UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia

Departamento de Computação

Controle e Servomecanismo

Prática 4 - Simulação de malha fechada e análise de estabilidade pelo critério de Routh

Professor: Prof. Roberto Santos Inoue

Alexandre Strabello, 770076, Engenharia Física

São Carlos, 6 de julho de 2023

1 Pré Tutorial

Todos os códigos, imagens e documentos utilizados para a escrita do presente relatório estão disponíveis no repositório `https://github.com/alestrab/Controle-e-Servo`, na pasta 'prática 4'.

Um sistema em malha fechada é dado por uma realimentação do sinal de saída, conforme pode ser observado na Figura 1.

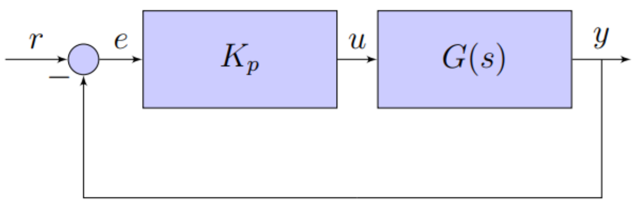

Figura 1: Diagrama de blocos de um sistema com malha fechada

Pode-se determinar a função de transferência do sistema, que será dada por:

$$\frac{Y(s)}{R(s)} = \frac{K_p\, G}{1 + K_p\, G}, \tag{1}$$

em que K_p é uma constante e $G(s) = \dfrac{K}{s\,(J\,s + b)\,(L\,s + R) + K^2}$, representando a função de transferência do motor CC que fornece a posição angular do motor.

Para análise da estabilidade podemos utilizar o chamado critério de Routh. Já observamos que a função $G(s)$ vai apresentar um polinômio de ordem 3 em seu denominador, com o termo referente à s^0 sendo nulo, de modo que já está definido que esse sistema é instável.

Em situações onde todos os coeficientes são positivos, a análise de estabilidade é seguida com a tabela de Routh. A mesma pode ser feita manualmente, porém o Matlab e outros softwares permitem a confecção dessa tabela de modo automático. Ao contrário do Scilab, que possui uma função $routh_t$ para elaboração dessa tabela, o Matlab não tem algo similar. A solução é utilizar uma função externa, desenvolvida por usuários do Matlab, que realize a confecção da tabela de Routh. O código abaixo é um exemplo, obtido em `https://www.mathworks.com/matlabcentral/fileexchange/58-routh-m`:

```
function RA=routh(poli,epsilon);

%ROUTH   Routh array.
%   RA=ROUTH(R,EPSILON) returns the symbolic Routh array RA for
%   polynomial R(s). The following special cases are considered:
%   1) zero first elements and 2) rows of zeros. All zero first
%   elements are replaced with the symbolic variable EPSILON
%   which can be later substituted with positive and negative
```

```
%   small numbers using SUBS(RA,EPSILON,...). When a row of
%   zeros is found, the auxiliary polynomial is used.
%
% Examples:
%
% 1) Routh array for s^3+2*s^2+3*s+1
%
%   >>syms EPS
%   >>ra=routh([1 2 3 1],EPS)
%   ra =
%
%        1.0000    3.0000
%        2.0000    1.0000
%        2.5000         0
%        1.0000         0
%
% 2) Routh array for s^3+a*s^2+b*s+c
%
%   >>syms a b c EPS;
%   >>ra=routh([1 a b c],EPS);
%   ra =
%
%   [          1,          b]
%   [          a,          c]
%   [ (-c+b*a)/a,          0]
%   [          c,          0]
%
%
%   Author:Rivera-Santos, Edmundo J.
%   E-mail:edmundo@alum.mit.edu
%

if(nargin<2),
  fprintf('\nError: Not enough input arguments given.');
  return
end

dim=size(poli);   %get size of poli

coeff=dim(2);       %get number of coefficients
RA=sym(zeros(coeff,ceil(coeff/2))); %initialize symbolic Routh array

for i=1:coeff,
  RA(2-rem(i,2),ceil(i/2))=poli(i); %assemble 1st and 2nd rows
end

rows=coeff-2;   %number of rows that need determinants
```

```
index=zeros(rows,1);  %initialize columns-per-row index vector

for i=1:rows,
  index(rows-i+1)=ceil(i/2); %form index vector from bottom to top
end

for i=3:coeff,          %go from 3rd row to last
  if(all(RA(i-1,:)==0)),     %row of zeros
      fprintf('\nSpecial Case: Row of zeros detected.');
      a=coeff-i+2;     %order of auxiliary equation
      b=ceil(a/2)-rem(a,2)+1; %number of auxiliary coefficients
      temp1=RA(i-2,1:b);  %get auxiliary polynomial
      temp2=a:-2:0;   %auxilirY polynomial powers
      RA(i-1,1:b)=temp1.*temp2; %derivative of auxiliary
  elseif(RA(i-1,1)==0),    %first element in row is zero
      fprintf('\nSpecial Case: First element is zero.');
      RA(i-1,1)=epsilon;  %replace by epsilon
  end
        %compute the Routh array elements
  for j=1:index(i-2),
    RA(i,j)=-det([RA(i-2,1) RA(i-2,j+1);RA(i-1,1) RA(i-1,j+1)])/RA(i
   -1,1);
  end
end
```

Qualquer aplicação que desejar construir a tabela de Routh deve então chamar a função $routh(den, eps)$, em que den representa o denominador da função de transferência, enquanto eps é dado por alguma variável sem valor definido, em que a tabela irá trabalhar com essa variável de forma algébrica.

2 Execução e Avaliação de Resultados

O desenvolvimento da prática consiste na avaliação dos sistemas de malha aberta e fechada, avaliando a estabilidade e como ambos se comportam para diferentes entradas.

2.1 Sistema em malha fechada - Matlab

No Matlab a função de transferência do sistema em malha fechada (1) foi construída com auxílio da função $feedback(y)$, que nesse caso faz uma realimentação do sistema com ganho 1. O código implementado está dado abaixo, onde as diferentes situações propostas para a prática, como a resposta aos degraus e alterações no valor de K_p estão contempladas.

```
clear
clc

%Parametros simulacao
t_i=0;
t_f=30;

%Parametros do motor
J = 0.01;
b = 0.1;
K = 0.01;
R = 1;
L = 0.5;
Kp=1;
%Fun  o de transferencia
num=Kp*K;
den = [L*J,(R*J)+(L*b),b*R+K^2,0];
%Funcao de transferencia do sistema de malha aberta
tf_open=tf(num,den,'InputDelay',1.0);
%Realimentacao para gerar a funcao de malha fechada
tf_closed=feedback(tf_open,1);

%Apresentacao dos resultados
%Resposta ao degrau unitario
subplot(2,2,1)
[y_open,t_op] = step(tf_open,t_f);
[y_closed,t_cl] = step(tf_closed,t_f);
plot(t_op,y_open,'r-',t_cl,y_closed,'b-','LineWidth',2)
legend('Malha Aberta','Malha Fechada','Location','northwest')
title('Resposta ao degrau unit rio')
ylabel('Posi  o [rad]')
xlabel('Tempo [s]')
grid()
ax = gca;
ax.FontSize = 20;


%Resposta degrau amplitude 2
subplot(2,2,2)
[y_open,t_op] = step(2*tf_open,t_f);
[y_closed,t_cl] = step(2*tf_closed,t_f);
plot(t_op,y_open,'r-',t_cl,y_closed,'b-','LineWidth',2)
legend('Malha Aberta','Malha Fechada','Location','northwest')
title('Resposta ao degrau com amplitude 2')
ylabel('Posi  o [rad]')
xlabel('Tempo [s]')
```

```
grid()
ax = gca;
ax.FontSize = 20;

%Reposta degrau unitario com Kp=10
subplot(2,2,3)
Kp=10;
num=Kp*K;
den = [L*J,(R*J)+(L*b),b*R+K^2,0];
tf_open=tf(num,den,'InputDelay',1.0);
tf_closed=feedback(tf_open,1);
[y_open,t_op] = step(tf_open,t_f);
[y_closed,t_cl] = step(tf_closed,t_f);
plot(t_op,y_open,'r-',t_cl,y_closed,'b-','LineWidth',2)
legend('Malha Aberta','Malha Fechada','Location','northwest')
title('Resposta ao degrau unit rio com Kp=10')
ylabel('P o s i   o [rad]')
xlabel('Tempo [s]')
grid()
ax = gca;
ax.FontSize = 20;
```

A função de transferência dos sistema em malha aberta foi separada em numerador (num) e denominador (den), onde este segundo foi expandido como um polinômio de terceiro grau, dada a necessidade imposta para utilização da função tf no Matlab. As respostas obtidas para o código desenvolvido estão apresentados nas Figuras 2 e 3.

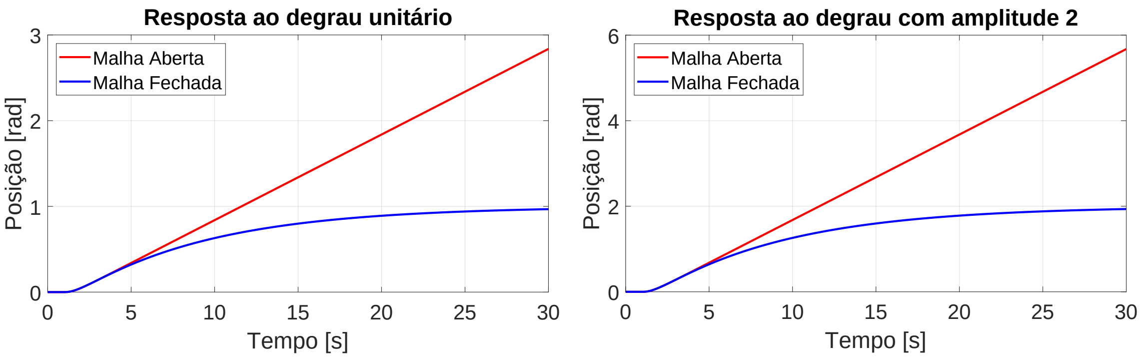

Figura 2: Resposta do sistema em malha fechada e malha aberta para entradas do tipo degrau

Conforme pode ser observado em todas os gráficos apresentados nas Figuras 2 e 3, o sistema em malha aberta é instável, como esperado pela análise pelo critério de Routh. Já para o sistema em malha fechada observa-se que o sistema tende ao valor de referência colocado pelo degrau para o caso presente na Figura 2. Isso se deve à realimentação, uma vez que ao chegar em valores próximos do sinal de referência, a subtração na entrada implica que a entrada na função de transferência é quase nula, de modo que não há alteração do sinal de saída.

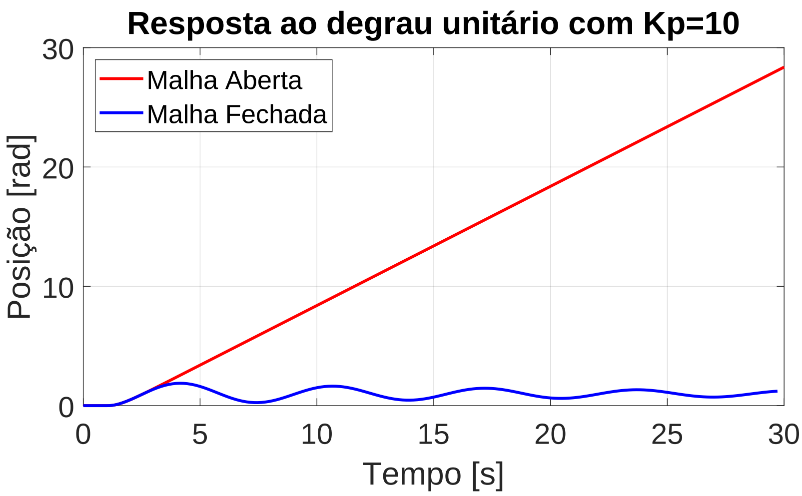

Figura 3: Resposta do sistema para um degrau unitário e com $K_p = 10$

Contudo, se K_p for muito alto, pode haver uma variação muito alta da saída do sistema, que leva o mesmo à uma posição acima da referência, conforme é observado na Figura 3. Nesse caso o sistema ainda é estável, uma vez que a amplitude das oscilações são reduzidas conforme o tempo passa. Entretanto, essa afirmação pode não ser correta dependendo dos valores utilizados para K_p.

2.2 Análise de estabilidade do sistema em malha fechada

Antes de proceder com a confecção da tabela de Routh, notemos novamente que o sistema em malha aberta não possui todos os coeficientes no denominador positivos, tal que esse sistema é instável. O sistema em malha fechada, por outro lado, passa por esse teste inicial, de modo que é necessário produzir a tabela de Routh para prosseguir com a análise. Para tanto, foi desenvolvido o código abaixo:

```
clear
clc

%Parametros simulacao
t_i=0;
t_f=30;

%Parametros do motor
J = 0.01;
b = 0.1;
K = 0.01;
R = 1;
L = 0.5;

%Fun  o de transferencia
syms Kp EPS
```

```
num=Kp*K;
den_op = [L*J,(R*J)+(L*b),b*R+K^2,0];
den_cl = [L*J,(R*J)+(L*b),b*R+K^2,Kp*K];

ra_cl=routh(den_cl,EPS)
```

Nesse caso utilizou-se K_p como uma variável qualquer, de modo que a tabela será dada como função desse parâmetro. A Tabela 1 apresenta o resultado gerado pelo código acima, onde a análise de estabilidade será dada pela verificação do sinal das células na primeira coluna.

Tabela 1: Tabela de Routh para o sistema em malha fechada

1/200	1001/10000
3/50	Kp/100
1001/10000 - Kp/1200	0
(Kp*(25*Kp - 3003))/(3000000*(Kp/1200 - 1001/10000))	0

Segundo o critério de Routh, o sistema é estável se todos os valores presentes na primeira coluna forem positivos. Nesse sentido, pode-se verificar como K_p influencia nas expressões, onde identifica-se que, se $0 < K_p < 120.12$, o sistema é estável.

2.3 Controle de posição utilizando o Simulink

Ao invés de utilizar o Matlab para confecção do sistema em malha fechada, pode-se utilizar o Simulink e sua representação em blocos para executar a tarefa. Uma integração entre o Simulink e o Matlab foi realizada para a presente prática, de modo que o diagrama de blocos consistiu apenas em uma entrada do tipo degrau unitário e um bloco representado a função de transferência do sistema em malha fechada, conforme mostra a Figura 4.

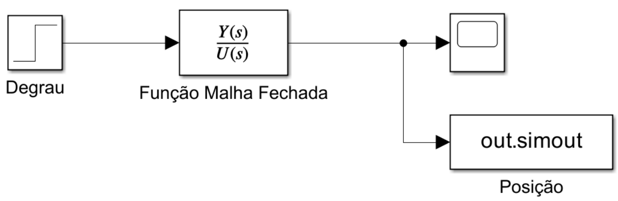

Figura 4: Representação em diagrama de blocos do sistema em malha fechada

Para controle da simulação utilizou-se o código desenvolvido a seguir:

```
clear
clc
```

```
%Parametros do motor
J = 0.01;
b = 0.1;
K = 0.01;
R = 1;
L = 0.5;

for i=1:6
    switch i
        case 1
            Kp=1;
            tit='Kp=1';
        case 2
            Kp=10;
            tit='Kp=10';
        case 3
            Kp=100;
            tit='Kp=100';
        case 4
            Kp=120.12;
            tit='Kp=120.12';
        case 5
            Kp=150;
            tit='Kp=150';
        otherwise
            Kp=200;
            tit='Kp=200';
    end
    num=Kp*K;
    den = [L*J,(R*J)+(L*b),b*R+K^2,Kp*K];
    sim('C:\Users\Alexandre\Documents\GitHub\Controle e Servo\pratica_4
   \simulink_malha.slx')
    t = ans.tout;
    y = ans.simout;
    disp(i)
    subplot(2,3,i)
    plot(t,y,'LineWidth',2)
    title(tit)
    ylabel('Posi  o Angular [rad]')
    xlabel('Tempo [s]')
    grid()
    ax = gca;
    ax.FontSize = 20;

end
```

Decidiu-se utilizar um *switch case* apenas para reduzir o tamanho do código, uma vez que apenas o valor de K_p foi alterado entre as simulações. Os resultados obtidos estão apresentados na Figura 5.

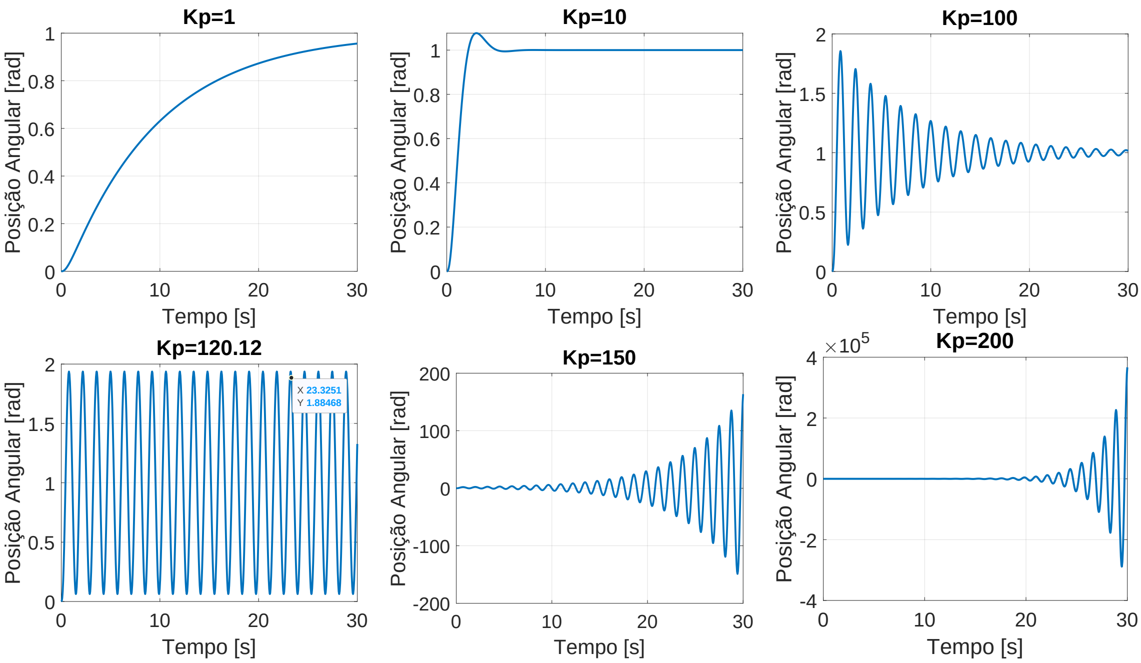

Figura 5: Resposta do sistema em malha fechada para diferentes valores de K_p

Como pode ser observado, o aumento no valor de K_p resulta primeiramente em uma redução no tempo para atingir o valor de referência, ao custo de uma maior oscilação, que é máxima exatamente no ponto onde o sistema passa a ser instável, com $K_p = 120.12$. Para valores superiores o sistema se torna instável cada vez mais rápido.

Pode-se fazer uma leve comparação entre a resposta do sistema e um oscilador harmônico amortecido. Em valores de K_p pequenos, há um amortecimento forte, que atrasa o tempo de resposta. A medida em que K_p aumenta, o amortecimento diminui, permitindo que o sistema oscile mais, porém ainda em um limite que leva à estabilidade. Quando $K_p = 120.12$ o fator de amortecimento é cancelado, convertendo então para um oscilador harmônico simples. Assim, se $K_p > 120.12$, obtém-se um fator de amortecimento negativo, que efetivamente age a favor da movimentação do sistema, resultando então em um sistema instável.