Trabalho Computacional 01

Última Atualização: 19 de agosto de 2024

Objetivo: O propósito deste trabalho é exercitar a solução de equações diferenciais lineares e não lineares usando um resolvedor numérico, em particular, os resolvedores disponíveis no Matlab e Octave.

Experimento 1: Considere o problema de controle de velocidade de um carro apresentado na aula 2 do curso. O Apêndice 1 apresenta parte do código mostrado em aula, que foi projetado para regular a velocidade do carro em 50 km/h. A tarefa é criar o sinal de pertubação w (inclinação da pista) de modo a descrever os seguintes trajetos:

(a) Começa em $w = 0$ e no instante $t = 4$ s inicia uma subida de inclinação de 20 graus, e assim permanece até $t = 8$ s, quando imediatamente entra em um declive de -30 graus. Em $t = 12$ s o carro volta a estar em um plano e vai até o final.

(b) Começa em $w = 0$ e no instante $t = 3$ s começa uma descida de inclinação de -40 graus, e assim permanece até $t = 6$ s, quando volta a um trajeto plano. Em entra $t = 8$ s começa uma subida de inclinação de 35 graus, e assim permanece até $t = 12$ s, quando volta a um trajeto plano.

Observação importante: os valores em graus devem ser convertidos para radianos na programação.

Apresentação dos resultados:

- Figura contendo o sinal $w(t)$ do caso (a), a trajetória $v(t)$ resultante do modelo não linear e a trajetória $v(t)$ resultante do modelo linear. Usar o comando `subplot` para colocar os três gráficos na mesma figura.
- Idem ao item anterior mas agora com $w(t)$ construído em (b).

Experimento 2: Considere o movimento de um pêndulo não-amortecido descrito por

$$\ddot{\theta}(t) + a \sin \theta(t) = 0, \qquad a = \frac{g}{\ell}$$

Uma representação de estados possível para essa dinâmica é dada por

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= -a \sin(x_1(t)) \end{aligned}$$

com $x_1 = \theta$ e $x_2 = \dot{\theta}$. Tarefas:

(a) Determine o modelo linearizado ($\dot{x} = Ax$) no ponto de equilíbrio $(x_1, x_2) = (0, 0)$.

(b) Adapte os códigos fornecidos no Experimento 1 para simular os modelos não linear e linearizado do pêndulo.

(c) Simule os modelos não linear e linearizado (por 20 segundos) para as seguintes condições iniciais

$$(a)\ x(0) = (0,0873\ \ 0,0873) \qquad (b)\ x(0) = (0\ \ 0,8727) \qquad (c)\ x(0) = (0,8727\ \ 0)$$

e considere $g = 9,81$ e $\ell = 10$.

Apresentação dos resultados: Apresente uma figura com três gráficos. Cada gráfico deve mostrar o primeiro estado ($x_1(t)$, posição angular) ao longo do tempo para os modelos não linear e linearizado, e considerando uma das condições iniciais propostas. Use o `subplot`.

Experimento 3: A equação de Duffing é uma equação diferencial ordinária não-linear de segunda ordem que descreve certos osciladores forçados e amortecidos. Uma realização de estados possível é dada por

$$\begin{aligned}\dot{x}_1(t) &= x_2(t)\\ \dot{x}_2(t) &= \gamma\cos(\omega t) - \delta x_2(t) - \alpha x_1(t) - \beta x_1(t)^3\end{aligned}$$

em que x_1 e x_2 são posição e velocidade, respetivamente, e γ, ω, δ, α e β são constantes dadas. A tarefa é simular essa dinâmica usando os valores $\gamma = 4$, $\omega = 0,5$, $\delta = 0,02$, $\alpha = 1$ e $\beta = 5$. O tempo de simulação deve ser de 100 segundos e condição inicial $x(0) = (1\ \ 0)$.

Apresentação dos resultados: Figura contendo dois gráficos (use `subplot`): o primeiro contendo x_1 e x_2 ao longo do tempo. No segundo deve ser mostrado o plano de fase $x_1 \times x_2$.

Formato de entrega: Arquivo PDF contendo a identificação da disciplina e dos alunos (nome e RA), e todas as figuras solicitadas. Todos os códigos fontes utilizados devem ser apresentados após as figuras.

Pontos Extras: Utilize uma linguagem de programação diferente de Matlab e Octave para resolver as equações diferenciais e fazer os gráficos, e ganhe 1,5 pontos extras (escala de 0 a 10).

Apêndice A

```
function simulaCarro()

global g m k mu u v0 u0 w0;

g=9.81;
m=1;
k=0.01;
mu=0.1;

w0=0;
v0=13.8889;
u0=k*v0*v0+mu*m*g*cos(w0)+m*g*sin(w0);
u=u0;

intervaloTempo= [0 20];
optOde = odeset ( 'maxStep' ,0.05) ;
[t,y] = ode45(@dinamicaCarro, intervaloTempo, v0,optOde);

subplot(2,1,1)
for i=1:length(t)
    ww(i)=sinalW(t(i));
end
plot(t,ww)
grid;
ylabel w;
xlabel t;

subplot(2,1,2)
[tl,yl] = ode45(@dinamicaCarro_linear, intervaloTempo, 0,optOde);

plot(tl,yl+v0,'b',t,y,'r')
legend('linear','nao linear');
grid;
ylabel v;
xlabel t;
%_______________________________________________________________
function dotV = dinamicaCarro(t,v)

global g m k mu u v0 u0 w0;

w = sinalW(t);

dotV = (-k/m)*v*v -mu*g*cos(w)-g*sin(w)+u;

%_______________________________________________________________
function dotDeltaV = dinamicaCarro_linear(t,deltaV)

global g m k mu v0 w0;
```

```

w = sinalW(t);
deltaW=(w-w0);
deltaU=0; % implica que u=u0;

dotDeltaV = ((-2*k/m)*v0)*deltaV +(mu*g*sin(w)-g*cos(w))*deltaW
   +(1/m)*deltaU;

%_______________________________________________________________
function w = sinalW(t)

%programe aqui os casos (a) e (b)
```