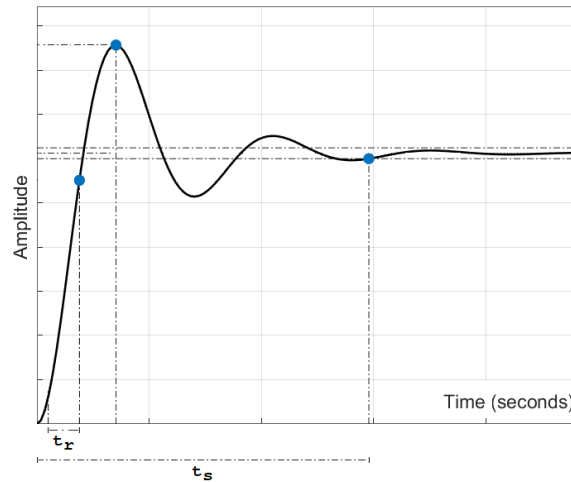
$$H(s) = \frac{V_o(s)}{V_i(s)} = \frac{PV(s)}{SP(s)} = \frac{\omega_n^2}{s^2 + 2\xi\omega_n \cdot s + \omega_n^2} \quad (3)$$

A classe `tf` do MATLAB e o método `control.tf()` do Python modelam uma Função de Transferência:

```
>> sys = tf(wn^2, [1 2*xi*wn wn^2]) ou
      sys = control.tf(wn^2, [1, 2*xi*wn, wn^2]).
```

O gráfico da Figura 2 apresenta o comportamento típico de um Sistema de Segunda Ordem para uma entrada do tipo Degrau. Como característica, a curva de resposta apresenta oscilações em torno do Valor Final, com amplitudes reduzidas no tempo em razão do amortecimento do sistema. As oscilações são caracterizadas por uma Amplitude de Pico - a maior amplitude da curva de resposta, definida como **Máximo Pico**. Essa amplitude representa uma porcentagem do Valor Final da resposta por um fator de **overshoot**.

Figura 2: Resposta Típica de Sistemas de Segunda Ordem.



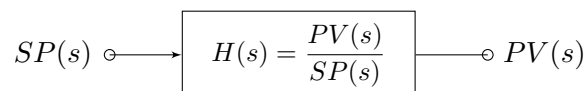
5 Malhas de Controle

As malhas de operação para sistemas de controle referem-se à maneira como os sinais são interpretados e regulados pelo processo. Existem diferentes tipos de malhas utilizadas, a depender de parâmetros como estabilidade, precisão e velocidade de resposta.

5.1 Malha Aberta

O esquema da Figura 3 opera em Malha Aberta, modelo no qual o sinal de saída não exerce ação de controle, ou seja, a amplitude da saída não é medida nem comparada com a entrada. A referência para o sistema é o próprio $SP(s)$.

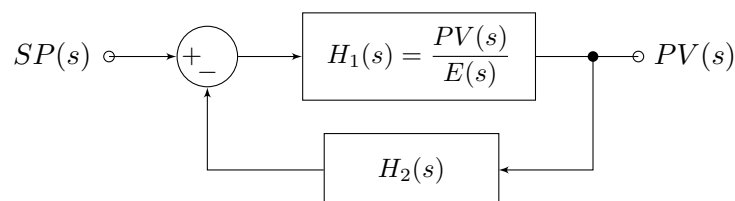
Figura 3: Sistema de Controle em Malha Aberta.



5.2 Malha Fechada

Um Sistema de Controle em Malha Fechada estabelece uma relação de comparação entre as amplitudes de saída e entrada, utilizando essa comparação como meio de controle. O esquema da Figura 4 opera com sistema em Malha Fechada, em que $H_1(s)$ é a **Malha de Controle** e $H_2(s)$, a de **Realimentação**.

Figura 4: Sistema de Controle em Malha Fechada.



Neste caso, a referência para o sistema é o erro do processo, com $E(s) = SP(s) - PV(s)$. A Função de Transferência equivalente em 4 relaciona os sinais de entrada e saída

em Malha Fechada, em que o sinal do denominador é oposto à realimentação de $PV(s)$.

$$H(s) = \frac{PV(s)}{SP(s)} = \frac{H_1(s)}{1 \pm H_1(s) \cdot H_2(s)} \quad (4)$$

A função `feedback()` do MATLAB e o método `control.feedback()` do Python modelam o sistema com operação em malha fechada, com as funções de transferência das malhas de controle e realimentação. Os exemplos a seguir consideram a realimentação padrão.

```
>> sys_fechada = feedback(sys, 1) ou
    sys_fechada = control.feedback(sys).
```

6 Atraso de Transporte

O Atraso de Transporte, ou tempo morto, corresponde ao tempo θ em que o sistema demora para responder ao estímulo de entrada. Em sistemas dinâmicos, o Atraso de Transporte pode afetar o processo com possíveis instabilidades ou dificuldades na sintonia do controlador.

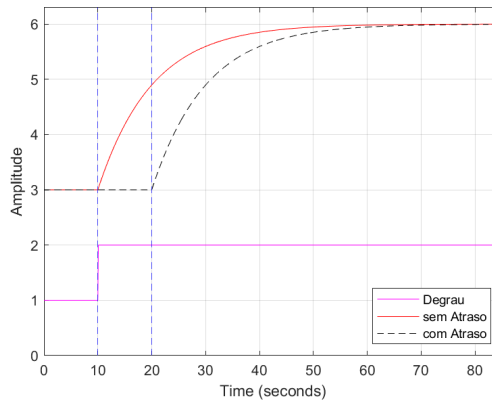
Esse atraso pode ser encontrado em sistemas térmicos, hidráulicos e pneumáticos causado por fatores como inércia ou a distância física entre a origem e o ponto de controle, por exemplo, no tempo para transportar fluido em uma tubulação. Com o desenvolvimento de sistemas de controle via redes, sensores e controladores passam a se conectar remotamente seguindo protocolos de comunicação sujeitos a atrasos de processamento e propagação de sinais.

Sendo θ o tempo de atraso, um Sistema de Primeira Ordem com Atraso de Transporte (FOPDT) tem como modelo a Função de Transferência em 5, em que a parcela irracional $e^{-\theta s}$ representa o atraso.

$$H(s) = \frac{V_o(s)}{V_i(s)} = \frac{PV(s)}{SP(s)} = \frac{k}{\tau s + 1} \cdot e^{-\theta s} \quad (5)$$

A Figura 5 compara um sistema com atraso para $\theta = 10$. O Atraso de Transporte desloca a curva de resposta ao longo do tempo, sem modificação na dinâmica de amplitudes. Para a dinâmica temporal $t_0 = t + \theta$.

Figura 5: Apresentação de Sistemas com Atraso de Transporte.



No MATLAB, a propriedade `'InputDelay'` da classe `tf` define Sistemas com Atraso de Transporte: `>> sys = tf(k, [\tau 1], 'InputDelay', theta).`

6.1 Aproximação de Padé

A aproximação de Padé é uma técnica matemática utilizada para aproximar funções irracionais por meio de frações racionais, ou seja, como a razão de dois polinômios num e den. O recurso de Padé é de grande importância para funções que necessitam de modelos racionais e não suportam modelos de sistemas com Atraso de Transporte, ou em ambientes de desenvolvimento que não admitem a propriedade '**InputDelay**'.

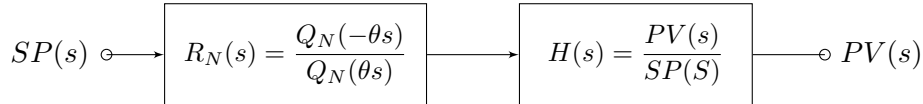
Seja $R_N(s)$ a aproximação racional por Padé da irracional $F(s)$. A aproximação $R_N(s)$ garante a igualdade das funções em torno do ponto $s = 0$ para o valor inicial e para as derivadas de N primeiras ordens, em que N é a ordem de $R_N(s)$ e o grau dos polinômios num e den. A aproximação para $R_N(s)$ é resultante da razão entre dois polinômios de aproximação $Q_N(s)$, dados como em 6 e 7.

$$Q_N(F(s)) = \sum_{j=0}^N \frac{(n+j)!}{j!(n-j)!} \cdot F(s)^{n-j} \quad (6)$$

$$R_N(s) = \frac{Q_N(-F(s))}{Q_N(F(s))} \quad (7)$$

A função `pade(T, N)` do MATLAB e o método `control.pade(T, N)` do Python realizam a aproximação de Padé de N -ésima ordem para um atraso $T = \theta$, retornando os polinômios num e den. Como resultante, a malha de controle do modelo *FOPDT* é um sistema puro em série com a aproximação racional da parcela $e^{-\theta s}$, demonstrado em 6. A função `series(sys1, sys2)` do MATLAB e o método `control.series(sys1, sys2)` do Python estabelecem a conexão entre as Funções de Transferência.

Figura 6: Modelo resultante do Sistema com Aproximação de Padé.



7 Controladores PID - proposta e sintonia

O Algoritmo de Controle PID é uma das abordagens mais comuns na área de controle, combinando três ações principais para regular a saída de um sistema em relação a uma referência desejada: Proporcional, Integral e Derivativa. A sintonia de um PID é a etapa de cálculo dos valores dos parâmetros da Função de Transferência do Controlador. Os métodos de sintonia são definidos por um critério de desempenho desejado para a malha, como um menor *overshoot* ou menor tempo de subida.

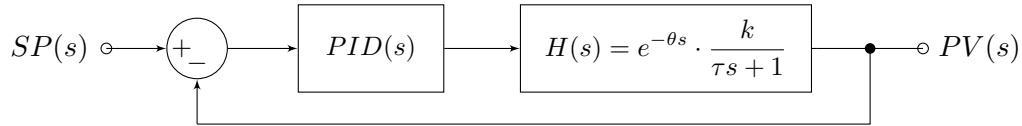
A parametrização se dá pela resposta ao degrau de um Sistema de Primeira Ordem com Atraso de Transporte em Malha Aberta, com Função de Transferência modelada por 5, em que a razão θ/τ entre o Atraso de Transporte e a Constante de Tempo é conhecida como **Fator de Incontrolabilidade**. Quanto maior esse fator, mais difícil se torna controlar o processo.

A sintonia define os valores de K_p , T_i e T_d , em que K_p é **Ganho Proporcional**, T_i o **Tempo de Integral** e T_d , **Tempo de Derivada**.

$$PID(s) = K_p \left(1 + \frac{1}{T_i \cdot s} + T_d \cdot s \right) = \frac{K_p T_d \cdot s^2 + K_p \cdot s + K_p / T_i}{s}$$

O Controlador PID opera na Malha de Controle do Sistema em Malha Fechada, alterando a ação de controle gerada pelo erro atuante no sistema.

Figura 7: Sistema com Controle PID.



7.1 Método de Ziegler Nichols Malha Aberta - Curva de Reação

O Método Ziegler Nichols Malha Aberta é definido a partir das características da resposta transitória do processo em Malha Aberta para uma entrada do tipo degrau. O Sistema em Malha Fechada apresenta uma taxa de decaimento de pelo menos $C/A = 1/4$. As regras de sintonia são como a Tabela 1.

Tabela 1: Parâmetros do PID para a Sintonia pela ZN Curva de Reação.

Controlador	K_p	T_i	T_d
PID	$\frac{1.2\tau}{k \cdot \theta}$	2θ	$\frac{\theta}{2}$

7.2 Método do Modelo Interno - IMC

Neste método, o Controlador possui um Modelo Interno do processo que utiliza a Função de Transferência da planta para determinar o ajuste dos parâmetros do PID, propondo a sintonia por um parâmetro ajustável λ , o qual determina a velocidade da resposta do sistema. Valores menores de λ deixam o sistema mais rápido e com melhor desempenho, porém mais sensível às perturbações, com $t_s = 4\lambda$. As regras de sintonia são como a Tabela 2 [1].

Tabela 2: Parâmetros do PID para a Sintonia Método do Modelo Interno.

Controlador	K_p	T_i	T_d	Critério Desempenho
PID	$\frac{2\tau + \theta}{k \cdot (2\lambda + \theta)}$	$\tau + \frac{\theta}{2}$	$\frac{\tau \cdot \theta}{2\tau + \theta}$	$\frac{\lambda}{\theta} > 0.8$

7.3 Método CHR

O Método CHR estabelece dois critérios de desempenho para o sistema considerando o problema servo (mudança de valor do *Setpoint*): resposta mais rápida sem *overshoot* ou resposta mais rápida com 20% de *overshoot*. As regras de sintonia são como as Tabelas 3 e 4, respectivamente.

Tabela 3: Parâmetros do PID para a Sintonia sem sobrevalor.

Controle	K_p	T_i	T_d
PID	$\frac{0.6\tau}{k \cdot \theta}$	τ	$\frac{\theta}{2}$

Tabela 4: Parâmetros do PID para a Sintonia com 20% de sobrevalor.

Controle	K_p	T_i	T_d
PID	$\frac{0.95\tau}{k \cdot \theta}$	1.357τ	0.473θ

7.4 Método Cohen e Coon

O Método de Cohen e Coon foi proposto para Sintonia de Sistemas com Atraso de Transporte mais elevados, quando a planta possui Fator de Incontrolabilidade maior que 0,3. Assim como o Método de Ziegler Nichols, o modelo é resultado do ensaio em Malha Aberta com um degrau aplicado na entrada. As regras de sintonia são como a Tabela 5.

Tabela 5: Parâmetros do PID para a Sintonia Método de Cohen e Coon.

Controlador	K_p	T_i	T_d
PID	$\frac{\tau}{k \cdot \theta} \left(\frac{16\tau + 3\theta}{12\tau} \right)$	$\theta \cdot \frac{32 + 6\theta/\tau}{13 + 8\theta/\tau}$	$\frac{4\theta}{11 + 2\theta/\tau}$

7.5 Método ITAE

O ITAE é um Índice de Desempenho baseado na minimização do erro, resultado da Integral da multiplicação do Erro Absoluto pelo tempo (*ITAE – Integral of Time multiplied by Absolute Error*). Um Sistema com Controle ITAE reduz grandes erros iniciais, assim como erros posteriores ocorridos na resposta transitória, tornando as oscilações bem amortecidas e reduzindo consideravelmente o *overshoot* [2].

As regras para Sintonia ITAE apresentadas na Tabela 6 são regidas por seis constantes, com $A = 0.965$, $B = -0.85$, $C = 0.796$, $D = -0.147$, $E = 0.308$ e $F = 0.929$.

Tabela 6: Parâmetros do PID para a Sintonia ITAE [2].

Controlador	K_p	T_i	T_d
PID	$\frac{A}{k} \cdot \left(\frac{\theta}{\tau} \right)^B$	$\frac{\tau}{C + D \cdot (\theta/\tau)}$	$\tau \cdot E \cdot \left(\frac{\theta}{\tau} \right)^F$

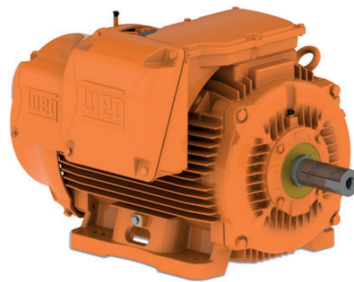
8 Sistemas de Controle para Velocidade de Motor com Acesso Remoto

Um processo industrial é caracterizado por uma série de operações com o objetivo de fabricar algum bem ou material, com segurança aplicada ao monitoramento e controle das variáveis do processo. Entre elas, o controle preciso da velocidade e rotação de motores elétricos é fundamental em várias aplicações industriais, como em sistemas de transporte e robótica. A capacidade de ajustar e manter a velocidade do motor em níveis desejados não só melhora a eficiência energética, mas também garante a segurança do sistema operacional.

A implementação de sistemas de controle remoto se torna especialmente relevante em ambientes industriais, em que a monitorização pode evitar comandos em áreas de maior risco. No entanto, um dos principais desafios enfrentados por esses sistemas é o atraso de transporte referente ao tempo de propagação e processamento das mensagens enviadas entre o controlador e o motor, que pode causar oscilações indesejadas e perda de estabilidade, tornando a necessidade do controle ainda mais crítica.

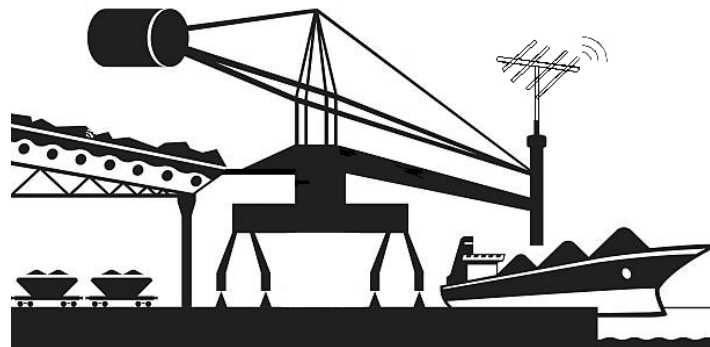
Um ambiente típico de mineração é notório por ser um dos mais severos ambientes de trabalho, com condições ambientais críticas, contaminação sólida e processos rigorosos. Neste trabalho, um motor da linha Mining da WEG é utilizado para movimentação de materiais em uma mineradora, em que o controle de velocidade é necessário para otimizar o consumo de energia a depender do peso e dimensão dos materiais transportados [3].

Figura 8: Motor *W22 Mining*, da WEG [3].



Nesta proposta de aplicação, como mostrado em 9, o motor para movimentação dos materiais é controlado por uma central remota, em que os dados são enviados por uma Interface Homem-Máquina, interpretados pela central e geram impulsos no motor. O tempo de deslocamento dos comandos à central gera um atraso de propagação, inferido como atraso de transporte para o processo de controle.

Figura 9: Transporte de materiais com *W22 Mining* no controle da esteira.



Projeto

Tendo em vista a necessidade do controle aplicado à velocidade de motores elétricos, as etapas seguintes visam projetar um Sistema de Controle PID para o *W22 Mining* no cenário do transporte de materiais.

1. Identifique o conjunto de dados para seu grupo, nomeado por '*Dataset_GrupoX*'.
2. Escolha o Método de Identificação da Planta - Smith ou Sundareshan, e determine os

valores de k , τ e θ para levantar a Função de Transferência do modelo de acordo com a resposta típica. **Justifique a escolha do método e do modelo.**

3. Compare a resposta original em relação à estimada e verifique se a aproximação foi satisfatória. Se necessário, realize o ajuste fino dos parâmetros, expondo o reflexo das alterações na resposta do sistema.
4. Plote as respostas do Sistema em Malha Aberta e Fechada e comente sobre as diferenças nos Tempos de subida e de acomodação e no erro do processo;
5. Sintonize um Controlador PID de acordo com os métodos especificados e verifique o comportamento do sistema controlado. Para a Sintonia IMC, a escolha de λ é livre de acordo com o desempenho. **Justifique o valor escolhido para o parâmetro.**

	Métodos	Conclusão
Grupo 1	Técnicas 1 e 4	(a) Tempo de resposta
Grupo 2	Técnicas 1 e 5	(b) <i>Overshoot</i>
Grupo 3	Técnicas 2 e 3	(a) Tempo de resposta
Grupo 4	Técnicas 2 e 4	(b) <i>Overshoot</i>
Grupo 5	Técnicas 2 e 6	(a) Tempo de resposta
Grupo 6	Técnicas 3 e 6	(b) <i>Overshoot</i>
Grupo 7	Técnicas 4 e 5	(a) Tempo de resposta
Grupo 8	Técnicas 4 e 6	(b) <i>Overshoot</i>

Tabela 7: Parâmetros do PID para os métodos de Sintonia especificados.

Técnica	Método	K_p	T_i	T_d
1	Ziegler Nichols Malha Aberta			
2	IMC			
3	CHR sem Sobrevalor			
4	CHR com Sobrevalor			
5	Cohen e Coon			
6	ITAE			

6. De acordo com a conclusão especificada, explique, entre os sistemas controlados, qual apresenta: **(a)** tempo de resposta mais rápido ou **(b)** menores índices de *overshoot*.
7. Crie uma interface que permita ao usuário definir os parâmetros do PID e do *Setpoint*. Opcionalmente, personalize a interface com outras métricas que julgar interessantes (possibilidade de nota extra).

Considerações

- O projeto pode ser feito em Python ou em MATLAB R2019a ou superior;
- A entrega do trabalho deve ser feita via GitHub. O repositório deve conter um *readme* detalhando a proposta da aplicação e todos os arquivos referência para o trabalho;
- A disposição do diretório é livre, importando o *dataset* e organizando pastas e arquivos de forma adequada;
- Trabalhe com `help` ou `lookfor` do MATLAB como auxílio em caso de dúvidas na utilização de funções e comandos;
- Todos os gráficos devem ser salvos para garantia da resolução.

Referências

- [1] RIVERA, Daniel E., MORARI, Manfred, SKOGESTAD, Sigurd. **Internal Model Control: PID Controller Design**. Industrial & Engineering Chemistry Process Design and Development, California Institute of Technology, California, 1985, 252-265p.
- [2] Metrópole Digital. **Sintonia de Controladores PID: Método da Integral do Erro**. URL: <<https://materialpublic.imd.ufrn.br/curso/disciplina/1/63/6/4>>.
- [3] WEG S.A. - HEADQUARTERS. https://static.weg.net/medias/downloadcenter/hae/h9c/50025494_portuguese_web.pdf. Acesso: 17-09-24.