UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

ENG04035 - SISTEMAS DE CONTROLE I
LABORATÓRIO

Prof. João Manoel Gomes da Silva Jr
Prof. Romeu Reginatto
Prof. Tristão Garcia

Setembro de 2003

Instruções

Para utilizar os diversos recursos computacionais do laboratório utilize os seguintes dados:

- Usuário/*Username*/em Login : **extensao**
- Senha/*Password*: **eletro99x**
- Domínio/*Domain* : **delet**

Página da disciplina: `http://www.eletro.ufrgs.br/~romeu/eng04035`

Laboratório 1

Introdução ao Matlab

1.1 Básico

O matlab possui uma janela de comando onde podem ser declaradas variáveis e executados comandos sobre estas variáveis. Também possui janelas gráficas, onde são apresentados resultados em forma gráfica. Certas operações podem ser executadas diretamente nas janelas gráficas (zoom, print, etc). Porém, a janela de comando é ainda a forma mais direta de utilizar o matlab.

O símbolo >> representa o *prompt* na janela de comando. A seguir são relacionados alguns comandos básicos:

1. >> help - lista todos os *toolboxes*
2. >> help <nome-toolbox> - lista todas as funções do *toolbox*
3. >> help <nome-funcao> - ajuda sobre a função nome-funcao
4. pwd - informa diretório de trabalho corrente
5. cd <novo-caminho>> - troca diretório corrente para <novo-caminho>
6. *Avançadas.* save, load, diary, edit.

1.2 Declarando e usando variáveis

O matlab lida com vários tipos de dados, dentre os quais vamos estudar: números complexos, vetores, matrizes e sistemas lineares.

1.2.1 Esclares: Números reais e complexos

1. i - variável pré-definida (i=$\sqrt{-1}$)
2. >> a=2 - declara a variável a valendo 2
3. >> b=a+i*3 - declara o número complexo $a + j3$
4. >> j=sqrt(-1) - declarando j=i
5. Operações com números complexos: +, -, *, /, ^ (potenciação)
6. >>abs(x) - retorna o módulo do número complexo x
7. >>angle(x) - retorna a fase no número complexo x
8. >>real(x) - retorna a parte real do número complexo x
9. >>imag(x) - retorna a parte imaginária do número complexo x

1.2.2 Vetores

1. `>>v=[1 2 4]` ou `>>v=[1, 2, 4]` - declara o vetor linha $v = [1, 2, 4]$
2. `>>w=[1; 3; 5]` ou `w=[1 3 5]'` - declara o vetor coluna $w = \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix}$
3. `>>ones(10,1)` - vetor coluna (10 linhas) com todos elementos 1
4. `>>zeros(1,5)` - vetor linha (5 colunas) com todos elementos 0
5. `>>v1=[-1:0.1:3]` - declara vetor linha com primeiro elemento igual a -1, último elemento igual a 3, e incrementos de 0.1 entre os elementos sucessivos
6. Operações com vetores: `+`, `-`, `*` (respeitadas as dimensões)
7. *Avançadas.* Funções `linspace` e `logspace`.

1.2.3 Matrizes

Segue a mesma sintaxe dos vetores.

1. `>>m1=[1 2 4; 3 5 6]` ou `>>v=[1, 2, 4; 3, 5, 6]` - declara a matriz $2{\times}3$ $m = \begin{bmatrix} 1 & 2 & 4 \\ 3 & 5 & 6 \end{bmatrix}$
2. `>>ones(10,2)` - matriz 10×2 com todos elementos 1
3. `>>zeros(3,5)` - matriz 3×5 com todos elementos 0
4. `>>diag([1 3 5])` - matriz diagonal 3×3 com os elementos 1, 3 e 5 na diagonal
5. `>>eye(5)` - matriz identidade 5×5
6. Operações com matrizes: `+`, `-`, `*` (respeitadas as dimensões)
7. `>>inv(A)` - inversa da matriz A
8. `>>A'` - transposta da matriz A
9. `>>det(A)` - determinante de A
10. `>>eig(A)` - autovalores da matriz A
11. *Avançadas.* Função `rand`

1.2.4 Operações elemento a elemento em matrizes e vetores

Nas operações elemento a elemento, vetores e matrizes são considerados como conjuntos de elementos. As operações são realizadas entre os elementos que tem o mesmo índice nos operandos (vetores ou matrizes). Exemplo.

```
>>v1=[1 2 3; 4 5 6];
>>v2=[-1 -3 -4; 2  4 1];
>>v1.*v2
  ans = [-1 -6 -12; 8 20 6]
```

1. `sin(v)` - retorna o seno de cada elemento de `v`. Isto vale em geral para funções pré-definidas como `sin`, `cos`, `log10`, `sqrt`, etc.
2. `v+1` - soma 1 a cada elemento de `v`.

3. `v1.^2` - eleva ao quadrado cada elemento de `v1`.
4. `.*`, `./` - produto e divisão elemento a elemento.
5. `sin(v).*cos(v)` - retorna o produto do seno e cosseno de cada elemento de `v`.

1.2.5 Polinômios

Um polinômio é representado por um vetor linha que contém como elementos os coeficientes do polinômio ordenados da potência mais elevada ao termo independente.

1. `>>p1=[1 2 5]` - representa o polinômio $p1 = s^2 + 2s + 5$
2. `>>p2=poly([-2 -5-i -5+i])` - cria um polinômio `p2` que possui raízes -2, $-5-j$ e $-5+j$
3. Operações com polinômios: `+`, `-` efetuadas como se fossem vetores (respeitar dimensões)
4. `>>conv(p1,p2)` - produto dos polinômios `p1` e `p2`
5. `>>deconv(p2,p1)` - divisão $p2/p1$
6. `>>roots(p1)` - raízes do polinômio `p1` (formato de vetor coluna)
7. `>>poly(A)` - polinômio característico da matriz `A`

1.3 Funções

A partir do janela de comando podem ser executadas funções pré-definidas ou funções que fazem parte de *toolboxes*.

Passando argumentos para funções. Qualquer tipo de dado pode ser passado como argumento de funções. Os parâmetros de funções são passados colocados entre parênteses (logo após o nome da função) e são separados por vírgulas. Exemplos:

1. `>>roots(p1)` ou `>>roots([1 2 5])` - um único argumento passado
2. `>>conv(p1,p2)` - dois argumentos passados

Muitas funções do matlab efetuam operações diferentes de acordo com o tipo e quantidade de argumentos que são fornecidos. Exemplo:

1. `>>poly([-2 -5-i -5+i])` - retorna um polinômio `p2` que possui raízes -2, $-5-j$ e $-5+j$
2. `>>poly(A)` - retorna o polinômio característico da matriz `A`

Obtendo os argumentos de retorno. Funções do matlab podem retornar um ou mais parâmetros (argumentos de retorno), podendo estes serem tipos de dados distintos. Os valores retornados podem ser atribuídos a variáveis ou ignorados. A variável `ans` sempre armazena o último resultado ignorado (não atribuído a nenhuma variável). Quando mais de um parâmetro é retornado, as variáveis que os armazenam são colocadas entre colchetes e separadas por vírgula. A atribuição é pela ordem: primeiro parâmetro atribuído a primeira variável, segundo à segunda e assim por diante.

1. `>>p2=poly([-2 -5-i -5+i])` - função `poly` retorna um único parâmetro. Resultado é atribuído a `p2`.
2. `>>[q,r]=deconv(p2,p1)` - função `deconv` retorna dois parâmetros (quociente e resto) que são atribuidos às variáveis `q` e `r`, respectivamente.
3. `>>q=deconv(p2,p1)` - função `deconv` retorna dois parâmetros (quociente e resto), mas somente o quociente (primeiro valor retornado) é guardado.

1.4 Sistemas Lineares

Sistemas lineares podem ser descritos no matlab através de funções de transferência e de equações de estado. Diversas funções pré-definidas podem ser utilizadas para manipular estes sistemas. Estas funções fazem parte do *control toolbox*. Para ver a lista das funções, veja `>>help control`.

1. `>>g1=tf([2 4],[1 5 6])` - declara uma **função de transferência** através do numerador e denominador, no caso, $g_1(s) = \frac{2s+4}{s^2+5s+6}$.
2. `>>zpk(g1)` - obtém a representação de `g1` em no formato zeros-polos-ganho.
3. `>>zero(g1)` - obtém os zeros de `g1`.
4. `>>pole(g1)` - obtém os pólos de `g1`.
5. `>>pzmap(g1)` - obtém os pólos e os zeros de `g1`.
6. `>>g2=zpk([],[0 -5 -20],100)` - declara uma função de transferência através lista de zeros, lista de pólos e valor do ganho, no caso, $g_2(s) = \frac{100}{s(s+5)(s+20)}$.
7. `>>s1=ss(A,B,C,D)` - declara um sistema linear no **espaço de estados** através das matrizes A, B, C e D. Teste com $A = [0,\ 1; -5,\ -2]$, $B = [0;\ 1]$, $C = [1,\ 3]$ e $D = 0$.
8. `>>tf(s1)` - obtém a função de transferência $Y(s)/U(s)$ de $s1$ a partir de sua representação de estados.
9. `set(g3,'InputDelay',2)` - adiciona **atraso de transporte** de 2 segundos na função de transferência `g3`.
10. `g2.InputDelay=2` - adiciona **atraso de transporte** de 2 segundos na função de transferência `g2`.
11. Operações com sistemas lineares:
    - `-, +` - associação em paralelo.
    - `*, /` - produto e divisão de funções de transferência.
    - `g3=0.5*g2` - Cria a função $g_3(s) = 10g_2(s) = \frac{50}{s(s+5)(s+20)}$.
    - `>>t=feedback(g1,g2)` - retorna a função de transferência de malha fechada resultante da associação em **realimentação negativa**: `g1` no caminho direto e `g2` no caminho de realimentação.
    - `>>t=feedback(g1,1)` - caso particular da **realimentação negativa unitária**: `g1` no caminho direto.
12. Forma alternativa. Teste esta forma de declarar uma função de transferência.

```
>>s=zpk([0],[],1)
>>g3=(s^2+3)/(s*(s+2)*(s+5))
```

1.5 Análise de sistemas realimentados

As funções `bode`, `nyquist`, `rlocus` e `step` admitem diversos parâmetros e modificam seu comportamento com base nos argumentos passados e na existência ou não de argumentos de retorno.

- **Gráfica.** Sem argumentos de retorno, elas apresentam o resultado de forma gráfica.

- **Numérica.** Com argumentos de retorno, os resultados são apresentados de forma numérica e armazenados nas variáveis correspondentes. Esta alternativa é fundamental para obter os resultados numéricos necessários em projetos de controladores.
- **Faixa.** As faixas de freqüência e/ou tempo é determinada automaticamente pelas funções, bem como o número de pontos que são avaliados. Você pode sobrepor este comportamento passando como argumentos os intervalos ou valores exatos em que quer obter seu resultado.

Para os exercícios abaixo, utilize uma ou mais das seguintes funções de transferência:

$$G(s) = \frac{s+5}{(s+2)(s+10)}, \quad G(s) = \frac{20}{s^2+10s+50}, \quad G(s) = \frac{200}{(s+5)(s^2+10s+50)}$$

1. `>>bode(g)` - constrói o **diagrama de Bode** de `g`, escolhendo automaticamente a faixa de freqüência e o número de pontos, e apresenta o resultado de forma gráfica.
2. `>>[m,f]=bode(g)` - retorna os dados de forma numérica: `m` é o módulo (linear) e `f` é a fase em graus. Use `squeeze(m)` e `squeeze(z)` para obtém os resultados na forma de vetores.
3. `>>[m,f]=bode(g,w)` (Veja também `>>help bode`). Retorna módulo e fase de $G(s)$ para a freqüência específicada em `w`. Com isto você pode determinar as freqüências de cruzamento de ganho e de fase na tentativa e erro.
4. `>>nyquist(g)` - constrói o **diagrama de Nyquist** de `g`, escolhendo automaticamente a faixa de freqüência e o número de pontos, e apresenta o resultado de forma gráfica. *Utilize os recursos de zoom para visualizar melhor os resultados.*
5. Analogamente à função `bode`, teste `>>[re,im]=nyquist(g)` e `[re,im]=nyquyst(g,w)` para obter a parte real e imaginária de `g`.
6. `>>rlocus(g)` - constrói o **lugar das raízes** da equação característica correspondente a `g`, escolhendo automaticamente a faixa de ganho e o número de pontos, e apresenta o resultado de forma gráfica.
7. `rlocfind(g)` - retorna o ganho de ajuste para um ponto escolhido sobre o lugar das raízes.
8. `>>step(g)` - constrói a **resposta ao degrau** de `g`, escolhendo automaticamente o intervalo de tempo e o número de pontos, e apresenta o resultado de forma gráfica.
9. Veja no `>>help step` como obter os dados numéricos da resposta. Com base nisto, determine os valores do tempo de acomodação t_s e da máxima sobre-elevação (máximo *overshoot*).

Avançadas. Outras funções úteis `evalfr`, `freqresp`, `squeeze`, `linsim`, `c2d`, `d2c`, `rltool`, `simulink`.

1.6 Adicionais: Gráficos

A interface gráfica do Matlab permite manipular diversar janelas gráficas simultanemante. Cada janela gráfica pode apresentar diversos gráficos. Cada gráfico pode apresentar diversas curvas em cores ou traçados distintos.

1. `>>x=[0:0.1:7]; y=sin(x);` - cria dois vetores.
2. `>>plot(y)` - traça o gráfico dos valores de `y` em função dos índices do vetor `y`. O traçado é feito sobre a janela gráfica corrente. Se não houver nenhuma, uma é criada.
3. `>>plot(x,y)` - traça `y` em função de `x`.

4. `>>figure(n)` - cada janela gráfica é identificada por um número inteiro maior ou igual a 1. A função `figure` permite tornar corrente uma janela gráfica já existente ou criá-la, caso ainda não exista.

5. `>>xlabel('Tempo (s)')` - adiciona *label* para o eixo das coordenadas.

6. `>>ylabel('seno(t)')` - adiciona *label* para o eixo das absissas.

7. `>>title('Grafico da funcao seno.')` - adiciona título para o gráfico.

8. `>>axis([xmin xmax ymin ymax])` - seleção manual das escalas dos eixos x e y (usar valores numéricos). Note que a função tem um único argumento, sendo este um vetor.

9. `>> grid` - adiciona grade no gráfico.

10. `>> hold on/off` - a função `plot` normalmente elimina as curvas anteriores existentes no gráfico. Usando `hold on`, as curvas são só adicionadas, sem eliminar as anteriores. Usando `hold off` volta-se ao modo normal. Note que a função `hold` se aplica ao gráfico corrente apenas.

11. `>>subplot(312)` - divide a janela gráfica corrente em uma matriz de 3 por 1 gráficos e seleciona o gráfico de número 2, contados da esquerda para direita e de cima para baixo. Teste o seguinte:

```
>>subplot(311)
>>plot(t,sin(t))
>>subplot(312)
>>plot(t,cos(t))
>>subplot(313)
>>plot(t,exp(-0.5t))
```

12. *Avançadas.* `semilogx`, `semilogy`, `loglog`, `mesh`, `surface`.

Laboratório 2

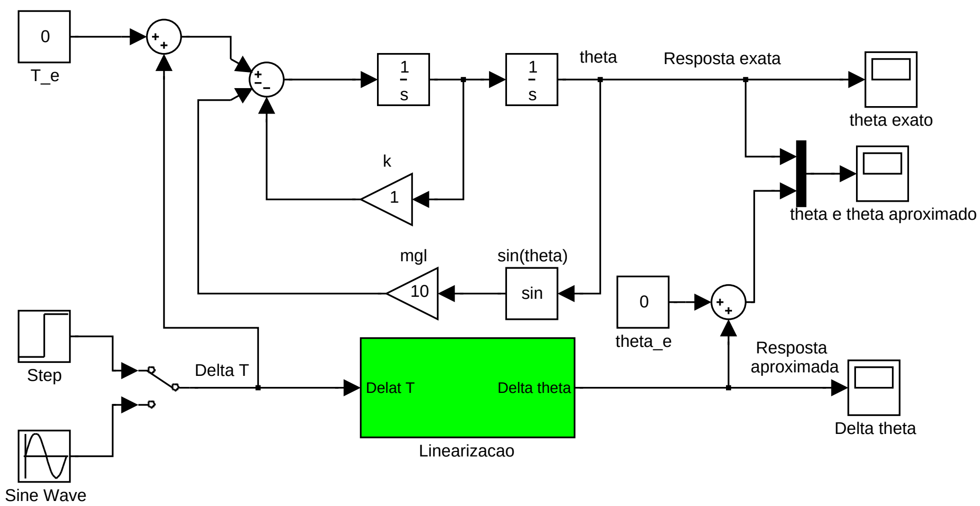
Linearização de sistemas não-lineares

Objetivos:

- Trabalhar a noção de ponto de equilíbrio de um sistema dinâmico.
- Realizar a linearização de um modelo não linear no entorno de um ponto de equilíbrio e obter a função de transferência válida para pequenas variações em torno do ponto de equilíbrio
- Visualizar e comparar a resposta do sistema não-linear com a do modelo linearizado e concluir sob as condições de validade do modelo linear.

Atividades:

1. Implemente no Matlab-Simulink o diagrama em blocos abaixo correspondente ao modelo de um pêndulo. Observe que o diagrama em blocos efetivamente implementa a equação diferencial
$$ml^2\ddot{\theta} + k\dot{\theta} + mgl\ sin(\theta) = T \tag{2.1}$$
com $m = 1Kg$, $g = 10m^2/s$, $l = 1m$, $k = 1N.m.s$.



2. Considerando o equilíbrio em $\theta(t) = \theta_e = 0$
   (a) Determine quanto será o torque de equilíbrio, T_e, para manter o pêndulo nesta posição.

(b) Faça a linearização do modelo em torno deste ponto de equilíbrio e obtenha uma função de transferência que relacione pequenas variações de ângulo ($\Delta\theta$) com pequenas variações sobre o torque de equilíbrio (ΔT).

(c) Programe a função de transferência obtida no bloco linearização Simule e compare a resposta do sistema não-linear com a do modelo linearizado aplicando uma variação de torque do tipo salto de amplitude $\Delta T = 0.5,\ 5,\ 10$. Para cada um destes valores calcule o valor do novo ponto de equilíbrio atingido a partir do modelo não-linear e linear, respectivamente. Justifique os resultados obtidos.

(d) Compare a resposta do sistema não-linear com a do modelo linearizado aplicando uma variação de torque do tipo senoidal de amplitude $\Delta T = 0.5,\ 5,\ 10$ e freqüência angular de $2rad/s$.

3. Repita o item anterior considerando o equilíbrio em $\theta(t) = \theta_e = \pi/4$

Laboratório 3

Comportamento Transitório de Sistemas Lineares e Dominância

Objetivos:

- Definição de sistemas e simulação temporal a partir do workspace do MATLAB.
- Relacionar os parâmetros da função de transferência e a resposta temporal do sistema que ela representa.
- Relacionar a posição dos pólos e zeros da função de transferência e a resposta temporal do sistema que ela representa.

Funções a serem utilizadas: `tf, zpk, step, hold, pzmap, set(G,'InputDelay',L)`

3.1 Sistemas de primeira e segunda ordem

1. Sistemas de Primeira Ordem. Simule a resposta ao degrau do sistema dado pela função de transferência $G(s) = \frac{K}{\tau s+1}$ para as seguintes condições: (i) $K = 2$ e $\tau = 1/10,\ 1,\ 10$; (ii) $\tau = 1$ e $K = 1/10,\ 1,\ 10$. Compare o valor de regime permante atingido em cada caso e o tempo que o sistema leva para chegar a este valor. Conclua sobre o efeito de K e τ sobre estes indicadores. Obtenha a resposta analítica (a partir de frações parciais e da Transformada de Laplace inversa) do sistema e justifique o obtido a partir da observação dos termos exponenciais.

2. Sistemas de Segunda Ordem. Simule a resposta ao degrau do sistema dado pela função de transferência $G(s) = \frac{K\omega_n^2}{s^2+2\xi\omega_n s+\omega_n^2}$ para as seguintes condições: (i) $\omega_n = 2$, $K = 8$, $\xi = 1,\ 0.7,\ 0.3$; (ii) idem para $\omega_n = 10$. Conclua sobre o efeito de ξ e ω_n sobre as seguintes características da resposta ao degrau: (i) tempo que o sistema leva para chegar próximo ao regime permanente; (ii) valor máximo atingido pela resposta relativo ao valor de regime permanente; (iii) existência e freqüência da oscilação amortecida da resposta.

3.2 Casos especiais

1. Sistemas Integradores. Simule a resposta ao degrau do sistema dado pela função de transferência $G(s) = \frac{K}{s(\tau s+1)}$. Interprete o ocorrido do ponto de vista físico, relacionando com o comportamento dos sistemas de primeira e segunda ordem.

2. Sistemas Instáveis. Simule a resposta ao degrau do sistema dado pela função de transferência $G(s) = \frac{2}{(s-1)(s+1)}$. Obtenha a resposta analítica (a partir de frações parciais e da Transformada de Laplace inversa) do sistema e justifique o obtido a partir da observação dos termos exponenciais.

3. Sistemas com Atraso de Transporte. Simule a resposta ao degrau do sistema dado pela função de transferência $G(s) = \frac{2e^{-Ls}}{5s+1}$ para $L = 1/10,\ 1,\ 10$. Observe o efeito do atraso na resposta do sistema.

3.3 Dominância de singularidades

Considere o sistema linear de segunda ordem com pólos complexos conjugados em $-5 \pm j5$ dado por

$$G(s) = \frac{50}{s^2 + 10s + 50}.$$

aqui considerado como *sistema padrão*. Simule a resposta ao degrau unitário e tome-a como referencial comparativo para os seguintes itens.

Influência da adição de um pólo. Considere o sistema

$$G(s) = \frac{50p}{(s^2 + 10s + 50)(s + p)}$$

para os casos: $p = 25$, $p = 5$, $p = 1$. Simule os 3 casos e compare a resposta ao degrau obtida em cada caso com a resposta do sistema padrão e associe as diferenças obtidas com a posição do pólo p frente aos pólos do sistema padrão.

Influência da adição de zeros. Considere o sistema

$$G(s) = \frac{(50/z)\,(s + z)}{s^2 + 10s + 50}$$

para os casos: $z = 25$, $z = 5$, $z = 1$.

- Simule os 3 casos e compare a resposta ao degrau obtida em cada caso com a resposta do sistema padrão e associe as diferenças obtidas com a posição do pólo p frente aos pólos do sistema padrão.
- Para avaliar a influência de zeros no semiplano direito, repita o item anterior para $z = -1$, $z = -5$, $z = -25$.

Laboratório 4

Identificação de parâmetros de funções de transferência

Objetivos:

- Realização de ensaio ao salto para a identificação dos parâmetros de uma função de transferência.
- Validação de um modelo obtido.

Atividades:

1. Explanação sobre a planta a ser ensaiada e o sistema de aquisição de dados.
   (a) Aplicativo: `Elipse Windows`.
   (b) No Elipse, carregue a aplicação `ensaio_degrau_malhaaberta`.
2. Planejamento e preparação do ensaio ao salto.
   (a) Defina o ponto de operação do sistema (no caso vamos considerar o sistema em repouso estando a temperatura ambiente e com atuação nula).
   (b) Defina a amplitude do desvio em relação ao ponto de operação a ser aplicado para o ensaio. O desvio será em forma de degrau. A atuação sobre a planta (MV) é definida em termos da potência entregue à resistência por meio de um PWM. A faixa de potência é dada em percentual, variando de 0% (potência zero) a 100% (potência máxima). Para o processo na condição de operação especificada, amplitudes entre 10% e 20% são adequadas.
   (c) Defina o período de amostragem (intervalo de tempo entre medidas sucessivas da temperatura de saída da planta). Deve ser especificado em ms no janela do supervisório. O valor mínimo é $500ms$, sendo aceitável algo próximo a $1s$.
   (d) Verificações finais. Ajuste a chave "potência de aquecimento" do kit em 100%. Cerifique-se que o ventilador está desligado.
3. Realização do Ensaio.
   (a) No supervisório, clique em `iniciar ensaio`. O valor definido para MV (Amplitude Degrau) é aplicado ao sistema e a temperatura começará a subir até chegar em um valor de regime. Isto deve levar alguns minutos e pode ser acompanhado no gráfico do supervisório e no display do controlador N1100.
   (b) Aguarde o sistema chegar ao regime permanente para ter a informação completa da resposta do sistema. Estando em regime, clique em finalizar ensaio.

(c) Os dados do ensaio são armazenados nos arquivos textos `temperatura.dat` e `potencia.dat` no diretório `c:\Elipse\matlab`. Cada arquivo contém uma tabela de dados organizada em colunas com os seguintes significados:

Arquivo	Colunas
`temperatura.dat`	Tempo, PV (Temperatura oC)
`potencia.dat`	Tempo, MV (Potencia %)

4. Análise dos Dados e Obtenção de um Modelo

(a) A análise será feita com auxílio do Matlab.

(b) No matlab carregue os arquivos criados pelo Elipse. Use o comando `cd` para mudar de diretório. Use o comando `load temperatura.dat` para carregar o arquivo `temperatura.dat`. O matlab criará uma matriz `temperatura` com a mesma estrutura de linhas e colunas do arquivo. Faça o mesmo para o arquivo `potencia.dat`.

(c) Suponha que o processo pode ser modelado, nesta região de operação, por uma função de transferência do tipo:

$$G(s) = \frac{\Delta T(s)}{\Delta P\%(s)} = \frac{Ke^{-Ls}}{\tau s + 1} \tag{4.1}$$

Observando o gráfico plotado e a matriz de dados, estime os parâmetros K, L e τ.

(d) Usando o modelo acima, simule a resposta do sistema a um degrau de mesma amplitude do ensaio. Compare o obtido com os valores experimentais. Se necessário, faça um ajuste fino dos parâmetros e re-simule até que a resposta dada pelo modelo aproxime satisfatoriamente a resposta do processo real.

Laboratório 5

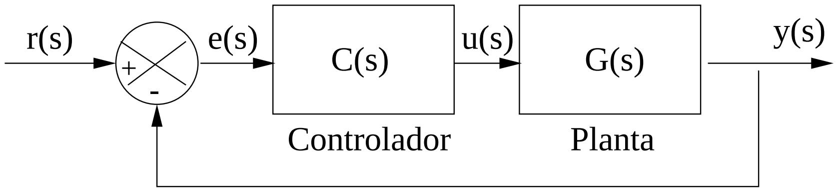
Ações básicas de controle

Objetivos:

- Visualizar e interpretar criticamente os efeitos das ações básicas de controle: on-off; proporcional; integral; derivativa.

Atividades:

1. (**Ações P, I, D**)Para o sistema abaixo com $G(s) = \frac{1}{(s+1)^3}$ e considerando uma entrada do tipo salto unitário:



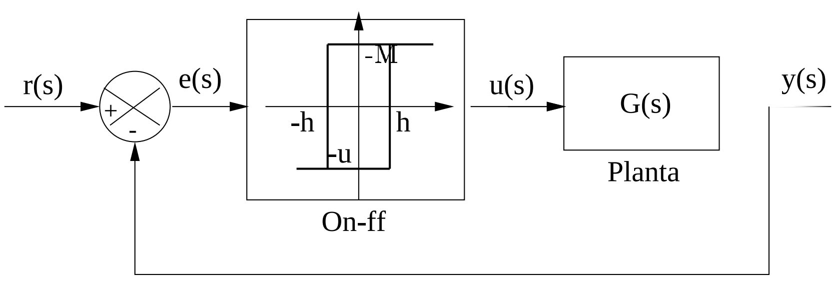
Observe sempre as variáveis $y(t)$, $u(t)$ e $e(t)$ nas simulações e procure entender o efeito de cada uma das ações de controle. Observe e justifique: os valores iniciais e finais para cada uma das ações de controle bem como a influência de cada uma delas tanto no desempenho em regime permanente quanto transitório.

a) Considere $C(s) = K$ (controlador proporcional). Simule para $K = 1, 2, 5$.

b) Considere $C(s) = K(1 + \frac{1}{T_i s})$ (controlador P.I.). Com $K = 1$, simule para $T_i = 10, 2, 1$.

c) Considere $C(s) = K(1 + \frac{1}{T_i s} + \frac{T_d s}{(\tau+1)}$ (controlador P.I.D) com $K = 1$ e $T_i = 2$. Simule para $T_d = 0.1, 0.7, 4.5$. Usar $\tau = 0.01$.

2. (**Ação liga-desliga**) Simule o sistema abaixo para uma entrada de referência do tipo salto de amplitude 0.2, considerando: $G(s) = \frac{1.5e^{-0.1s}}{(s+5)}$; $M = 1$; $h = 0$ (on-off puro). Repita para um salto de 0.5.



a) Observe $y(t)$, $u(t)$ e $e(t)$.

b) Justifique, interprete e tire conclusões sobre os sinais obtidos.

c) Repita o exercíco considerando h=0.2 (on-off com histerese).

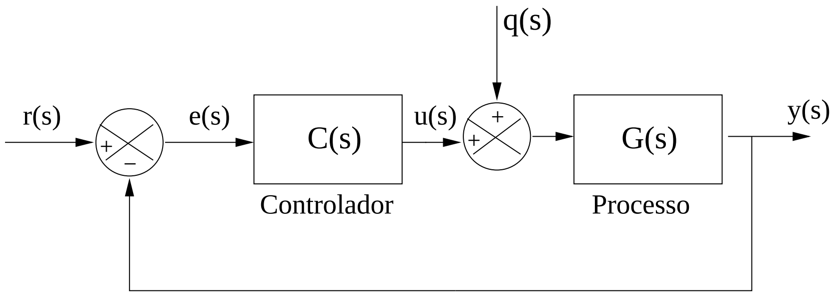
Laboratório 6

Seguimento de referência e rejeição a perturbações

Objetivo. Visualisar e analisar: a resposta de um sistema a referência e perturbações; erros em regime permanente; problemas de seguimento de referência e rejeição a perturbações.

Para este laboratório, programe no Simulink o sistema apresentado no diagrama de blocos abaixo considerando o processo

$$G(s) = \frac{2e^{-0.2s}}{(s+1)(s+3)}$$



6.1 Controle Proporcional

Considere $C(s) = K$ (controle proporcional). Através de simulação avalie os sinais $e(t)$, $u(t)$ e $y(t)$ em cada condição descrita abaixo identificando sempre **justificativas físicas e matemáticas** para os resultados obtidos.

1. Seguimento da referência. Simule o sistema para uma referência tipo degrau unitário considerando $q(t) = 0$ (sem perturbação) para os casos $K = 1$, 5, 20.

2. Rejeição de perturbações. Adicione uma perturbação tipo degrau de amplitude $1/2$ iniciando em $t = 5s$. Para os mesmos valores de ganho avalie o comportamento do sistema observando o efeito tanto da referência quanto da perturbação.

6.2 Controle Integral

Considere $C(s) = \frac{1}{T_i\, s}$. Novamente avalie os sinais $e(t)$, $u(t)$ e $y(t)$ em cada condição descrita abaixo identificando sempre **justificativas físicas e matemáticas** para os resultados obtidos.

1. Repita os mesmos experimentos realizados no caso do controle proporcional, agora para $T_i = 2$, $1/2$, $1/4$. Quais as diferenças em relação ao seguimento de referência e rejeição da perturbação? Explique.

2. Simule o sistema considerando agora uma referência do tipo rampa unitária, para os mesmos valores de T_i. Confirme os valores de erro em regime permanente medidos da simulação com os calculados a partir do modelo.

6.3 Ponto de Entrada da Perturbação

Avalie a influência do ponto de entrada da perturbação em relação a posição do integrador na malha.

Tome por base o resultado do item 1 do experimento com controle integral, isto é, com

$$C(s) = \frac{1}{T_i s}, \qquad G(s) = \frac{2e^{-0.2s}}{(s+1)(s+3)}$$

Considere novamente uma referência constante (degrau) e a perturbação também constante (degrau) aplicada em $t = 5s$ e com amplitude $1/2$.

1. Simule e compare os resultados em relação ao seguimento de referência e rejeição de perturbações, para $K = 1/2$, 2, 4 agora com

$$C(s) = K, \qquad G(s) = \frac{2e^{-0.2s}}{s(s+1)(s+3)}$$

   O que há de diferente e por quê?

Laboratório 7

Sintonia de controladores PID por Ziegler& Nichols

Objetivos:

- Aprender a operar e programar controladores PID tipo "single-loop".
- Realização de ensaio ao salto para a sintonia de controladores PID.
- Observação da operação do controlador PID e seus efeitos no processo.

7.1 O controlador PID N1100

O controlador Novus N1100 implementa um controlador PID industrial através de um microprocessador. O controlador implementa a seguinte lei de controle:

$$MV = \frac{100}{Pb}\left(1 + \frac{Ir}{60}\int_0^t e(\tau)\,d\tau + Dt\,\frac{d\,e(t)}{dt}\right)$$

onde

- Pb é a banda proporcional determinada pela relação

  $$Pb = \frac{100}{K}$$

  onde K é o ganho proporcional. O controlador espera o ganho proporcional com as seguintes unidades:

  $$[K] = \frac{\%P}{\%T}$$

  onde $\%P$ é percentual de potência (unidade da atuação do controlador - MV) e $\%T$ é percentual de temperatura (temperatura expressa em percentual de fundo de escala do sensor). A planta utiliza como sensor de temperatura um termopar do tipo K, cuja faixa de medida de temperatura vai de -90^oC até 1370^oC. Assim, a conversão para $\%T$ é determinada por

  $$\text{Temperatura em } [\%T] = \frac{100}{1460}\text{Temperatura em } [^oC]$$

- Ir é o ganho integral, expresso em $\frac{1}{\text{minutos}}$.
- Dt é o tempo derivativo, expresso em segundos, em valores inteiros.

Para maiores informações sobre o controlador N1100, seu funcionamento e uso, consulte o material de apoio ao laboratório ou o manual do próprio controlador disponível no laboratório.

7.2 Atividades

1. No programa Elipse carregue e coloque a rodar a aplicação supervisória `sintonia_pid_degrau.app`.
2. Realização do ensaio ao salto.
   (a) Programe o controlador para operar em modo manual (não automático ou malha aberta).
   (b) Ajuste a potência a ser aplicada na planta igual a 20%. Para isto, ajuste a potência no controlador (MV) igual a 20.
   (c) Ajuste a chave "potência de aquecimento" do kit em 100%. Certifique-se que o ventilador está desligado.
   (d) Coloque o controlador em operação. Para isto ajuste YES na tela RUN.
   (e) A temperatura começará a subir até chegar em um valor de regime. Isto deve levar alguns minutos. Estando em regime, desligue o controlador e passe a fazer a análise da curva de resposta do ensaio a partir do supervisório Elipse.
3. Com base no ensaio ao salto realizado, determine os ganhos do PID utilizando a tabela de Chien dada abaixo:

Overshoot	0%			20%		
Tipo de controlador	K	T_i	T_d	K	T_i	T_d
P	$\frac{0,3}{a}$	—	—	$\frac{0,7}{a}$	—	—
PI	$\frac{0,35}{a}$	$1,2\tau$	—	$\frac{0,6}{a}$	τ	—
PID	$\frac{0,6}{a}$	τ	$0,5L$	$\frac{0,95}{a}$	1.4τ	$0,47L$

4. Programe os ganhos determinados no item anterior e coloque o controlador para operar em modo **automático** (malha fechada). Teste as 3 configurações do controlador como P, PI e PID.
5. Observe o comportamento da variável de processo (temperatura) e da variável de controle (potência aplicada) durante o transitório e em regime permanente nos 3 casos. Conclua a respeito. Obs: Para este ensaio, considere como "set-point" (referência) uma temperatura próxima a a temperatura de regime permanente obtida durante o ensaio ao salto.

Laboratório 8

Método do lugar das raízes

Objetivos:

- Aprender a utilizar ferramentas computacionais para o traçado do lugar das raízes de polinômios;
- Compreender o traçado do lugar das raízes com auxílio de ferramentas computacionais.

Atividades:

1. As funções que o matlab implementa relacionadas ao traçado do lugar das raízes, e que serão utilizadas neste laboratório, são as seguintes:

- `rlocus(F)`. Traça o lugar das raízes do polinômio $d(s) + kn(s)$ para uma certa faixa de ganho k, escolhida automaticamente, onde F é a função racional $n(s)/d(s)$ definda como uma função de transferência (veja `tf` ou `zpk`).
- `rlocfind(F)`. Retorna o ganho `k` que corresponde à localização de raízes escolhida sobre o traçado (localizar com o *mouse* sobre a figura e marcar com o botão esquerdo).
- `roots(p)`. Retorna as raízes do polinômio `p`. Note que o polinômio $p(s) = s^2 + 2s + 3$ é definido no matlab como um vetor linha `p=[1 2 3]`.
- `evalfr(F,s1)`. Retorna o valor da função de transferência `F` no ponto complexo dado por `s1`.

Leia o *help* de cada uma destas funções e veja as possíveis variantes quanto aos argumentos de retorno e parâmetros.

Verifique. Para verificar seu entendimento desta parte, faça o traçado do lugar das raízes do polinômio, já estudado em aula, $q(s) = s(s + 10) + k$ para $k \geq 0$.

2. Considere os seguintes polinômios dependentes do parâmetro $k \geq 0$.

- $q(s) = s(s + 2)(s + 12) + 48k$
- $q(s) = s^3 + 20s^2 + 25ks + 25k$
- $q(s) = s(s + 20)(s^2 + 10s + 50) + 12k(s^2 + 14s + 98)$

Para cada um dos polinômios:

1. (Em casa.) Esboce o lugar das raízes identificando: (i) o ponto de partida das raízes; (ii) a tendência das raízes quando $k \to \infty$; (iii) o número de assíntotas e seus ângulos; (iv) os trechos de existência de lugar das raízes sobre o eixo real.

2. Traçe o lugar das raízes utilizando a função `rlocus` e compare o resultado com seu esboço.

3. Utilizando a função `rlocfind`, determine o ganho `k` para 2 localizações distintas de raízes do polinômio. Para cada uma destas localizações, verifique a consistência do resultado calculando as raízes do polinômio $q(s)$ para o valor de `k` determinado.

4. Procure compreender o traçado do lugar das raízes através das regras de construção que você já conhece e solicite o auxílio do professor em caso de dúvida.

3. Considere o polinômio $q(s) = (s + 50)(s^2 - 10s + 50) + 12k(s + z)$.

1. Utilizando o matlab (função `rlocus`), trace o lugar das raízes deste polinômio para $k \geq 0$ para os seguintes valores de z: (i) $z = 20$; (ii) $z = 1$; (iii) $z = -1$ (iv) $z = -20$.

2. Avalie as alterações no lugar das raízes que decorrem das mudanças em z. O que ocorre com o centro das assíntotas e a tendência das raízes para $k \to \infty$? Em quais casos é possível escolher raízes somente reais para o polinômio?

3. Avalie agora o lugar das raízes para (i) $z = -0.05$; (ii) $z = -0.13$; (iii) $z = -0.2$. Analise a possibilidade de ocorrência de raízes múltiplas em cada caso. Quais os ganhos que permitem escolher raízes múltiplas?

Laboratório 9

Análise de sistemas realimentados pelo método do lugar das raízes

Objetivos:

- Compreender o uso do lugar das raízes na análise de sistemas de controle realimentados.
- Utilizar ferramentas computacionais (rltool) para estudo de sistemas realimentados com o método do lugar das raízes.

Atividades:

1. **Compreensão.** Considere o controle integral ($C(s) = \frac{1}{T_i s}$) do processo $G(s) = \frac{Y(s)}{U(s)} = \frac{5}{(s+1)(s+5)}$.

- (Em casa.) Obtenha a função de transferência do sistema realimentado $T(s) = Y(s)/R(s)$ e a equação característica do mesmo na forma $q(s) = d(s) + kn(s)$, onde k é relacionado ao parâmetro T_i. Obtenha $GH_{eq}(s) = kF(s)$ para o traçado do lugar das raízes da equação característica $q(s)$.
- Utilizando a função `rlocus(F)`, trace o lugar das raízes da equação característica (pólos do sistema realimentado) em função do parâmetro T_i.
- Utilizando a função `rlocfind(F)`, determine o tempo integral necessário para ajustar um par de pólos complexos conjugados dominantes tal que resulte a resposta ao degrau com máximo overshoot próximo de 5%.
- Com este valor de T_i determinado, obtenha a função de transferência do sistema realimentado $T(s)$ utilizando a função `T=feedback(C*G,1)`. Aqui, C refere-se a função de transfência do controlador e G à função de transferência do processo (ambas definidas por meio da funções `tf` ou `zpk`). O segundo argumento na função `feedback` se refere a realimentação unitária. Veja `help feedback` para mais informações.
- Se não ficarem evidenciados os pólos de $T(s)$, obtenha os mesmos através da função `pole(T)`. Localize os pólos de $T(s)$ sobre o traçado do lugar das raízes e verifique a consistência dos resultados.
- Obtenha a resposta ao degrau do sistema realimentado utilizando a função `step(T)`. Avalie o máximo overshoot e verifique se atende a especificação inicial.

2. rltool. O matlab disponibiliza o pacote `rltool` para facilitar a análise de sistemas realimentados, especialmente através do lugar das raízes. O pacote pode ser inicializado digitando `rltool` na linha de comando do matlab. Outras formas de inicializar o `rltool`, mais convenientes, são as seguintes:

- `rltool(G)` inicializa o `rltool` com o modelo da planta (processo) definido pelo parâmetro `G`. Neste caso, o `rltool` já traça o lugar das raízes da equação característica do sistema realimentado considerando um controle proporcional do processo.

- `rltool(G,C)` inicializa o `rltool` com o processo `G` e o controlador `C`. Neste caso, o `rltool` também traça o lugar das raízes da equação característica do sistema realimentado considerando o processo `G`, controlador `C`, e considera um ganho ajustável no controlador como parâmetro para o lugar das raízes. Assim, o lugar das raízes traçado se refere as raízes da equação $q(s) = \text{numerador}\{1 + kC(s)G(s)\}$.

Algumas funções facilitadas pelo `rltool`:

1. Adicionar e remover pólos e zeros do controlador.
2. Modificar a posição de pólos do controlador diretamente sobre o lugar das raízes (clique e arraste).
3. Determinar o ganho de ajuste localizando os pólos com o *mouse* sobre o lugar das raízes (clique e arraste).
4. Mostrar a posição de pólos do sistema realimentado para o ajuste de ganho definido (em vermelho).
5. Avaliar a resposta ao degrau do sistema realimentado para o ajuste de ganho definido (*step* na base do gráfico).
6. Observar a localização de todos os pólos do sistema realimentado e sua movimentação com a variação do ganho de ajuste.

Repita o problema do item 1 agora utilizando o `rltool`.

3. Considere o processo $G(s) = \frac{1}{s(s^2+10s+50)}$ e o controlador PD na forma $C(s) = k(s+z)/(s+50)$. Utilizando o `rltool`:

1. Para $z = 50$, isto é, controle proporcional, determine o ganho crítico (ganho que leva o sistema realimentado ao limite da estabilidade BIBO).
2. Utilizando o *mouse*, clique e arraste o zero do controlador para a direita (até próximo de -1) e avalie a influência sobre o lugar das raízes (centro de assíntotas, ângulo de assíntotas, ganho crítico, pontos de entrada e saída do eixo real.
3. Determine o valor do ganho crítico especificamente para $z = 25$, $z = 10$ e $z = 5$.
4. Para cada caso, identifique a possibilidade de dominância de pólos para o sistema realimentado e determine um ganho de ajuste que leve a resposta mais rápida possível em termos de tempo de acomodação.

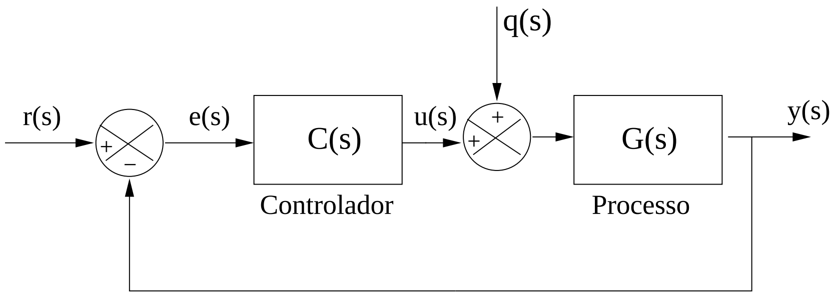
Laboratório 10

Projeto de controladores PID pelo lugar das raízes - I

Objetivos:

- Projeto de controladores via o método do lugar das raízes.
- Observação do efeito de cancelamento pólo-zero.

Atividades:
Para as atividades abaixo, considere o sistema do diagrama abaixo.



onde

$$G(s) = \frac{80}{(s+16)(s+2)}$$

considere a unidade dos pólos como min^{-1}

1. Simule o *processo* a um salto unitário e observe o tempo de acomodação do mesmo. Relacione o obtido com os parâmteros da função de transferência.

2. Projete um controlador proporcional $C(s) = K$ de forma a satisfazer as seguintes especificações: $Mo \leq 10\%$, $t_s \leq 0.5min$. Determine o valor de K que propicia mínimo erro percentual ao salto sem violar as especificações acima. Com este valor de K simule o sistema em malha fechada para uma um degrau de referência aplicado no instante $t = 0$ e a ação de uma perturbação constante aplicada no instante $t = 2min$.

3. Usando a técnica de cancelamento pólo-zero, projete um controlador PI ($C(s) = K(1 + \frac{1}{T_i s})$) de forma a satisfazer as seguintes especificações: erro nulo a referências tipo salto, rejeição assintótica de perturbações constantes, $Mo \leq 10\%$, $t_s \leq 0.5min$, tempo de subida o menor possível. Com o controlador projetado simule o sistema em malha fechada para uma um degrau de referência aplicado no instante $t = 0$ e a ação de uma perturbação constante aplicada no instante $t = 2min$.

4. Considerando o controlador projetado acima, obtenha as funções de transferência $T_r(s) = \frac{y(s)}{r(s)}$ e $T_q(s) = \frac{y(s)}{q(s)}$. Com base nestas funções, justifique o resultado da simulação e conclua sobre os efeitos do cancelamento pólo-zero na rejeição de perturbações.

5. Re-projete o controlador de forma a deixar a resposta a perturbação com tempo de acomodação menor que $1min$ e com uma reposta a mudança de referência também da ordem de $1min$ e com $Mo \leq 10\%$. Com o controlador projetado simule o sistema em malha fechada para uma mudança de referência constante aplicada no instante $t = 0$ e a ação de uma perturbação constante aplicada no instante $t = 3min$ e verifique se estas especificações são realmente satisfeitas.

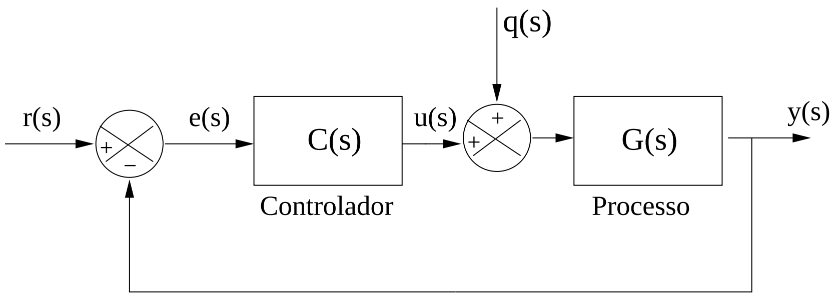
Laboratório 11

Projeto de controladores PID pelo lugar das raízes - II

Objetivos:

- Compreender o efeito de modos lentos na resposta de um processo integrador (i.e. cuja função de transferência possui um pólo na origem) controlado por um PI.
- Observar o efeito do pólo limitador de ganho em alta freqüência no projeto de controladores PD.

Atividades:
Para as atividades abaixo, considere o sistema do diagrama abaixo.



1. Considere $G(s) = \frac{15}{(s+5)s}$. Com o objetivo de alcançar rejeição assintótica a perturbações constantes, $ts \approx 2s$ e $Mo \approx 0\%$ ajuste um controlador PI. Determine a margem de ganho do sistema com o controlador obtido.

   OBS: nesta atividade faça primeiramente o projeto sem o auxílio do RLTOOL, utilizando-o apenas em uma segunda etapa para a validação e ajuste fino do controlador projetado.

2. Considere a $G(s)$ do item anterior e $C(s) = \frac{0.434(s+z)}{s}$. Para $z = -0.1$ e $z = -0.01$ obtenha:

   a) a função de transferência do sistema em malha fechada (use a função FEEDBACK).
   b) usando a expansão em frações parciais obtenha resposta analítica a referências do tipo salto (use a função RESIDUE).
   c) simule o sistema para uma mudança de referência do tipo salto (use a função STEP).
   d) com base no obtido nos itens anteriores conclua sobre o efeito da posição do zero do controlador na resposta do sistema em malha fechada.

3. Considere $G(s) = \frac{4}{(s+7)(s+2)s}$. Com o objetivo de alcançar $ts \approx 2s$, $Mo \leq 5\%$ ajuste um controlador PD considerando que o pólo limitador de ganho em alta freqüência não é fornecido pelo fabricante do controlador industrial, e, portanto, faça o projeto desprezando seu efeito.

4. Considerando agora $C(s) = K\frac{(s+z)}{(\tau s+1)}$, com K e z sendo os mesmos ajustados no item anterior, simule o sistema para referências do tipo salto para $\tau = -0.01$, $\tau = -0.05$ e $\tau = -0.1$ e conclua sobre o efeito do pólo limitador de alta freqüência no ajuste do controlador.

Laboratório 12

Projeto e implementação de controlador PID pelo lugar das raízes

Objetivos:

- Identificar a função de transferência de um sistema e validá-la por simulação.
- Utilizar a aproximação de Padé na modelagem do atraso de transporte.
- Projetar um controlador via método do lugar das raízes.
- Implementar o controlador no N1100.

Atividades:

1. Considerando que a função de transferência da planta térmica protótipo, para pequenas variações em torno de um ponto de operação, pode ser modelada pela seguinte função de transferência
$$G(s) = \frac{Temp(s)}{\%Pot(s)} = \frac{Ke^{-Ls}}{(\tau s + 1)}$$
aplicar um salto a planta, e, a partir da observação da resposta do sistema, identificar os parâmetros K, τ e L.

2. Carregar os valores experimentais (disponíveis através do arquivo gerado pelo programa supervisório) no MATLAB e plotar o gráfico do ensaio. Comparar a resposta experimental com a simulação da função de transferência obtida. Verificar se é necessário algum tipo de ajuste fino dos parâmetros.

3. Usar a aproximação de Padé para aproximar o atraso de transporte L e repetir o item anterior comparando a resposta experimental com a simulação.

4. Projetar um controlador PI, usando o método do lugar das raízes, que garanta as seguintes especificações mínimas em regime permanente:
   - seguimento de referências constantes em regime permanente;
   - rejeição assintótica a perturbações constantes;
   - tempo de estabilização menor que 2τ.

5. Implementar no N1100 os ganho referentes ao controlador projetado no item anterior, colocar o controlador a operar em modo automático e avaliar o desempenho do sistema em malha fechada, verificando se as especificações foram atendidas. Fazer, se necessário, pequenas modificações nos parâmetros.

OBS. IMPORTANTE: não esqueça da normalização com relação ao fundo de escala do sensor e das unidades nas quais devem ser programados os ganhos no N1100.