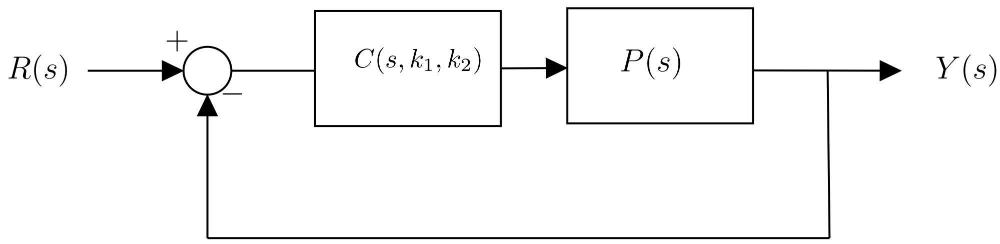

Figura 1: Sistema de controle.

Trabalho Computacional 03

Última Atualização: 3 de outubro de 2024

Objetivo: O propósito deste trabalho é determinar a região de estabilidade de um sistema de controle composto de uma planta e um controlador com dois ganhos. Para isso utiliza-se a tabela de Routh-Hurwitz e a manipulação de variáveis simbólicas disponível no Matlab/Octave.

Definição do Sistema: Considere o sistema de controle ilustrado na Figura 1 em que $P(s)$ é uma planta a ser controlada e $C(s, k_1, k_2)$ é um controlador que depende de dois ganhos de controle k_1 e k_2. O objetivo é determinar faixas para k_1 e k_2 tais que o sistema realimentado $G(s) = Y(s)/R(s)$ seja BIBO estável.

Tarefas: A primeira tarefa é implementar um *script* para construir a tabela de Routh-Hurwitz considerando k_1 e k_2 como variáveis simbólicas. Com a primeira coluna da tabela determinada, implementa-se a segunda tarefa, que é construir uma figura para ilustrar graficamente a região definida pelas desigualdades geradas pela primeira coluna da tabela.

A terceira tarefa consiste em aplicar o *script* programado nos sistemas de controle que possuem os seguintes polinômios característicos:

1. $\Delta(s) = s^4 + s^3 + (k_1k_2 + 2)s^2 + (k_2 + 3)s + k_1 + 4$;
2. $\Delta(s) = s^4 + 7s^3 + (2k_2 - k_1)s^2 + (5 + k_1k_2)s + 1$;
3. $\Delta(s) = s^4 + 2s^3 + (k_1 + 2)s^2 + (3 + k_2)s + k_1k_2$;
4. Proponha um polinômio com grau entre 3 e 10, e com k_1 e k_2 aparecendo nos coeficientes (escolha arbitrária) de modo que exista uma região estável não vazia.

A especificação do polinômio característico de interesse e a chamada do *script* desenvolvido são exemplificadas no Apêndice A.

Auxílio para desenhar o gráfico: No apêndice B é apresentado um código que determina a região de interesse a partir da primeira coluna da tabela usando a "força bruta". Basicamente o código realiza uma grade no espaço de k_1 e k_2 e testa as desigualdades ponto a ponto. No final tem-se armazenado todos os pares (k_1, k_2) tais que a primeira

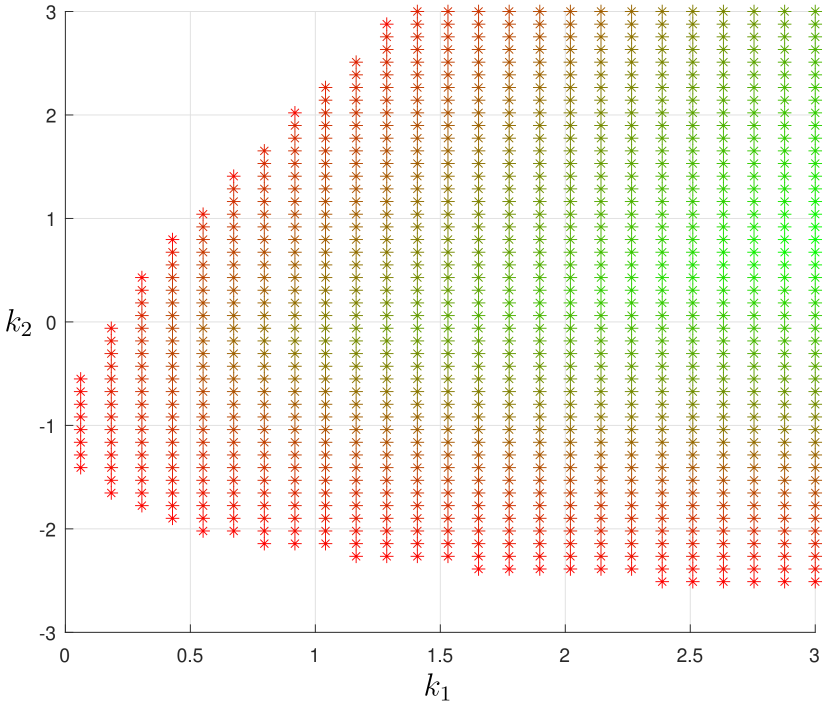

Figura 2: Região estável para $k_i \in [-3, 3]$, $i = 1, 2$, com $\Delta(s) = s^4 + 2s^3 + (k_1 + 2)s^2 + (3 + k_2)s + 1$.

coluna é positiva, ou seja, o sistema é estável. Quando mais pontos forem considerado na grade, melhor é o resultado gráfico, mas ao preço de um maior esforço computacional. O espaço de k_1 e k_2 está limitado às faixas:

$$-3 \leq k_1 \leq 3, \quad -3 \leq k_2 \leq 3$$

Com os pontos da região de estabilidade determinados, a função `plot` pode ser utilizada para desenhá-los. A Figura 2 mostra como o resultado deve ficar.

Formato de entrega: Arquivo PDF contendo a identificação da disciplina e dos alunos (nome e RA), e

- Quatro gráficos, um para cada um dos polinômios característicos indicados anteriormente.
- Informar quantos pontos foram considerados na grade, e o tempo computacional (sem segundos) demandado para criar as figuras.
- Todos os códigos fontes utilizados, incluindo aqueles que foram disponibilizados;

Pontos Extras:

- A região de interesse deve ser desenhada com um gradiente entre duas cores C_1 e C_2 arbitrárias. Seja $t_{\min}$ o menor valor obtido na primeira coluna da tabela para certos valores de k_1 e k_2. O menor valor de $t_{\min}$ (entre todos os pares (k_1, k_2)

admissíveis) deve ser desenhado na cor C_1 e o maior na cor C_2. Interpola-se os casos intermediários. A Figura 2 foi feita considerando essa opção. Valor da tarefa: 0,5 ponto (escala de 0 a 10);

- Utilize uma linguagem de programação diferente de Matlab e Octave, e ganhe 0,5 ponto;
- Substitua o código de desenho baseado na força bruta por algo melhor. A nova proposta deve funcionar para qualquer outro polinômio característico que o professor queira testar. Valor da tarefa: 2,0 pontos.

Apêndice A

O código a seguir ilustra como o script programado deve ser chamado, por exemplo, para o polinômio característico $\Delta(s) = s^4 + 2s^3 + (k_1 + 2)s^2 + (3 + k_2)s + 1$.

```

syms k1 k2;

polinomio = [1 2 k1+2 3+k2 1];
regiaoEstavel_k1k2(polinomio);
```

Apêndice B

O código a seguir fornece uma maneira de determinar os pares (k_1, k_2) que estabilizam o sistema em malha fechada. A variável `tabela` é a tabela de Routh-Hurwitz construída. `vars` é uma variável contendo as variáveis simbólicas, neste caso k_1 e k_2. Os limites para k_1 e k_2 são especificados por meio de k_1 e k_2 na forma

$$\texttt{limk1} = [-3, 3], \quad \texttt{limk2} = [-3, 3].$$

Finalmente, a variável `numPontos` especifica o número de pontos a serem utilizados na grade. A variável de saída `ptsEstaveis` é estruturada na forma

$$\begin{bmatrix} k_{1_1} & k_{2_1} & t_{1_{\min}} \\ k_{1_2} & k_{2_2} & t_{2_{\min}} \\ k_{1_3} & k_{2_3} & t_{3_{\min}} \\ \vdots & \vdots & \vdots \end{bmatrix}$$

sendo (k_{1_i}, k_{2_i}) os valores de k_1 e k_2 de uma ocorrência i e $t_{i_{\min}}$ o menor valor da primeira coluna associado (garantidamente positivo).

```matlab
%-----------------------------------------------------------------
function [ptsEstaveis] = regiaoEstavel(tabela,vars,limk1,limk2,numPontos)

k1=linspace(limk1(1),limk1(2),numPontos);
k2=linspace(limk2(1),limk2(2),numPontos);

ptsEstaveis = [];
for i=1:length(k1)
    for j=1:length(k2)
        estavel = 1;
        minV=1e10;
        for k=1:size(tabela,1)
            elemento=tabela(k,1);
            if ~isempty(symvar(elemento))
                valor=subs(elemento,vars,[k1(i) k2(j)]);
                if valor <= 0
                    estavel = 0;
                    break;
                end
                if valor < minV
                    minV = valor;
                end
            end
        end
        if estavel
            ptsEstaveis = [ptsEstaveis; k1(i) k2(j) minV];
        end
    end
end
```