Experiência 4:

Simulação de Sistemas Dinâmicos

Última atualização: 12 de Maio de 2021

Roteiro

Última atualização: 12 de Maio de 2021

Nota: Os símbolos g, t, d, s e v indicam a necessidade de produção de um gráfico, desenvolvimento teórico (ou valores que devem ser calculados), diagrama simulink, *script* Matlab e vídeo, respectivamente. Detalhes sobre o relatório e vídeo demonstrativo:

- O relatório deve ser entregue em formato zip (ou rar) e deve conter um arquivo PDF com o conteúdo do relatório e eventuais diagramas simulink (arquivos com extensões `.slx` ou `.mdl`);
- As respostas e valores (indicados por t) e devem ser informados no texto do relatório;
- Os gráficos devem ser incorporados no texto do relatório;
- Os *scripts* devem ser apresentados no final do relatório;
- O vídeo demonstrativo do relatório (obrigatório) e os vídeos solicitados no roteiro não devem ser enviados para o Google Classroom. Apenas os *links* devem ser informados (no relatório).

Atividade 1 Considere um sistema dinâmico representado pela equação diferencial

$$\dddot{y}(t) + (2+6p)\ddot{y}(t) + (9+12p)\dot{y}(t) + 18y(t) = 18x(t),$$

em que p é um parâmetro desconhecido, mas que certamente pertence à faixa $p \in [0.1, 1.2]$. Um dos requisitos de desempenho importantes em um projeto de controle é o máximo sobressinal M_p (em inglês, *overshoot*) para uma entrada constante, como ilustra a Figura 1 (no caso, $x(t)$ é o degrau unitário $u(t)$). O valor M_p é dado por $(y_{max} - y_{final})/y_{final}$. Para esse sistema, em função da incerteza do parâmetro p, a determinação do máximo sobressinal pode ser feita por meio de simulações exaustivas, testando diversos valores de p. Para fazer esse cálculo, realize as tarefas listadas a seguir.

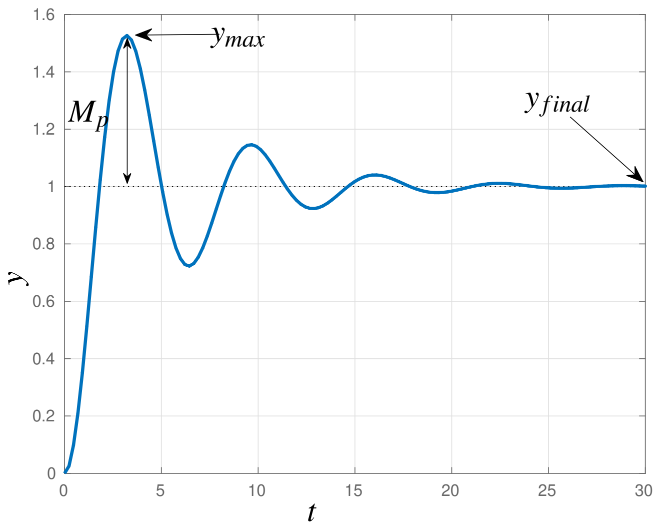

Figura 1: Máximo sobressinal M_p para uma entrada constante ($x(t) = u(t)$).

1. Implemente um diagrama em Simulink/Matlab (utilizando integradores) que represente a equação diferencial dada para condições iniciais nulas ⓓ. Para o sinal de entrada, considere o bloco `step` (degrau). Sugestão para o nome do diagrama: `sistemaLinearOrdem3.slx`

2. O parâmetro p deve ser implementado como uma variável. Utilize um bloco do tipo `out` para armazenar a saída $y(t)$.

3. Implemente um *script* em Matlab ⓢ que faz a chamada do diagrama desenvolvido no item 1 para diversos valores de p (sugestão: testar 50 valores igualmente espaçados na faixa de p). Para cada valor de p, o seguinte comando pode ser executado:

   ```
   simOut = sim('sistemaLinearOrdem3','SrcWorkspace','current','maxstep','0.1');
   ```

   As variáveis `simOut.yout{1}.Values.Time` e `simOut.yout{1}.Values.Data` são os instantes de tempo e os valores de y, respectivamente, gerados pela simulação. O máximo sobressinal pode ser calculado determinando-se o maior valor de y. Dica: utilize o comando `[m,pm]=max(y)` para determinar o valor máximo, sendo que `pm` informa a posição do vetor em que o valor máximo está armazenado.

4. Valores que devem ser computados a partir da implementação:

   (a) O maior valor de M_p ⓣ e para qual valor de p M_p ocorre ⓣ. Também forneça o instante de tempo em que esse M_p acontece ⓣ. Outros dois critérios importantes em sistemas de controle são

   - *tempo de subida* (t_s): tempo necessário para a trajetória alcançar 100% do valor de referência aplicado na entrada.
   - *tempo de acomodação* (t_a): instante a partir do qual a trajetória permanece dentro de uma faixa de 2% de seu valor final.

   Compute t_s e t_a ⓣ. Preferencialmente esses valores devem ser determinados pelo script construído. Contudo, a inspeção gráfica também pode ser utilizada.

   (b) Para qual subfaixa dentro do intervalo de p o sobressinal é menor ou igual 0.07 (7%) ⓣ?

   (c) Para o valor de p que gera o valor de M_p mais próximo de 7%, determine os polos do sistema ⓣ. Crie um objeto `tf` (*transfer function*) e utilize o comando `tfdata` para determinar os polos.

   (d) Para qual subfaixa dentro do intervalo de p não existe sobressinal (numericamente, considere $M_p \leq 0.01$) ⓣ?

Atividade 2 Considere o pêndulo simples ilustrado na Figura 2, no qual g é a aceleração da gravidade e ℓ é o comprimento da haste rígida (presa pela extremidade superior) que balança uma massa m. O movimento do pêndulo é determinado pela equação diferencial

$$\ddot{\theta}(t) + \beta\dot{\theta}(t) + \frac{g}{\ell}\operatorname{sen}\theta(t) = 0,$$

em que β é um coeficiente de amortecimento proporcional à velocidade angular do pêndulo. Adotando $\beta = 0$ (pêndulo não amortecido), realize as tarefas apresentadas a seguir.

(a) Represente o sistema em espaço de estados e mostre que os pontos de equilíbrio são $(n\pi, 0)$, $n = 0, \pm 1, \pm 2, \ldots$ ⓣ;

(b) Forneça o modelo linearizado em torno dos pontos de equilíbrio $(0,0)$ ⓣ e $(\pi, 0)$ ⓣ. Adote $\ell = 1$;

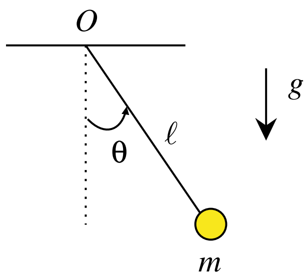

Figura 2: Pêndulo simples.

(c) Analise a estabilidade dos sistemas linearizados (estável, assintoticamente estável ou instável) obtidos no item anterior ⓣ;

(d)
- Utilizando o comando `ode45`, implemente um *script* ⓢ que simula o sistema linearizado e o sistema não linear (por 10 segundos) para a condição inicial $(\theta(0), \dot{\theta}(0)) = (\pi/10, 0)$, e armazene as trajetórias. Os instantes de tempo das duas trajetórias devem ser os mesmos.
- Utilizando como apoio o *script* `geraVideo.m` apresentado no Apêndice A, produza um vídeo ⓥ ilustrando o movimento do pêndulo para as trajetórias obtidas no item anterior. As duas trajetórias (do sistema linearizado e do não linear) devem ser desenhadas simultaneamente. O pêndulo para a trajetória do modelo linearizado (não linear) deve ser desenhado na cor azul (vermelha). O comando `line([0 xc],[0 yc])` pode ser utilizado para desenhar a haste, sendo `xc` e `yc` as coordenadas do centro da massa (note que a origem da haste é o ponto $(0,0)$). O comando `plot(xc,yc,'o','MarkerSize',10,'MarkerFaceColor','r')` pode ser utilizado para desenhar a massa (cor vermelha).
- Repita o procedimento para a condição inicial $(\theta(0), \dot{\theta}(0)) = (\pi/4, 0)$. Um segundo vídeo deve ser gerado ⓥ. Apenas para essa condição inicial, forneça um gráfico contendo o plano de fase das trajetórias dos pêndulos não linear (utilize a cor vermelha) e linearizado (utilize a cor azul) ⓖ.
- Determine os instantes de tempo, para as duas simulações, a partir dos quais existe uma defasagem maior ou igual a dois graus entre os dois pêndulos ⓣ. Comente sobre eventuais diferenças no movimento dos dois modelos (linear e não linear) em função das condições iniciais ⓣ.

Atividade 3 Considere o pêndulo duplo ilustrado na Figura 3, em que m_1 e m_2 são as massas dos corpos. O valor ℓ_1 é a distância da origem O ao corpo de massa m_1 e ℓ_2 é a distância entre o corpo de massa m_1 e o corpo de massa m_2. A dinâmica do movimento dos pêndulos pode ser modelada por meio do seguinte conjunto de equações diferenciais não lineares

$$
\begin{aligned}
\dot{\theta}_1 &= \omega_1 \\
\dot{\omega}_1 &= \frac{-g(2m_1+m_2)\mathrm{sen}(\theta_1) - m_2 g\,\mathrm{sen}(\theta_1-2\theta_2) - 2\mathrm{sen}(\theta_1-\theta_2)m_2(\omega_2^2\ell_2+\omega_1^2\ell_1\cos(\theta_1-\theta_2))}{\ell_1(2m_1+m_2-m_2\cos(2\theta_1-2\theta_2))} \\
\dot{\theta}_2 &= \omega_2 \\
\dot{\omega}_2 &= \frac{2\mathrm{sen}(\theta_1-\theta_2)(\omega_1^2\ell_1(m_1+m_2)+g(m_1+m_2)\cos(\theta_1)+\omega_2^2\ell_2 m_2\cos(\theta_1-\theta_2))}{\ell_2(2m_1+m_2-m_2\cos(2\theta_1-2\theta_2))}
\end{aligned}
$$

em que ω_i e $\dot{\omega}_i$ são a velocidade e a aceleração do corpo i, respectivamente.

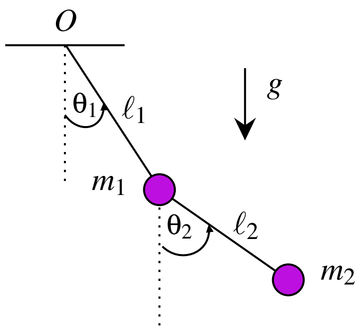

Figura 3: Pêndulo duplo.

1. Utilizando o comando `ode45`, desenvolva um *script* para simular a dinâmica do pêndulo ⓢ, produzindo um vídeo de 10 segundos para alguma condição inicial (escolha livremente, mas informe no relatório) ⓥ.

2. Os valores de m_1, m_2, ℓ_1 e ℓ_1 podem ser escolhidos livremente dentro da faixa $[0.5, 2]$. Os valores escolhidos devem ser informados no relatório ⓣ.

3. Para facilitar o desenvolvimento, o Apêndice B fornece as equações do movimento implementadas em Matlab. Utilize o código do Apêndice A como ponto de partida para o desenvolvimento do vídeo.

Apêndice A

```
function geraVideo(tempo,estadosN,estadosL,tempoTotal,L)
% entradas:
%       tempo: vetor contendo os instantes de tempo da simulação
%       estadosN: vetor contando os estados (theta e dot_theta)
   da simulação não linear
%       estadosL: vetor contando os estados (theta e dot_theta)
   da simulação linear
%       tempoTotal: tempo total da simulação
%       L: comprimento do fio (metros)

writerObj = VideoWriter('pendSimples.avi','Motion JPEG AVI');
writerObj.FrameRate = ceil(size(tempo,2)/tempoTotal);
open(writerObj);
fig = figure();
ori = [0 0]; % origem do pêndulo
f=1;
while f <= length(tempo)
        theta = estadosN(1,f);
        %determine aqui as coordenadas do corpo para o caso não
           linear (use L)
        x1 = ...
        y1 = ...
        %determine aqui as coordenadas do corpo para o caso
           linearizado (use L)
        x2 = ...
        y2 = ...


        axis([-2 2 -2.5 0.5]);
        hold on;
        %desenhe os pêndulos aqui

        hold off;
        F = getframe;
        writeVideo(writerObj,F);
        clf(fig);
        f=f+1;
end
close(writerObj);
```

Apêndice B

```
function dX = penduloDuplo(t,x)

global L1;
global L2;
global m1;
global m2;
global g;

%variaveis de estado x1=theta_1, x2=w_1, x3=theta_2, x4=w_2
dX(1,1) = x(2);
num1 = -g*(2*m1 + m2)*sin(x(1)) -m2*g*sin(x(1) -2*x(3)) -2*sin(x
    (1)-x(3))*m2*(x(4)^2*L2 + x(2)^2*L1*cos(x(1) -x(3)));
den1 = L1*(2*m1+m2-m2*cos(2*x(1)-2*x(3)));
num2 = 2*sin(x(1)-x(3))*(x(2)^2*L1*(m1+m2)+g*(m1+m2)*cos(x(1))+x
    (4)^2*L2*m2*cos(x(1)-x(3)));
den2 = L2*(2*m1+m2-m2*cos(2*x(1)-2*x(3)));
dX(2,1) = num1/den1;
dX(3,1) = x(4);
dX(4,1) = num2/den2;
```