Universidade Federal de Minas Gerais - Departamento de Engenharia Eletrônica

ELT062 - OFICINA DE SIMULAÇÃO ANALÓGICA E DIGITAL EM CONTROLE

TUTORIAL 8

Instrutores: Alexandre R. Mesquita e Bruno O. S. Teixeira

Parte 1

Num motor DC controlado pela corrente de armadura, o torque τ_m fornecido é proporcional à corrente de armadura i_a:

$$\tau_m = K_{ma} i_a \ .$$

Esse torque afeta a velocidade angular ω do motor conforme a equação

$$J\dot{\omega} + c\omega = \tau_m + \tau_d \ ,$$

em que J é o momento de inércia da carga, c é o coeficiente de atrito e τ_d é um torque devido à carga. O movimento do motor gera uma tensão v_b (força contra-eletromotriz) que se contrapõe à tensão de armadura:

$$v_b = K_b \omega$$

O circuito de armadura pode ser modelado como um circuito RL, o que resulta numa função de transferência

$$\frac{1/L_a}{s + R_a/L_a}$$

de $V_a(s) - V_b(s)$ para $I_a(s)$. As equações descritas podem ser colocadas na forma do diagrama de blocos da Figura 1.

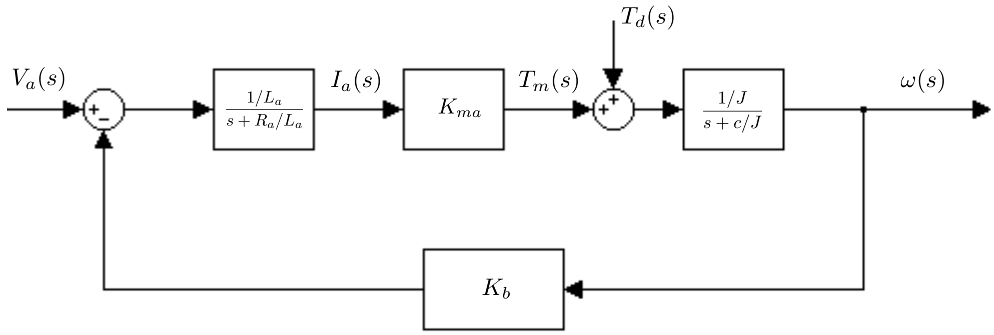

Figura 1. Diagrama de blocos para um motor DC

Nosso objetivo é encontrar a função de transferência de $T_d(s)$ e $V_a(s)$ para $\omega(s)$. Se não existisse a força contra-eletromotriz, apenas teríamos que multiplicar as funções de transferência dos três blocos restantes. A força contra-eletromotriz tem um efeito a que chamamos de realimentação (feedback): a saída do sistema afeta a própria entrada. Um exemplo do que chamamos de realimentação negativa é mostrado na Figura 2.

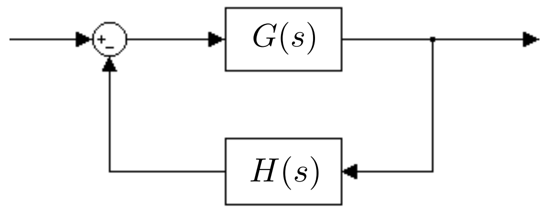

FIGURA 2. Diagrama de blocos para realimentação negativa

Para obter a função de transferência equivalente a uma ligação de realimentação negativa, podemos usar o comando

```
T=feedback(G,H);
```

No caso de realimentação positiva, ambos sinais no bloco de soma seriam positivos e usaríamos o comando

```
T=feedback(G,H,1);
```

Outra forma de conexão de sistemas que aparece na Figura 1 é a ligação em série. No Matlab, a ligação em série pode ser obtida usando-se o operador de multiplicação * ou o comando `series`. De forma semelhante, a ligação de sistemas em paralelo pode ser obtida usando o operador soma + ou o comando `parallel`.

É importante notar que usar os comandos acima de maneira indiscriminada pode resultar em problemas numéricos. Vimos três formas de representar sistemas no Matlab: `ss`, `zpk` e `tf`. Destas, `ss` é a mais estável numericamente quando realizamos interconexões de vários sistemas. Portanto, é uma boa prática usar as formas `zpk` e `tf` apenas para visualizar e criar sistemas, e evitá-las ao realizar operações. Vale notar ainda que basta que um sistema seja do tipo `ss` para que o Matlab converta os demais para o mesmo tipo antes de realizar as interconexões.

Crie um arquivo `tut8a.m`, em que calcularemos a função de transferência desejada usando dois métodos diferentes para efetuar interconexões de blocos no Matlab. Começamos definindo os parâmetros numéricos de nosso motor.

```
clear;
clc;
close all;

La=1;
Ra=1;
J=10;
c=1;
Kma=0.1;
Kb=1;
```

Em seguida definimos as funções de transferência dos três blocos superiores e a do bloco de realimentação.

```
G1=tf([1/La],[1 Ra/La]);
G2=tf(Kma,1);
G3=tf([1/J],[1 c/J]);
H=tf(Kb,1);
```

Suponha que $T_d = 0$. Nesse caso, o diagrama da Figura 1 se assemelha ao da Figura 2. Podemos, portanto, usar o comando feedback para encontrar a função de transferência $T(s)$ entre $\omega(s)$ e $V_a(s)$:

```
G1=ss(G1);
T(1,1)=zpk(feedback(series(series(G1,G2),G3),H));
```

Suponha agora que $V_a = 0$. Temos novamente uma estrutura de realimentação semelhante à da Figura 2, com a diferença de que se trata agora de realimentação positiva. Assim, obtemos a função de transferência entre $\omega(s)$ e T_d:

```
T(1,2)=zpk(feedback(G3,-H*G1*G2,1))
pause
```

Note que desta vez usamos o operador * para obter a conexão em série. Dedique algum tempo para compreender como o comando acima obtém a função de transferência desejada. Note que os índices de T(1,2) indicam a função de transferência da segunda entrada para a saída.

Um método comum para visualizar funções de transferência é usando diagramas de Bode, que representam a resposta em frequência do sistema em escala logarítmica.

```
bode(T)
grid
```

Em seguida, vamos obter a mesma função de transferência T usando o comando connect, que pode ser usado para diagramas de blocos bem mais complexos que um loop de realimentação negativa. Primeiramente, vamos unir o bloco G1 ao somador que o antecede, o que pode ser feito concatenando duas cópias do bloco original:

```
G1=[G1 -G1];
G1.inputname={'Va','Vb'};
G1.outputname='Ia';
```

Dessa forma, o bloco G1 passa a ter duas entradas e uma saída. Usamos as propriedades inputname e outputname para identificar as entradas e saídas nominalmente. Fazemos o mesmo para o segundo bloco G2:

```
G2.inputname={'Ia'};
G2.outputname='Tm';
```

De forma análoga, unimos o terceiro bloco ao somador que o precede:

```
G3=[G3 G3];
G3.inputname={'Tm','Td'};
G3.outputname='w';
H.inputname='w';
H.outputname='Vb';
```

Quando usamos o comando connect, as entradas e saídas de mesmo nome serão conectadas:

```
T=connect(G1,G2,G3,H,{'Va','Td'},'w')
```

Note que usamos os dois últimos argumentos para indicar as variáveis de entrada e de saída do sistema resultante.

```
zpk(T)
pause
```

Quando executar o arquivo, você observará que $T(1,1)$ tem um pólo e um zero idênticos. Isso acontece porque em geral o Matlab não cancela pólos e zeros automaticamente. Para realizar o cancelamento, utilizamos o comando minreal:

```
T=minreal(T);
zpk(T)
```

Para visualizar as propriedades do sistema T, usamos o comando get

```
get(T)
```

Execute o arquivo e verifique se as funções de transferência coincidem. Observe as propriedades do sistema T.

Complete o arquivo para que as respostas a degrau e a impulso do sistema $T(s)$ sejam plotadas. Execute novamente e observe as respostas obtidas.

Comentário: Note entre as propriedades do sistema T que é possível definir atrasos de entrada ou saída. Quando trabalhamos com atrasos, o ideal é manter o sistema no formato ss, especialmente se desejamos fazer conexões. De fato, é impossível fazer conexões de sistemas com atraso se estes estão no formato zpk ou tf, dado que as funções de transferência resultantes não são racionais.

Parte 2

Um motor montado sobre uma plataforma está desbalanceado de forma que seu centro de massa está a uma distância r de seu centro de rotação. Se modelarmos a plataforma como um sistema massa-mola-amortecedor, conforme mostrado na Figura 3, teremos a seguinte equação para o deslocamento vertical x da plataforma:

$$M\ddot{x}(t) + c\dot{x}(t) + kx(t) = f(t) = mr\omega^2 \text{ sen } \omega t \ , \tag{1}$$

onde m é a massa do motor, ω é a frequência angular de operação do motor, k é a constante de mola da plataforma, c é seu fator de amortecimento e M é a massa combinada do motor e da plataforma.

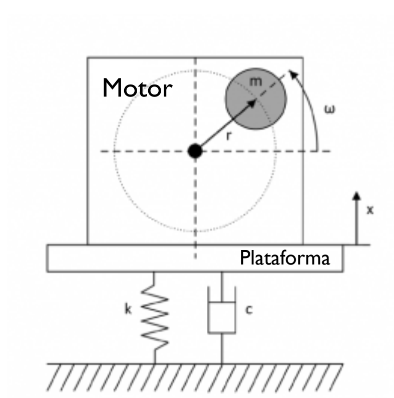

Figura 3. Diagrama equivalente de um motor desbalanceado sobre uma plataforma.

Nosso objetivo é calcular a amplitude de oscilação da plataforma devido à ressonância com a rotação do motor. Aplicando a transformada de Laplace a (1), obtemos

$$Ms^2X(s) - Msx(0) - M\dot{x}(0) + csX(s) - cx(0) + kX(s) = F(s) = mr\omega^2 \frac{\omega}{s^2+\omega^2} \ .$$

Note que desta vez queremos levar em conta as condições iniciais. Resolvendo para $X(s)$, temos

$$X(s) = \frac{1}{Ms^2+cs+k}F(s) + \frac{Ms+c}{Ms^2+cs+k}x(0) + \frac{M}{Ms^2+cs+k}\dot{x}(0)$$

Conforme sugerido pela equação acima, as condições iniciais podem ser entendidas como entradas do sistema com suas respectivas funções de transferência. Os parâmetros do sistema são $\omega = 1633$ [rpm], $M = 29$ [kg], $m = 27$ [kg], $r = 1$ [cm], $k = 8.35 \cdot 10^5$ [N/m], $c = 0.6 \cdot 10^3$ [N$\cdot$ s/m], $x(0) = 3$ [cm] e $\dot{x}(0) = 0$ [m/s].

Crie um arquivo tut8b.m e inicie com o código abaixo, preenchendo os valores dos parâmetros no SI.

```
clear;
clc;
close all;

w=
k=
c=
M=
m=
r=

x0=
```

Em seguida, calculamos a função de transferência $G(s) = X(s)/F(s)$ e plotamos a resposta em magnitude da mesma

```
G=tf([1],[M c k]);
figure
bodemag(G);
```

Para calcular a amplitude de oscilação induzida pelo motor, usamos o comando freqresp, que avalia a função de transferência na frequência indicada. Assim, a amplitude de oscilação para uma entrada do tipo $A\cos\omega t$ será $|A||G(j\omega)|$.

```
amplitude=m*r*w^2*abs(freqresp(G,w))
```

Notaremos que, embora o motor esteja desbalanceado de apenas um 1 [cm], a plataforma sofrerá oscilações de cerca de 7 [cm]. Esse fenômeno pode ser explicado pela proximidade entre a frequência de operação do motor e a frequência de ressonância da plataforma. Calculemos o pico e a frequência de ressonância do sistema, isto é, o valor e a frequência para os quais $|G(j\omega)|$ é máximo:

```
ww=10.^(-2:0.01:3);
[gr,indwr]=max(abs(freqresp(G,ww)));
wr=ww(indwr)
```

Compare essa frequência com a frequência de operação do motor. Um método mais preciso para calcular a frequência de ressonância é usando o comando norm:

```
[gr,wr]=norm(G,Inf)
```

Por fim, queremos encontrar a solução exata para a equação (1). Para isso, usamos o comando series para achar a contribuição devida à entrada $F(s)$:

```
F=m*r*w^2*tf([w],[1 0 w^2]);
X=series(G,F);
t=0:0.002:0.7;
xf=impulse(X,t);
```

Note que, como sabemos que não há cancelamentos de pólos e zeros aqui, dispensamos o uso de minreal. Em seguida, calculamos a contribuição devida à condição inicial:

```
X0=x0*tf([M c],[M c k]);
xt0=impulse(X0,t);
```

Encerramos plotando a solução obtida e calculando a amplitude de oscilação após o transitório.

```
figure
y=xf+xt0
plot(t,y)

amplitude1=max(y(end/2:end))
```

Execute o arquivo e confirme as observações indicadas. Verifique se a amplitude calculada a partir da resposta temporal coincide com aquela calculada pela resposta em frequência.