

UNIVERSIDADE DE SÃO PAULO



ESCOLA POLITÉCNICA

PMR3401 - MECÂNICA COMPUTACIONAL PARA MECATRÔNICA

Relatório: Exercício Programa 1

*Henrique Yda Yamamoto - 9349502  
João Pedro do Patrocínio Ceccarelli - 8932154*

SÃO PAULO - SP

# **Conteúdo**

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Equacionamento Analítico e Numérico</b>	<b>3</b>
<b>3</b>	<b>Códigos</b>	<b>5</b>
3.1	Explicação sobre os códigos:	11
<b>4</b>	<b>Simulações</b>	<b>13</b>
4.1	Resultados	13
4.2	Discussões sobre a influência do passo	29
4.3	Perguntas	30
<b>5</b>	<b>Conclusões</b>	<b>38</b>

# 1 Introdução

No primeiro exercício programa deve-se simular as oscilações resultantes quando um veículo desloca-se, em movimento retilíneo uniforme, com duas cargas de reboque em sua traseira. Este fenômeno, que é chamado de *sway*, ocorre devido aos reboques serem pesados e também porque o carro nunca irá se movimentar apenas para frente (devido, por exemplo, à curvas, mudança de faixa etc). Dessa forma, uma pequena oscilação no movimento do veículo motor pode desencadeia fortes oscilações no sistema, sendo até mesmo possível que o sistema tombe.



Figura 1: Carro capotando, na Austrália, devido ao efeito mencionado. Retirado do vídeo <https://www.youtube.com/watch?v=nd-hUX8memY>

O objetivo deste trabalho é simular o movimento de um veículo com reboque em diferentes condições de peso e velocidade. Assim é possível estudar como estas variáveis influenciam nas oscilações.

## 2 Equacionamento Analítico e Numérico

Utilizando o diagrama de corpo livre dado na figura 2:

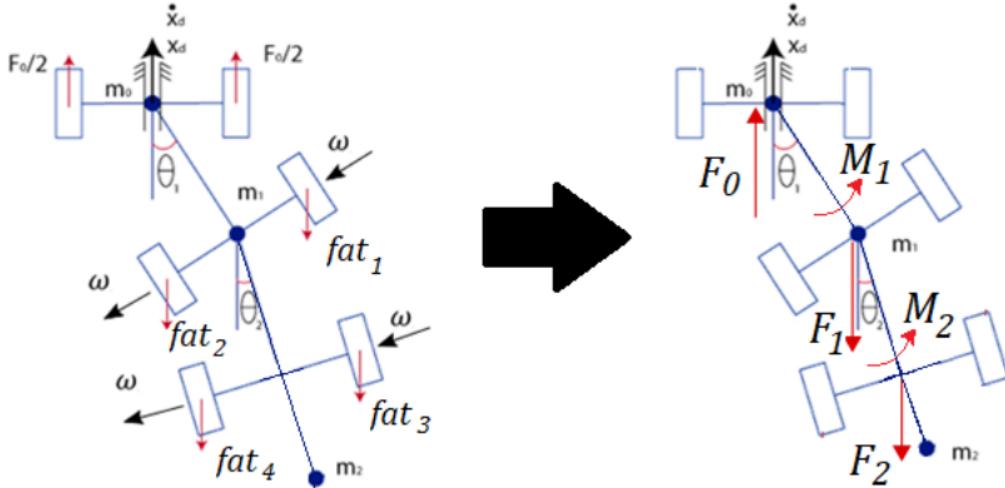


Figura 2: Diagrama de corpo livre do carro e reboque.

Temos que o sistema é dado por:

$$\ddot{\theta}_1 = \frac{A_1 \dot{\theta}_1^2 + A_2 \dot{\theta}_2^2 + A_3 \dot{\theta}_1 + A_4 \dot{\theta}_2 + A_5}{A_0} \quad (1)$$

$$\ddot{\theta}_2 = \frac{B_1 \ddot{\theta}_1 + B_2 \dot{\theta}_1^2 + B_3 \dot{\theta}_2 + B_4}{B_0} \quad (2)$$

E suas equações, por:

$$A_0 = L_1^2 L_2 R [m_2 \cos(2\theta_1 - 2\theta_2) - 2m_1 - m_2] \quad (3)$$

$$A_1 = L_1^2 L_2 R m_2 \sin(2\theta_1 - 2\theta_2) \quad (4)$$

$$A_2 = 2L_2^2 L_1 R m_2 \sin(\theta_1 - \theta_2) \quad (5)$$

$$A_3 = -2L_2 \mu I_z V e l \quad (6)$$

$$A_4 = -2L_2 \mu I_z V e l \cos(\theta_1 - \theta_2) \quad (7)$$

$$B_0 = L_2^2 R m_2 \quad (8)$$

$$B_1 = -L_1 L_2 R m_2 \cos(\theta_1 - \theta_2) \quad (9)$$

$$B_2 = L_1 L_2 R m_2 \sin(\theta_1 - \theta_2) \quad (10)$$

$$B_3 = -\mu I_z V e l \quad (11)$$

$$B_4 = L_{2eixo} \operatorname{sen}(\theta_2) R F_2 \quad (12)$$

As constantes são:

$$L_1 = 2\text{m}; L_2 = 2,5\text{m}; L_{2eixo} = 1,8\text{m}; m_1 = 450\text{kg}; m_2 = 650\text{kg}; F_1 = -0,5*m_1*g \text{ N}; F_1 = -0,5*m_2*g \text{ N}; \mu I_z = 2,7 \text{ kgm}^2; R = 0,3\text{m} g = 9,81 \frac{\text{m}}{\text{s}^2}; \dot{x}_d = Vel = 80 \frac{\text{km}}{\text{h}}$$

O método de Euler com 5 variáveis, sendo duas delas derivadas de outras duas para resolver o sistema de equações com passo predefinido  $h$  é:

$$f_{i+1}(t, x, \dot{x}, y, \dot{y}) = f_i(t, x, \dot{x}, y, \dot{y}) + h * f'_i(t, x, \dot{x}, y, \dot{y}) \quad (13)$$

Substituindo  $x, \dot{x}, y$  e  $\dot{y}$  por  $\theta_1, \theta_2, \dot{\theta}_1$  e  $\dot{\theta}_2$  na equação anterior:

$$f_{i+1}(t, \theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) = f_i(t, \theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) + h * f'_i(t, \theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) \quad (14)$$

Agora, definindo  $f_{i+1}(t, \theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) = \theta_{1,i+1}$ :

$$\theta_{1,i+1} = \theta_{1,i} + h * \dot{\theta}_{1,i} \quad (15)$$

Pode-se também fazer a equação (15) para a derivada  $\dot{\theta}_1$ :

$$\dot{\theta}_{1,i+1} = \dot{\theta}_{1,i} + h * \ddot{\theta}_{1,i} \quad (16)$$

Mas como visto na equação (1), já temos o valor de  $\ddot{\theta}_1$ , que é simplesmente  $\ddot{\theta}_1 = (A_1\dot{\theta}_1^2 + A_2\dot{\theta}_2^2 + A_3\dot{\theta}_1 + A_4\dot{\theta}_2 + A_5)/A_0$ :

$$\dot{\theta}_{1,i+1} = \dot{\theta}_{1,i} + h \frac{(A_{1i}\dot{\theta}_{1,i}^2 + A_{2i}\dot{\theta}_{2,i}^2 + A_{3i}\dot{\theta}_{1,i} + A_{4i}\dot{\theta}_{2,i} + A_{5i})}{A_{0i}} \quad (17)$$

A partir disso, já que temos condições iniciais para cada variável ( $\theta_1 = \theta_2 = 0, \dot{\theta}_1 = 0,4 \frac{\text{rad}}{\text{s}}$  e  $\dot{\theta}_2 = -0,1 \frac{\text{rad}}{\text{s}}$ ), fazendo essas substituições consecutivas a partir dessas condições iniciais pode-se obter os resultados para as equações diferenciais para cada iteração  $i$  que irá ser percorrida. Analogamente, pode-se deduzir o mesmo para  $\theta_2$  seguindo o mesmo procedimento anterior e utilizando a equação simbolizada por (2) ao invés da equação (1) na etapa correspondente.

Analogamente, para os métodos de Runge-Kutta o raciocínio é o mesmo. A diferença é que tais métodos possuem etapas intermediárias que diminuem os erros consideravelmente. Método de Runge-Kutta de segunda ordem, com  $x$  sendo um número natural qualquer de variáveis:

$$f_{i+1}(t, x) = f_i(t, x) + \frac{h(k_1 + k_2)}{2} \quad (18)$$

Sendo  $k_1 = f'_i(t, x)$  e  $k_2 = f'_i(t + \frac{h}{2}, x + \frac{k_1}{2})$  e para o método de Runge-Kutta de quarta ordem temos:

$$f_{i+1}(t, x) = f_i(t, x) + \frac{h(k_1 + 2k_2 + 2k_3 + k_4)}{6} \quad (19)$$

Sendo  $k_1 = f'_i(t, x)$ ,  $k_2 = f'_i(t + \frac{h}{2}, x + \frac{k_1}{2})$ ,  $k_3 = f'_i(t + \frac{h}{2}, x + \frac{k_2}{2})$  e  $k_4 = f'_i(t + \frac{h}{2}, x + k_3)$ . Conforme dito, apenas deve-se ir calculando mais etapas para obter melhoria na precisão do resultado final.

### 3 Códigos

Utilizamos cinco arquivos diferentes no MATLAB. O primeiro é a função  $f$ , que contém as funções correspondentes às segundas derivadas de  $\theta_1$  e  $\theta_2$  (e as constantes associadas à elas). O código abaixo corresponde à função  $f$

```
1 function [f] = f(y)
2 %f = | f1 = Omega1 |
3 %     | f2 = Omega2 |
4 %     | f3 = Omega1' |
5 %     | f4 = Omega2' |
6 % Retorna o vetor com as derivadas primeiras correspondentes s 4
7 % variáveis de estado do vetor y, montado da seguinte maneira:
8 %y = | y1 = Theta 1 |
9 %     | y2 = Theta 2 |
10 %    | y3 = Omega 1 |
11 %    | y4 = Omega 2 |
12 theta1 = y(1);
13 theta2 = y(2);
14 omega1 = y(3);
15 omega2 = y(4);
16 phi = theta1 - theta2; % Variável auxiliar utilizada para relações ...
17 % trigonométricas, de modo a
18 % simplificar a manipulação algébrica e ...
19 isolar
20 % constantes
21 g = 9.81;
22 l1 = 2;
23 l2 = 2.5;
24 l2e = 1.8;
25 m1 = 450;
26 m2 = 650;
27 F1 = -0.5*m1*g;
28 F2 = -0.5*m2*g;
29 uIz = 2.7;
30 R = 0.3;
31 xd = 80/3.6;
32 vel = xd;
33
34 % equações
35 a0 = (l1^2)*l2*R*(m2*cos(2*phi)-2*m1-m2);
36 a1 = (l1^2)*l2*R*m2*sin(2*phi);
37 a2 = 2*l1*(l2^2)*R*m2*sin(phi);
```

```

38 a3 = -2*l2*uIz*vel;
39 a4 = -2*l1*uIz*vel*cos(phi);
40 a5 = -R*l1*(l2e*F2*sin(theta1 - 2*theta2)+2*sin(theta1)*(F1*l2+(l2e...
    *F2)/2));
41
42 b0 = (l2^2)*R*m2;
43 b1 = -l1*l2*R*m2*cos(phi);
44 b2 = l1*l2*R*m2*sin(phi);
45 b3 = -uIz*vel;
46 b4 = l2e*sin(theta2)*R*F2;
47
48 f3 = (a1*omega1^2+a2*omega2^2+a3*omega1+a4*omega2+a5)/a0;
49 % Acima, encontra-se a equacao do omega1' isolado
50 f4 = (b1*f3+b2*(omega1^2)+b3*omega2+b4)/b0;
51 % Acima, encontra-se a equacao do omega2' isolado
52 f1 = y(3);
53 f2 = y(4);
54 f = [f1 f2 f3 f4];
55 end

```

Arquivo *f.m*:

Os outros três códigos abaixo correspondem às funções que contém os métodos de Euler, Runge-Kutta de 2<sup>a</sup> ordem e Runge-Kutta de 4<sup>a</sup> ordem, respectivamente:

```

1 close all;
2
3 h = [1, 0.5, 0.1, 0.01, 0.001]; %Vetor com todos os passos de teste
4 precision = [0, 1, 1, 2, 3]; %Precis o casas decimais do comando ...
    sprintf
5
6 for k = 1:length(h) %Resolve as equa es para cada passo no vetor...
    h
7 y0 = [0 0 0.4 -0.1]; %Vetor com as condic es iniciais
8 t0 = 0; % Tempo inicial da simulacao
9 tf = 60; % Tempo final da simulacao
10 N = (tf - t0)/h(k); % N mero de passos
11 y = y0; %criando o vetor y
12 t = t0; %criando vetor t
13 p0 = [0.4 -0.1 0 0]; %valores iniciais da plotagem
14 p = p0; %cria vetor de plotagem
15
16 for i = 1:N %loop para calcular as iterac es no metodo de ...
    Euler
        f_ponto = f(y(i,:)); % Derivadas primeiras calculadas na ...

```

```

    itera o i]
18     p(i,:) = f_ponto; %passa para o vetor de plotagem
19     y(i+1,:) = y(i,:)+ h(k)*f_ponto; % Y(i+1) = Y(i) + h(k)*f(Y...
(i))
20     t(i+1) = t(i) + h(k); % Atualizando tempo de simula o
21 end

22
23 y(end,:) = []; %elimina a ultima linh(k)a de y para ter o mesmo...
tamanho de p
24 y(:,end) = []; %elimina a ultima coluna de y para ter o mesmo ...
tamanho de p
25 t(end) = []; %elimina a ultima linh(k)a de t para ter o mesmo ...
tamanho de p
26
27 %passa os valores de theta1 e theta2 para p
28 for i = 1:N-1
29     p(i, 5) = y(i, 1);
30     p(i, 6) = y(i, 2);
31 end
32 Plot('Euler', k, p, t, precision, h); %chama a fun o Plot
33 end

```

Arquivo: *Euler.m*

Código do método de Runge-Kutta de 2<sup>a</sup> ordem:

```

1 close all;
2
3 h = [1, 0.5, 0.1, 0.01, 0.001]; %Vetor com todos os passos de teste
4 precision = [0, 1, 1, 2, 3]; %Precis o casas decimais do comando ...
sprintf
5
6 for k = 1:length(h) %Resolve as equa es para cada passo no vetor...
h
7 y0 = [0 0 0.4 -0.1]; %Vetor com as condi es iniciais
8 t0 = 0; % Tempo inicial da simulacao
9 tf = 60; % Tempo final da simulacao
10 N = (tf - t0)/h(k); % N mero de passos
11 y = y0; %criando o vetor y
12 t = t0; %criando vetor t
13 p0 = [0.4 -0.1 0 0]; %valores iniciais da plotagem
14 p = p0; %cria vetor de plotagem
15
16 for i = 1:N % Loop para executar as itera es do m todo de ...
Runge Kutta de 2 ordem

```

```

17     k1 = f(y(i,:)); % K1 = f(Yi)
18     k2 = f(y(i,:)+h(k)*k1); % K2 = f(Yi + h(k)*k1)
19     p(i,:) = k1; %passa para o vetor de plotagem
20     y(i+1,:) = y(i,:)+ (h(k)/2)*(k1+k2); % Y(i+1) = Y(i) + h(k...
21     )/2*(k1+k2)
22     t(i+1) = t(i) + h(k); %Atualizando o tempo de simulacao
23     end
24
25     y(end,:)=[]; %elimina a ultima linha de y para ter o mesmo ...
26     tamanho de p
27     y(:,end)=[]; %elimina a ultima coluna de y para ter o mesmo ...
28     tamanho de p
29     t(end)=[]; %elimina a ultima linha de t para ter o mesmo ...
30     tamanho de p
31
32     %passa os valores de theta1 e theta2 para p
33     for i = 1:N-1
34         p(i, 5) = y(i, 1);
35         p(i, 6) = y(i, 2);
36     end
37
38     Plot('RK2', k, p, t, precision, h); %chama a fun o Plot
39 end

```

Arquivo: *RK2.m*

Código do método de Runge-Kutta de 4<sup>a</sup> ordem:

```

1 close all;
2
3 h = [1, 0.5, 0.1, 0.01, 0.001]; %Vetor com todos os passos de teste
4 precision = [0, 1, 1, 2, 3]; %Precis o casas decimais do comando ...
5     sprintf
6
7 for k = 1:length(h) %Resolve as equa es para cada passo no vetor...
8     h
9     y0 = [0 0 0.4 -0.1]; %Vetor com as condic oes iniciais
10    t0 = 0; % Tempo inicial da simulacao
11    tf = 60; % Tempo final da simulacao
12    N = (tf - t0)/h(k); % N mero de passos
13    y = y0; %criando o vetor y
14    t = t0; %criando vetor t
15    p0 = [0.4 -0.1 0 0]; %valores iniciais da plotagem
16    p = p0; %cria vetor de plotagem
17

```

```

16     for i = 1:N % Loop para executar as iterações do m todo de ...
Runge Kutta de 4 ordem
17         k1 = f(y(i,:)); % K1 = f(Yi)
18         k2 = f(y(i,:)+(h(k)/2)*k1); % K2 = f(Yi + (h(k)/2)*k1)
19         k3 = f(y(i,:)+(h(k)/2)*k2); % K3 = f(Yi + (h(k)/2)*k2)
20         k4 = f(y(i,:)+h(k)*k3); % K4 = f(Yi + h(k)*k3)
21         p(i,:) = k1; %passa para o vetor de plotagem
22         y(i+1,:) = y(i,:) + (h(k)/6).*(k1+2.*k2+2.*k3+k4); % Y(i+1)...
= Y(i) + (h(k)/6)/(k1 + 2*k2+ 2*k3 + k4)
23         t(i+1) = t(i) + h(k); %Atualizando o tempo de simulacao
24     end
25
26     y(end,:) = []; %elimina a ultima linha de y para ter o mesmo ...
tamanho de p
27     y(:,end) = []; %elimina a ultima coluna de y para ter o mesmo ...
tamanho de p
28     t(end) = []; %elimina a ultima linha de t para ter o mesmo ...
tamanho de p
29
30     %passa os valores de theta1 e theta2 para p
31     for i = 1:N-1
32         p(i, 5) = y(i, 1);
33         p(i, 6) = y(i, 2);
34     end
35
36     Plot('RK4', k, p, t, precision, h); %chama a função Plot
37 end

```

Arquivo: *RK4.m*

O último código faz a plotagem correta das simulações colocando título, eixos e legenda e salva as figuras em .JPG.

```

1 function Plot(metodo, k, p, t, precision, h)
2
3     fontSize = 36; %Tamanho da fonte do título
4     fontSizeAxis = 24; %Tamanho da fonte dos eixos
5     lineWidth = 3; %Espessura da linha do gráfico
6
7     %Inicia o subplot
8
9     figure(k*2-1);
10    subplot(3,2,1);
11    plot(t,p(:,5),'LineWidth', lineWidth);
12    grid on;

```

```

13 xlabel('$t[s]$', 'interpreter', 'latex','FontSize', fontSize);
14 ylabel('$\theta_1[rad]$', 'interpreter', 'latex','FontSize', ...
fontSize);
15 set(gca,'FontSize',fontSizeAxis);
16
17 subplot(3,2,2);
18 plot(t,p(:,6),'LineWidth', lineWidth);
19 grid on;
20 xlabel('$t[s]$', 'interpreter', 'latex','FontSize', fontSize);
21 ylabel('$\theta_2[rad]$', 'interpreter', 'latex','FontSize', ...
fontSize);
22 set(gca,'FontSize',fontSizeAxis);
23
24 subplot(3,2,3);
25 plot(t,p(:,1),'LineWidth', lineWidth);
26 grid on;
27 xlabel('$t[s]$', 'interpreter', 'latex','FontSize', fontSize);
28 ylabel('$\dot{\theta}_1[rad/s]$', 'interpreter', 'latex',...
FontSize', fontSize);
29 set(gca,'FontSize',fontSizeAxis);
30
31 subplot(3,2,4);
32 plot(t,p(:,2),'LineWidth', lineWidth);
33 grid on;
34 xlabel('$t[s]$', 'interpreter', 'latex','FontSize', fontSize);
35 ylabel('$\dot{\theta}_2[rad/s]$', 'interpreter', 'latex',...
FontSize', fontSize);
36 set(gca,'FontSize',fontSizeAxis);
37
38 subplot(3,2,5);
39 plot(t,p(:,3),'LineWidth', lineWidth);
40 grid on;
41 xlabel('$t[s]$', 'interpreter', 'latex','FontSize', fontSize);
42 ylabel('$\ddot{\theta}_1[rad/s^2]$', 'interpreter', 'latex',...
FontSize', fontSize);
43 set(gca,'FontSize',fontSizeAxis);
44
45 subplot(3,2,6);
46 plot(t,p(:,4),'LineWidth', lineWidth);
47 grid on;
48 xlabel('$t[s]$', 'interpreter', 'latex','FontSize', fontSize);
49 ylabel('$\ddot{\theta}_2[rad/s^2]$', 'interpreter', 'latex',...
FontSize', fontSize);
50 set(gca,'FontSize',fontSizeAxis);
51
52 sgttitle(sprintf('Simula o para m todo de %s para passo h = ...
%1.*f', metodo, precision(k), h(k)),'FontSize', fontSize);

```

```

53     set(gcf, 'Units', 'Normalized', 'OuterPosition', [0 0 1 1]);
54
55     saveas(gcf, sprintf('C:\\\\Users\\\\henri\\\\Documents\\\\EP1\\\\EP1\\\\ex1...
56     \\\\%sh%1.*fseparado.jpg', metodo, precision(k), h(k))); %Salva o ...
57     arquivo em .JPG
58
59     %Plota todas as curvas no mesmo gr fico
60     figure(k*2)
61     plot(t, p(:,5), t, p(:,6), t, p(:,1), t, p(:,2), t, p(:,3), t, ...
62     p(:,4), 'LineWidth', lineWidth);%plotagem do grafico
63     title(sprintf('Simula o para m todo de %s para passo h = ...
64     %1.*f', metodo, precision(k), h(k)), 'FontSize', fontSize) ...
65     Titulo do grafico
66     legend('$\theta_1[rad]$', '$\theta_2[rad]$', '$\dot{\theta}_1[rad...
67     /s]$', '$\dot{\theta}_2[rad/s]$', '$\ddot{\theta}_1[rad/s^2]$', '$\...
68     \ddot{\theta}_2[rad/s^2]$', 'interpreter', 'latex', 'FontSize', ...
69     fontSize) %Legenda das curvas
70     xlabel('$t[s]$', 'interpreter', 'latex', 'FontSize', fontSize) ...
71     Titulo do eixo x
72     ylabel('$[rad]$ | $[rad/s]$ | $[rad/s^2]$','interpreter', '...
73     latex', 'FontSize', fontSize) % Titulo do eixo y
74     set(gca,'FontSize',fontSizeAxis);
75     set(gcf, 'Units', 'Normalized', 'OuterPosition', [0 0 1 1]);
76     saveas(gcf, sprintf('C:\\\\Users\\\\henri\\\\Documents\\\\EP1\\\\EP1\\\\ex1...
77     \\\\%sh%1.*f.jpg', metodo, precision(k), h(k))); %Salva o arquivo ...
78     em .JPG
79 end

```

Arquivo: *Plot.m*

### 3.1 Explicação sobre os códigos:

Os códigos que contém os métodos de Euler, RK2 e RK4 criados possuem todos um funcionamento bem semelhante:

- Nas primeiras linhas tem-se a escolha do valor do passo  $h$ , criação dos vetores  $y$ ,  $t$  e  $p$ , cálculo do número total de passos  $N$  e também inserção dos valores iniciais para os vetores.
- O vetor  $y$  irá se tornar uma matriz Nx4, sendo que cada uma das linhas corresponde à um dos passos calculados (em ordem crescente) e cada uma das colunas às variáveis  $\theta_1$ ,  $\theta_2$ ,  $\frac{d\theta_1}{dt}$  e  $\frac{d\theta_2}{dt}$ , de 1 a 4, respectivamente.
- O vetor  $t$  é um vetor com  $N$  posições que conta o tempo, ou seja, marca cada iteração  $i$ .
- O vetor  $p$  é um vetor que irá se tornar uma matriz (N-1)x6, e junta os valores de  $y$  com os

recebidos pela função  $f$  (que irá ser explicada em breve). É usado para plotar os gráficos.

Assim, entra-se no loop que executa as iterações de cada método:

- Sendo  $i$  a iteração atual, o loop pega os valores de  $y(i, :)$  e chama a função  $f$  a partir dos valores de  $y(i, :)$  (ou seja, faz  $f(y(i,:))$ ). A função  $f$  calcula  $\frac{d^2\theta_1}{dt}$  e  $\frac{d^2\theta_2}{dt}$ . Os valores retornados por  $f(y(i, :))$  (que são  $\frac{d\theta_1}{dt}$ ,  $\frac{d\theta_2}{dt}$ ,  $\frac{d^2\theta_1}{dt}$  e  $\frac{d^2\theta_2}{dt}$ ) passam para os vetores voláteis  $f\_ponto$  (no caso do método de Euler) ou  $K1$  (nos casos dos outros dois métodos). Logo após, esses valores são guardados na matriz  $p(i, :)$ .
- Os valores de  $K2$  (para RK2 e RK4),  $K3$  e  $K4$  (apenas para RK) são calculados a partir do que vai-se obtendo, de forma que possa-se calcular o ponto para a próxima iteração ( $y(i, :)$ ) usando as correspondentes fórmulas para cada método.
- O vetor tempo 't' é atualizado.

As próximas linhas de código correspondem à ajustes:

- Códigos que contém o comando *end* apagam as últimas linhas de  $y$  e  $t$  para que possa-se plotar o gráfico corretamente (pois  $y$  é Nx4,  $t$  é N:1 e  $p$  é (N-1)x4, o que gera problemas no final).
- No próximo loop passa-se os valores de  $theta_1$  e  $theta_2$  para  $p$ . Isso não foi feito antes pois no loop anterior  $p$  ainda continha 4 colunas, o que acarretaria em erro.

Enfim, a chamada da função Plot no final corresponde à plotagem dos gráficos, legendas e títulos.

## 4 Simulações

Utilizando as constantes dadas no enunciado foram feitas simulações para os três métodos variando decrescentemente o passo  $h$ . Foram registradas simulações com passo 1; 0,5; 0,1; 0,01 e 0,001.

### 4.1 Resultados

Os gráficos obtidos são ilustrados nesta seção. Foram plotadas as variáveis  $\theta_1$ ,  $\theta_2$ ,  $\dot{\theta}_1$ ,  $\dot{\theta}_2$ ,  $\ddot{\theta}_1$ ,  $\ddot{\theta}_2$  ao longo do tempo de simulação = 60s. Para cada caso são plotados dois gráficos: com todas as variáveis em um mesmo gráfico e com as variáveis em gráficos separados.

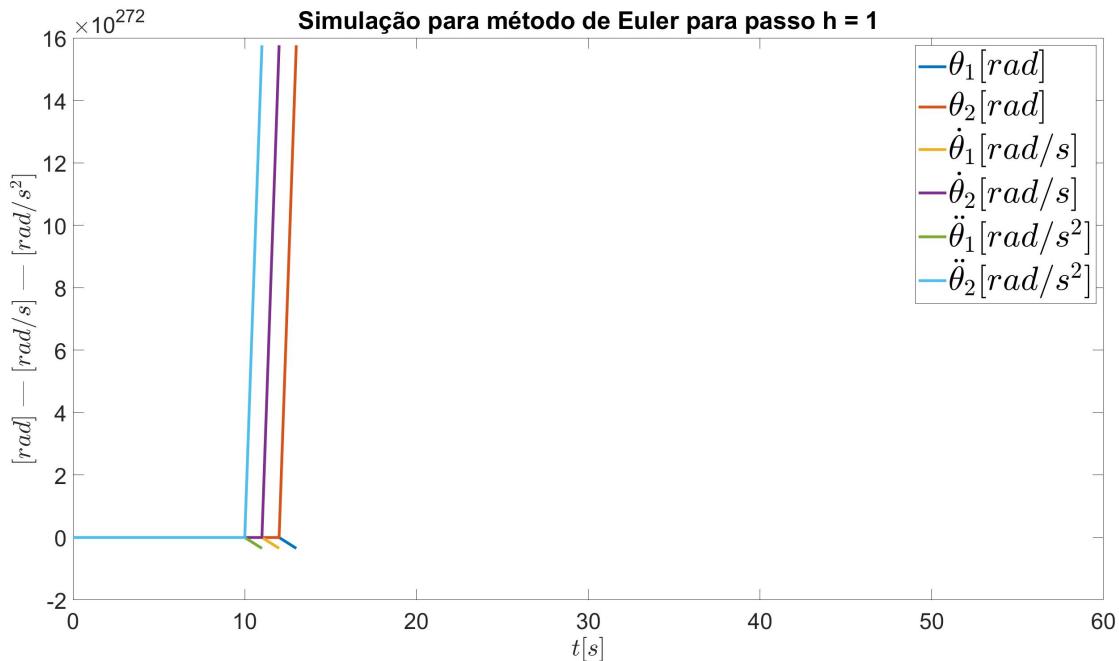


Figura 3: Método de Euler (passo  $h = 1$ )

### Simulação para método de Euler para passo $h = 1$

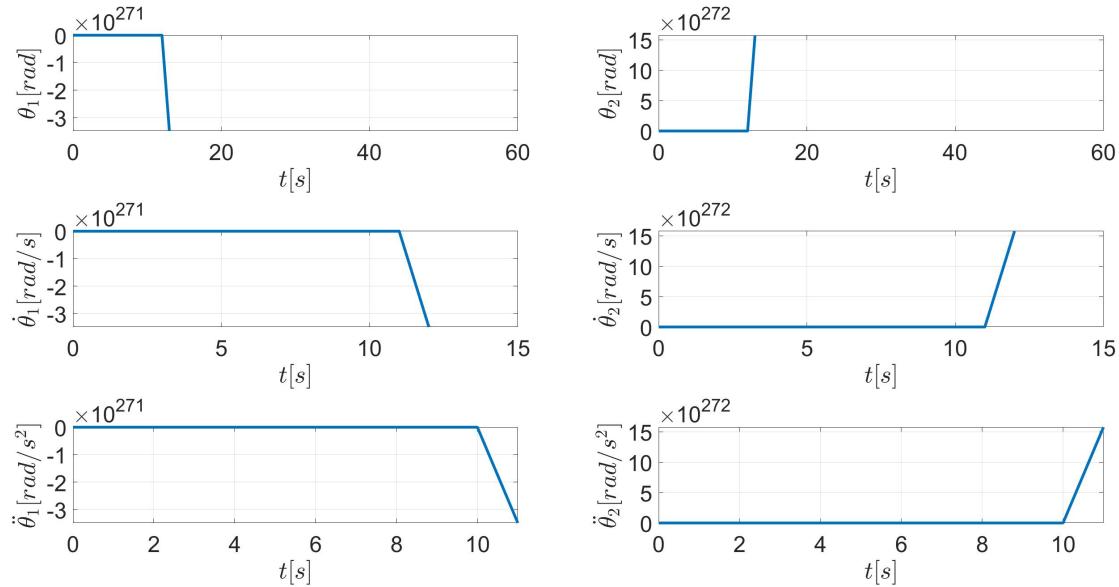


Figura 4: Método de Euler (passo  $h = 1$ ). Gráfico com subplots.

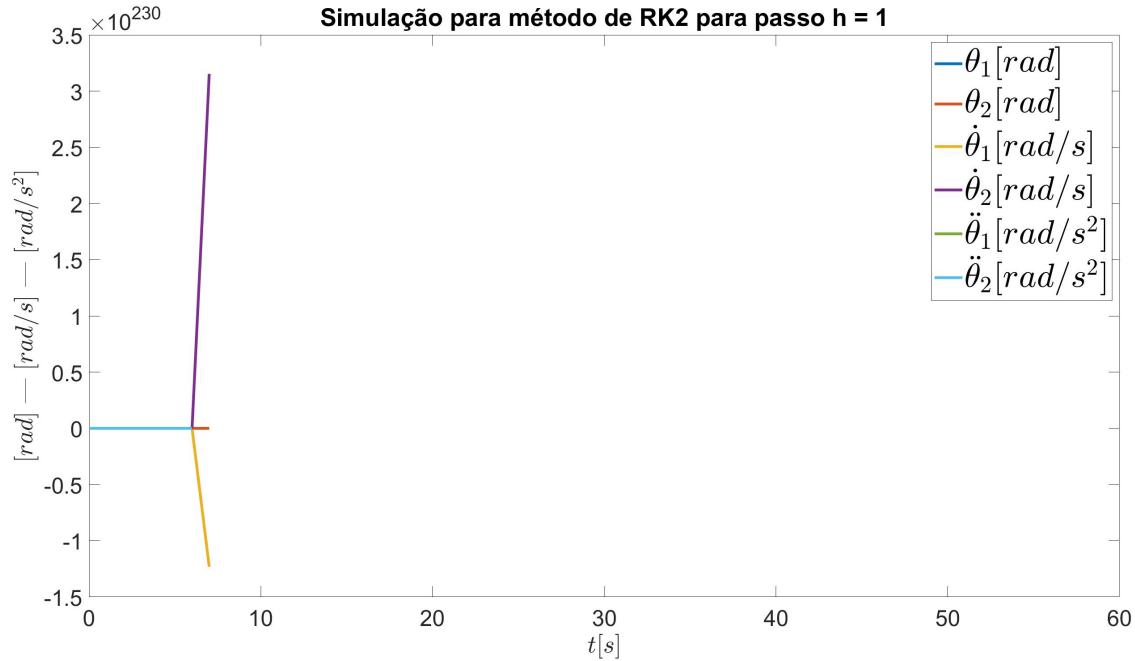


Figura 5: Método de Runge-Kutta de 2<sup>a</sup> ordem (passo  $h = 1$ )

Simulação para método de RK2 para passo  $h = 1$

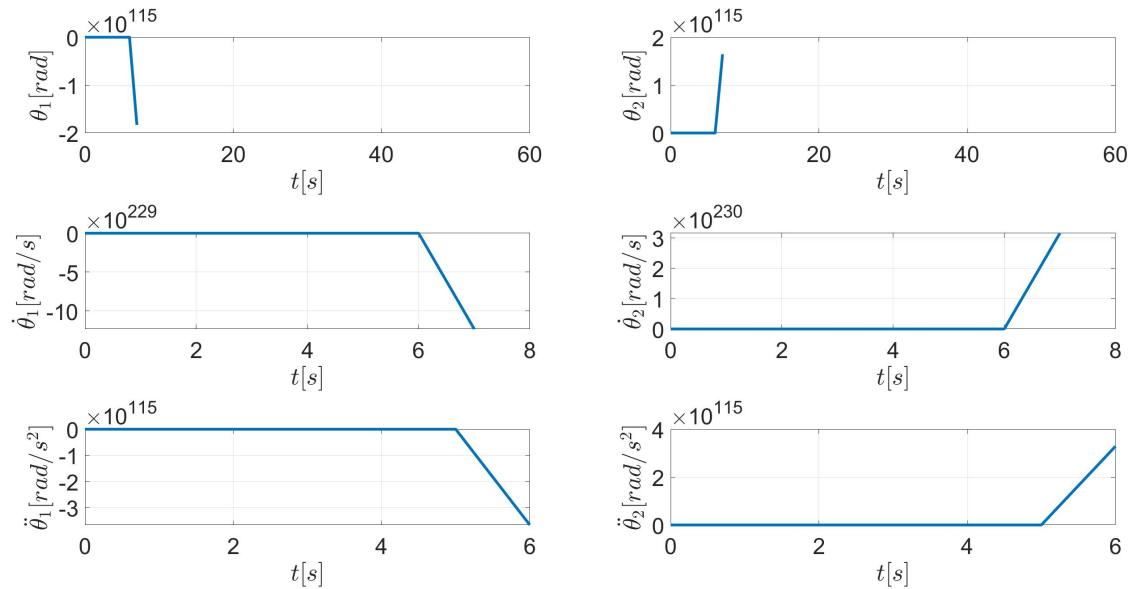


Figura 6: Método de Runge-Kutta de 2<sup>a</sup> ordem (passo  $h = 1$ ). Gráfico com subplots.

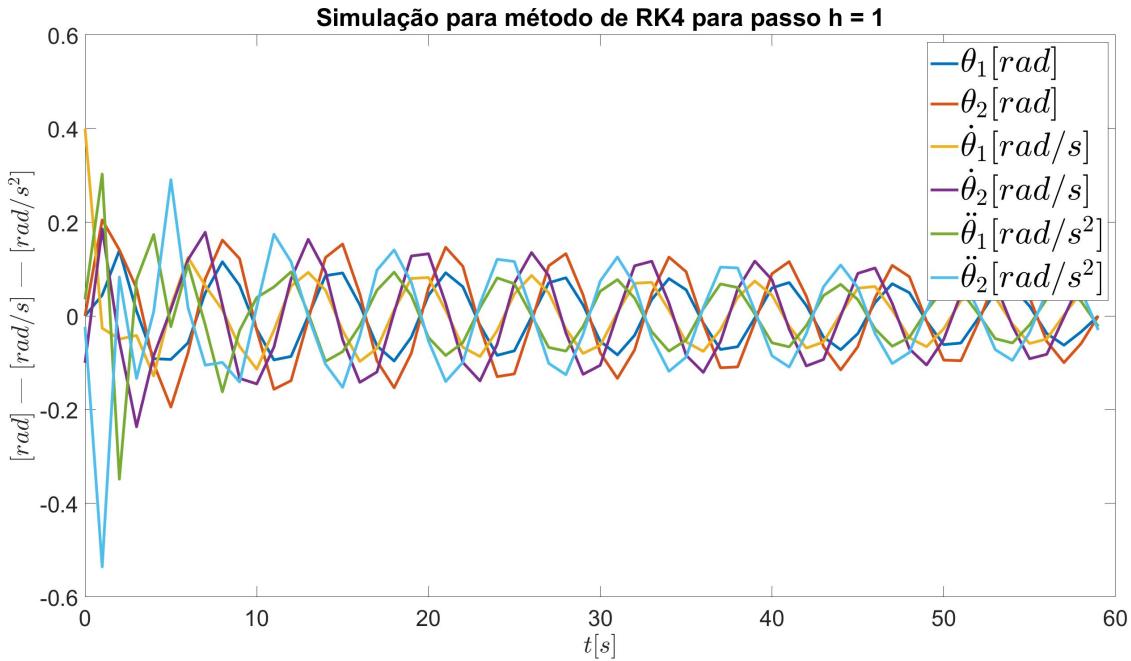


Figura 7: Método de Runge-Kutta de 4<sup>a</sup> ordem (passo  $h = 1$ )

Simulação para método de RK4 para passo  $h = 1$

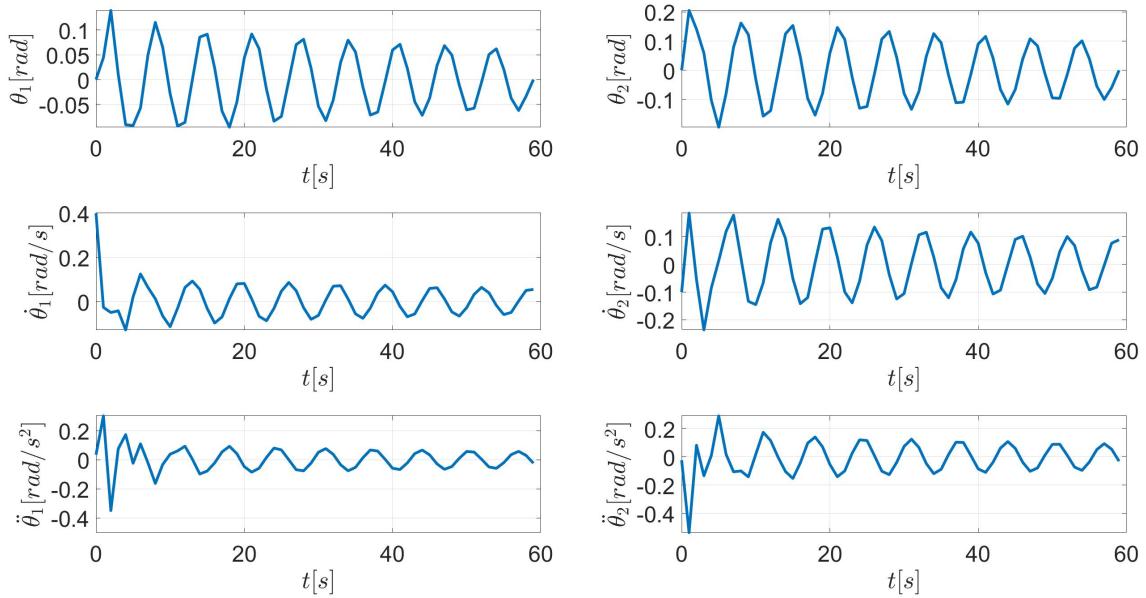


Figura 8: Método de Runge-Kutta de 4<sup>a</sup> ordem (passo  $h = 1$ ). Gráfico com subplots.

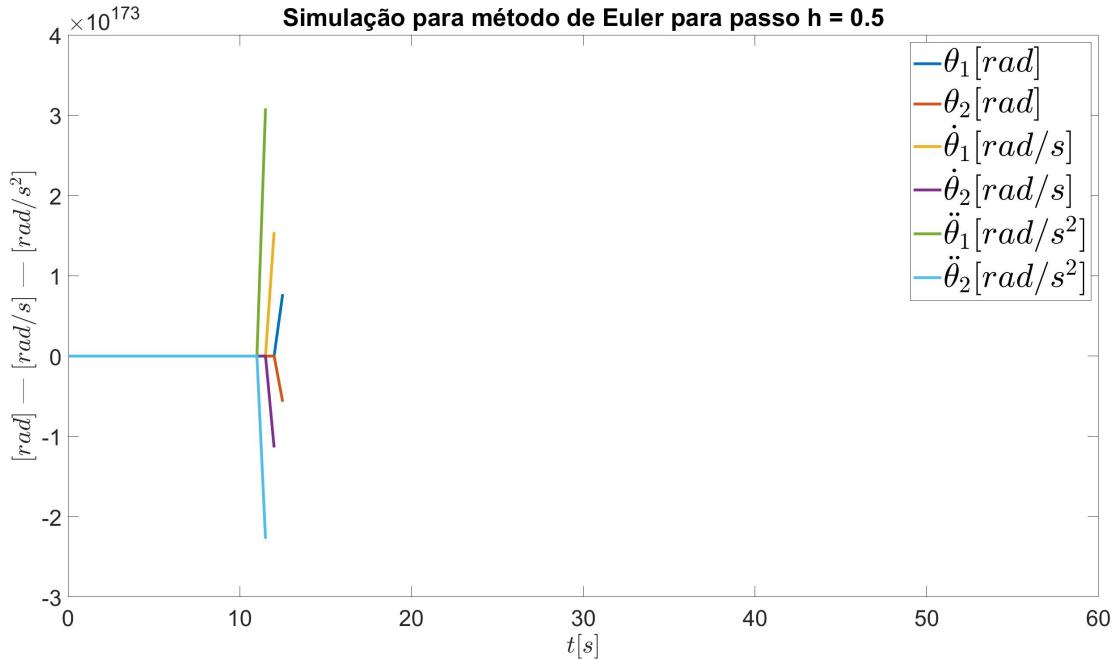


Figura 9: Método de Euler (passo  $h = 0,5$ )

Simulação para método de Euler para passo  $h = 0.5$

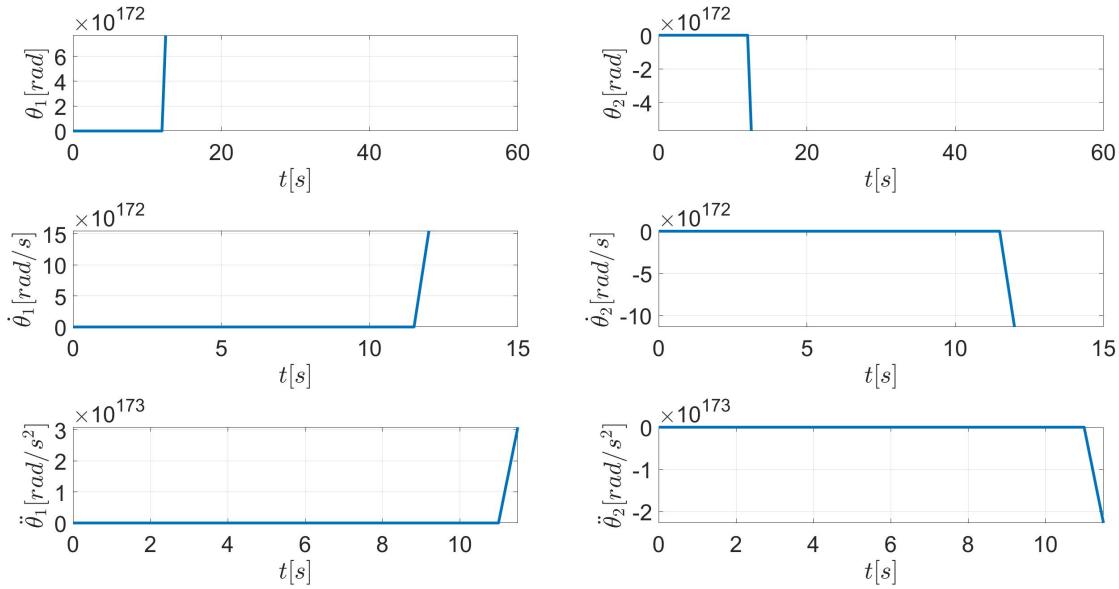


Figura 10: Método de Euler (passo  $h = 0,5$ ). Gráfico com subplots.

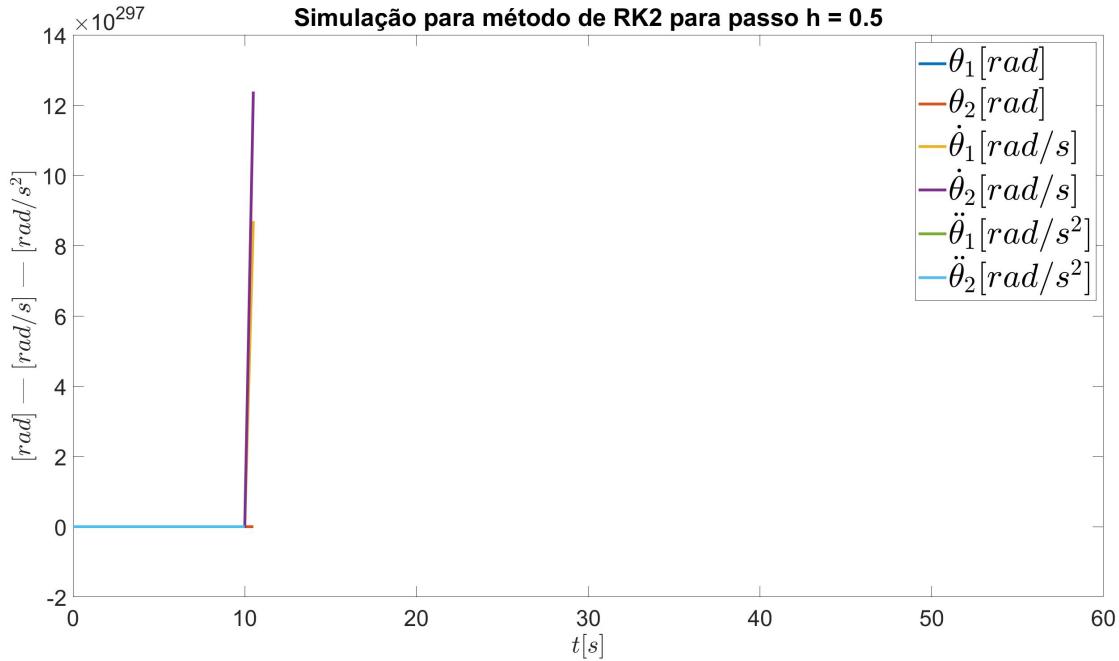


Figura 11: Método de Runge-Kutta de 2<sup>a</sup> ordem (passo  $h = 0,5$ )

Simulação para método de RK2 para passo  $h = 0.5$

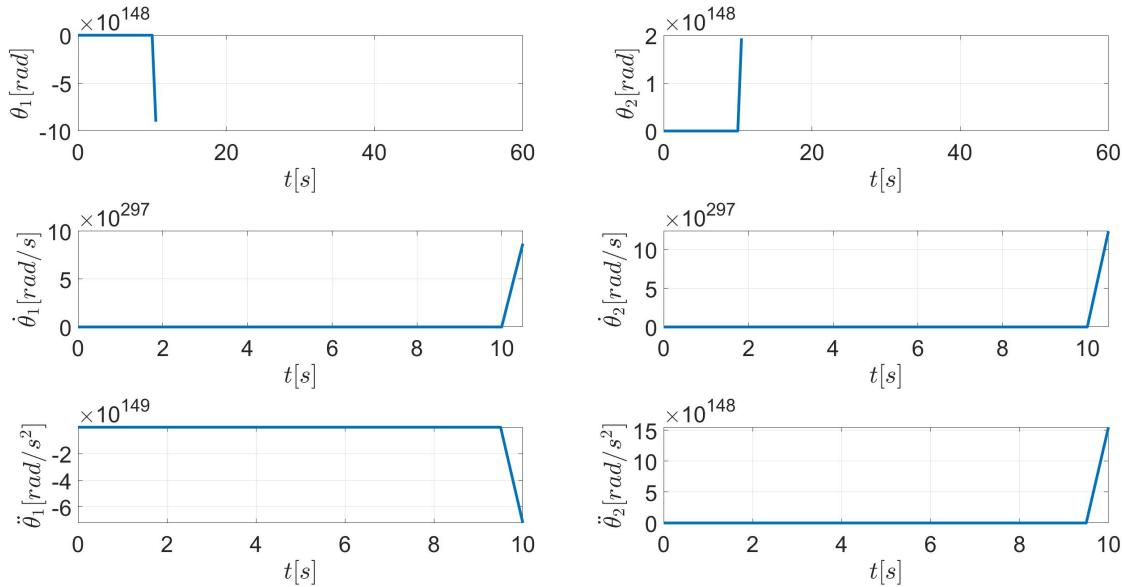


Figura 12: Método de Runge-Kutta de 2<sup>a</sup> ordem (passo  $h = 0,5$ ). Gráfico com subplots.

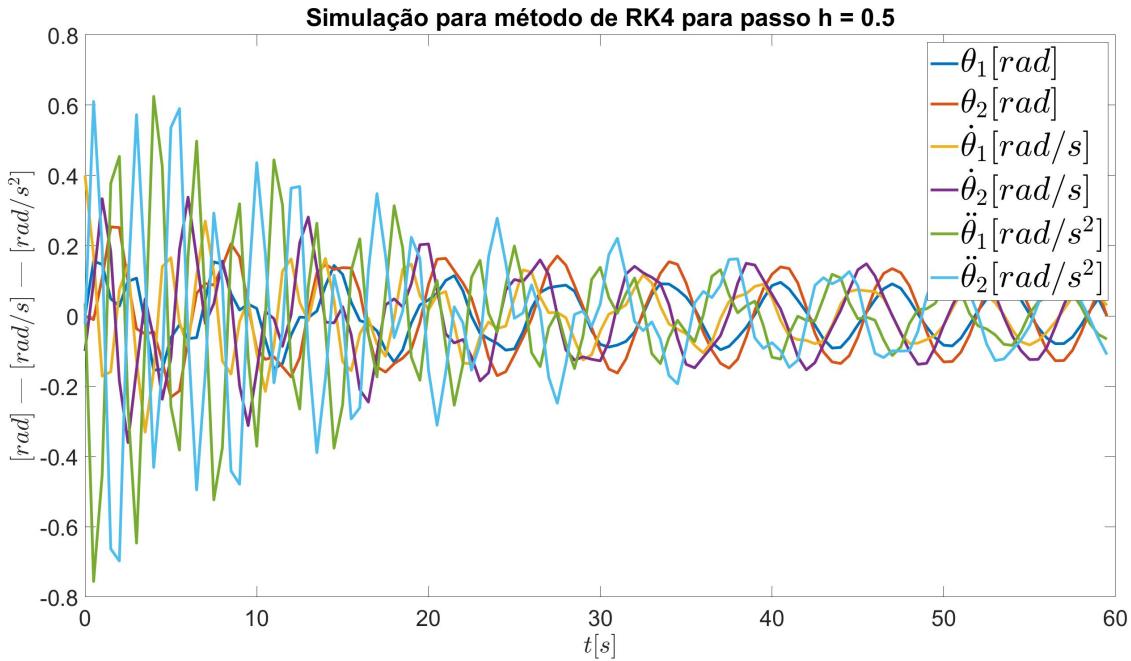


Figura 13: Método de Runge-Kutta de 4<sup>a</sup> ordem (passo  $h = 0,5$ )

Simulação para método de RK4 para passo  $h = 0.5$

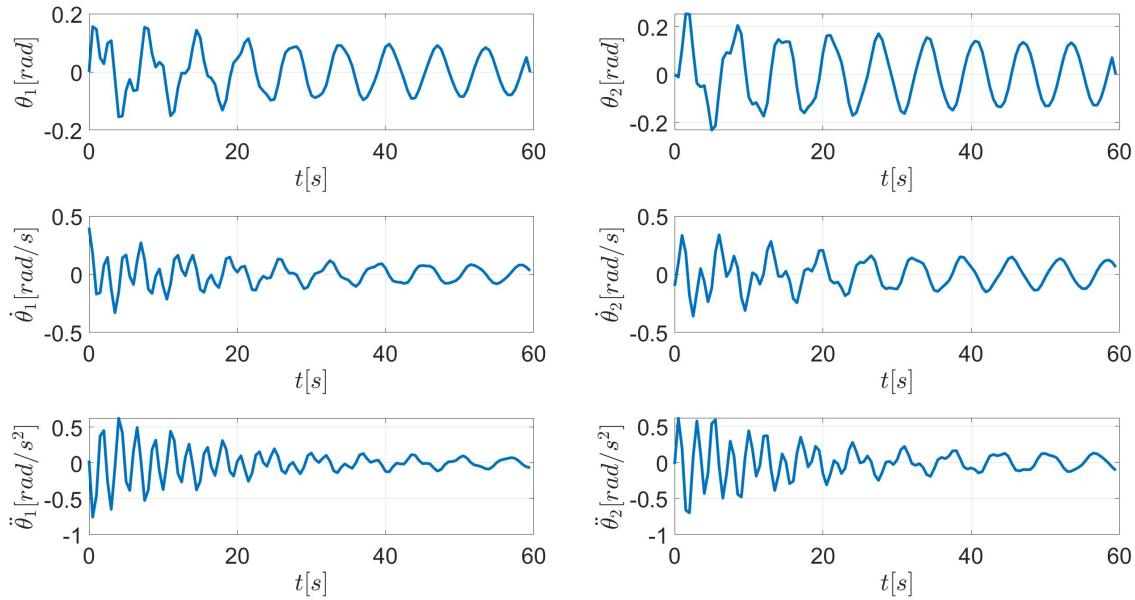


Figura 14: Método de Runge-Kutta de 4<sup>a</sup> ordem (passo  $h = 0,5$ ). Gráfico com subplots.

Simulação para método de Euler para passo  $h = 0.1$

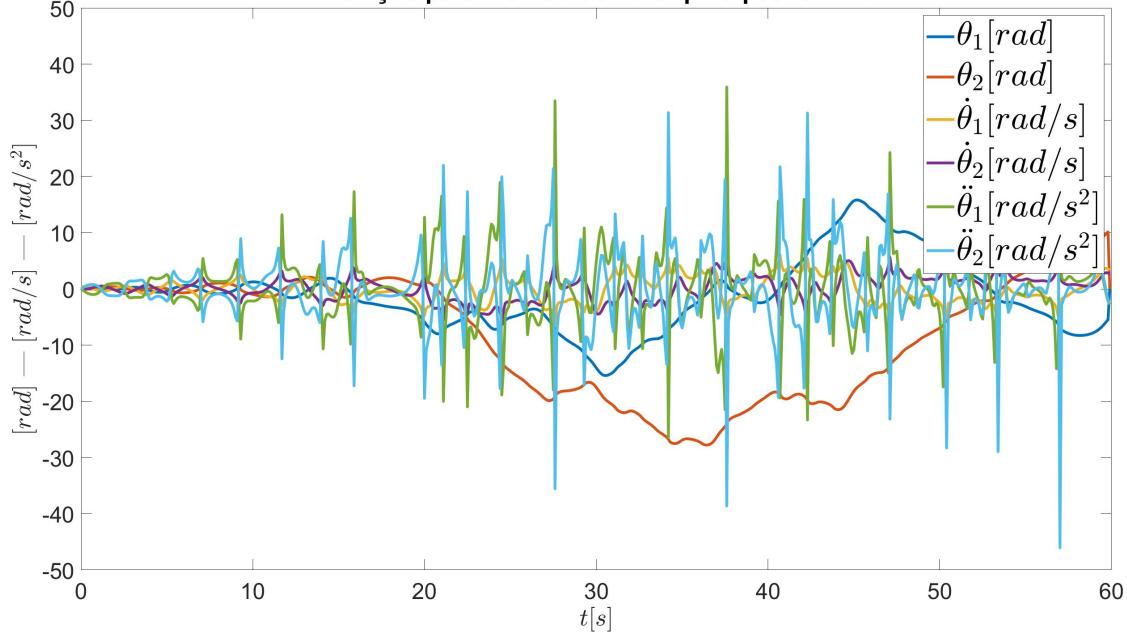


Figura 15: Método de Euler (passo  $h = 0,1$ )

Simulação para método de Euler para passo  $h = 0.1$

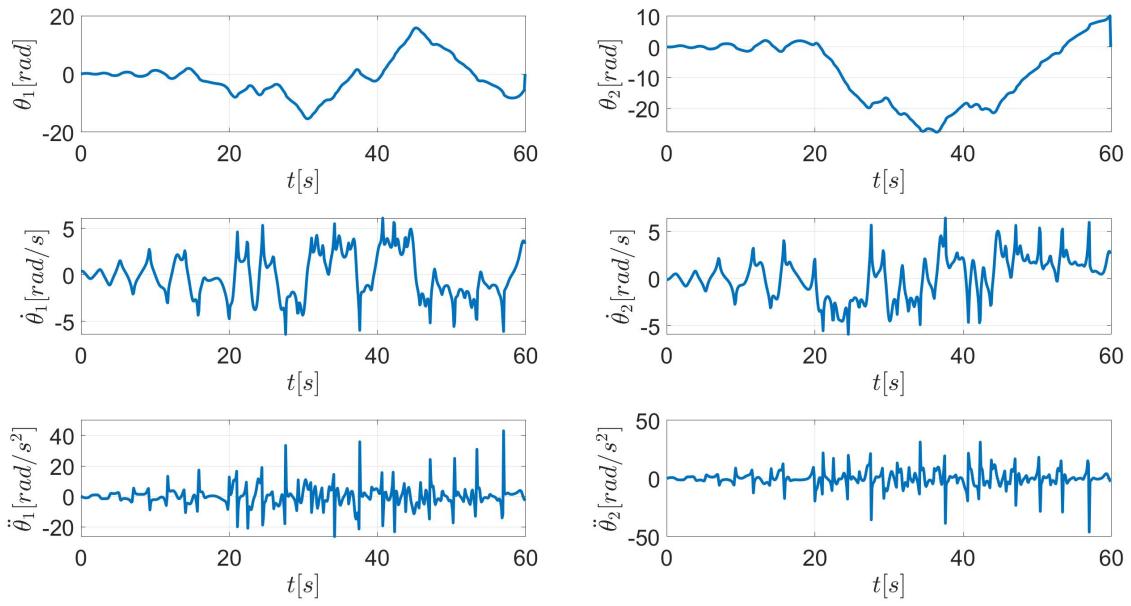


Figura 16: Método de Euler (passo  $h = 0,1$ ). Gráfico com subplots.

Simulação para método de RK2 para passo  $h = 0.1$

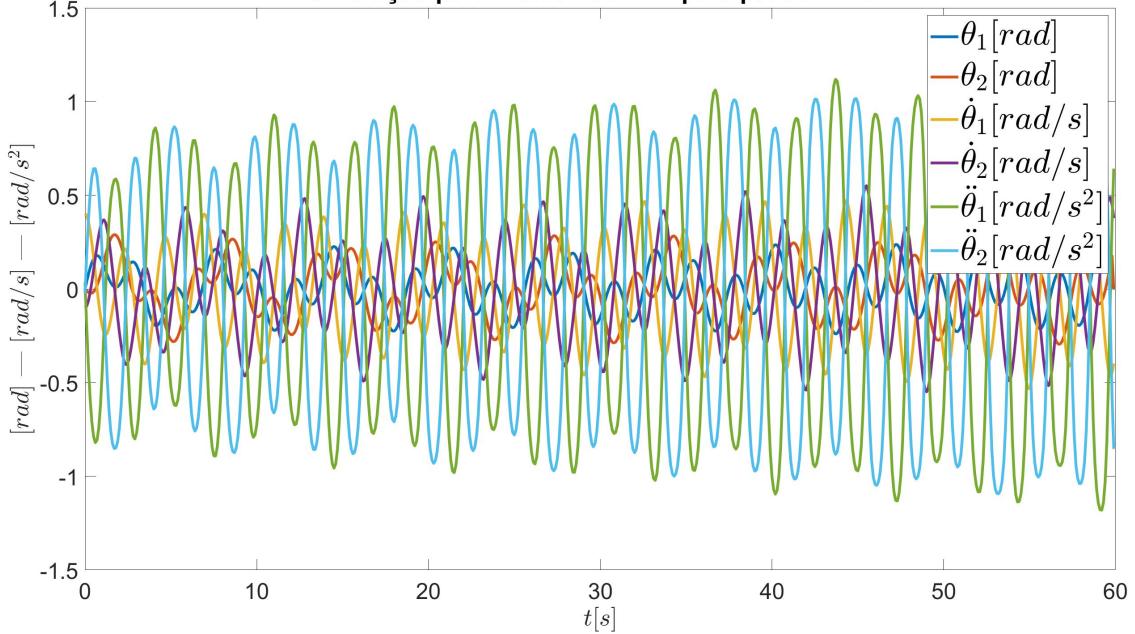


Figura 17: Método de Runge-Kutta de 2<sup>a</sup> ordem (passo  $h = 0,1$ )

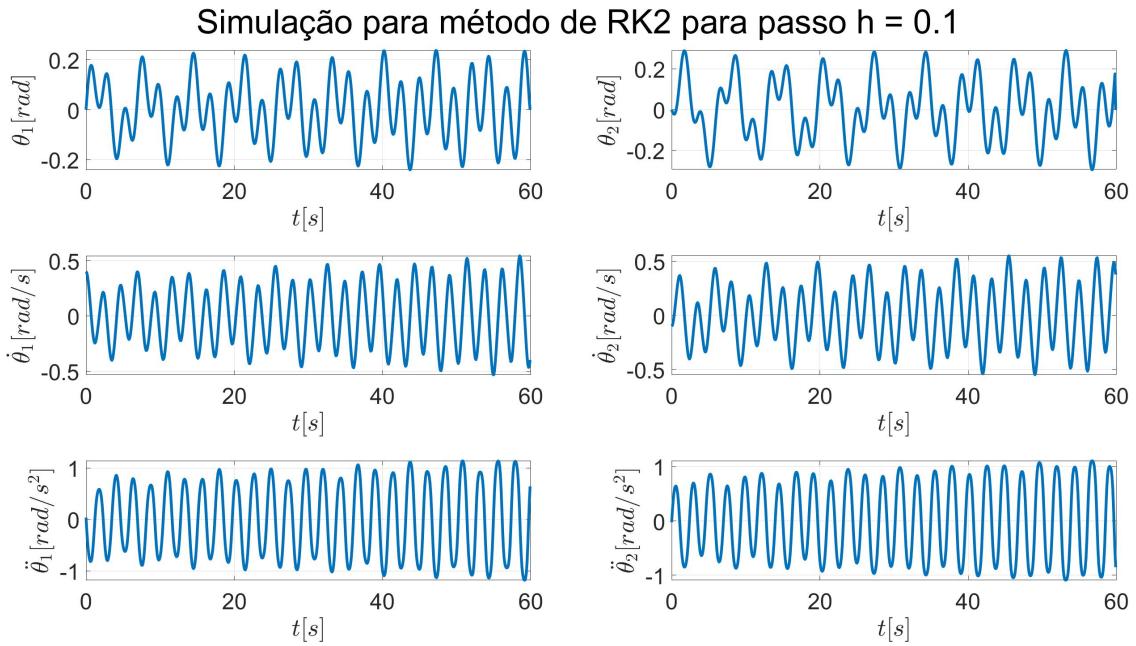


Figura 18: Método de Runge-Kutta de 2<sup>a</sup> ordem (passo  $h = 0,1$ ). Gráfico com subplots.

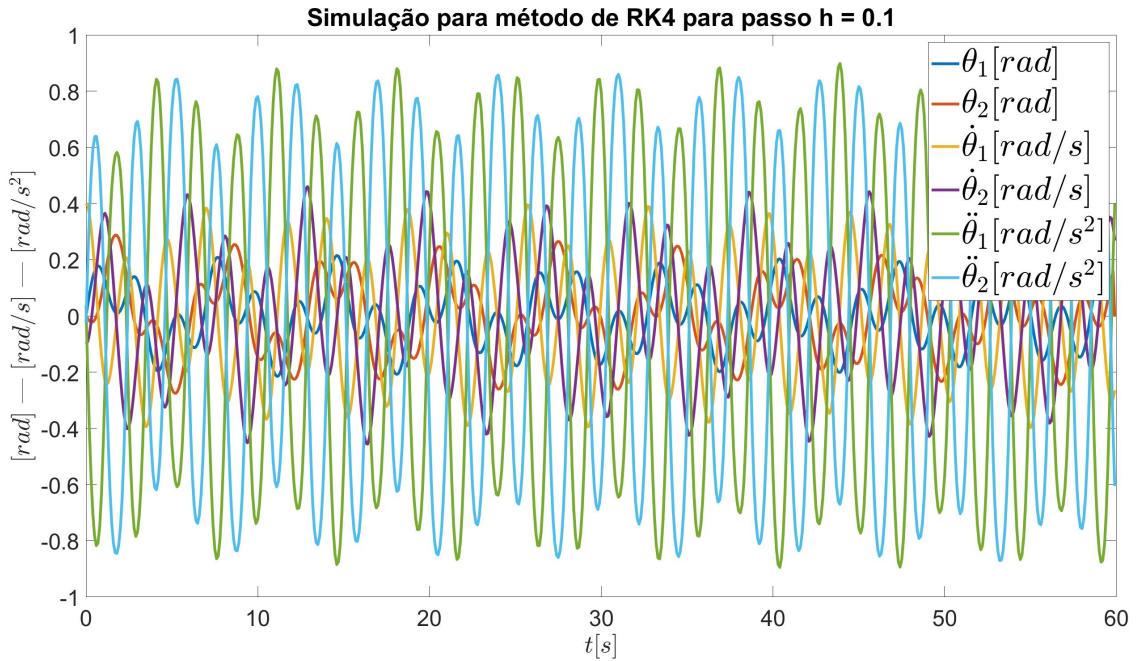


Figura 19: Método de Runge-Kutta de 4<sup>a</sup> ordem (passo  $h = 0,1$ )

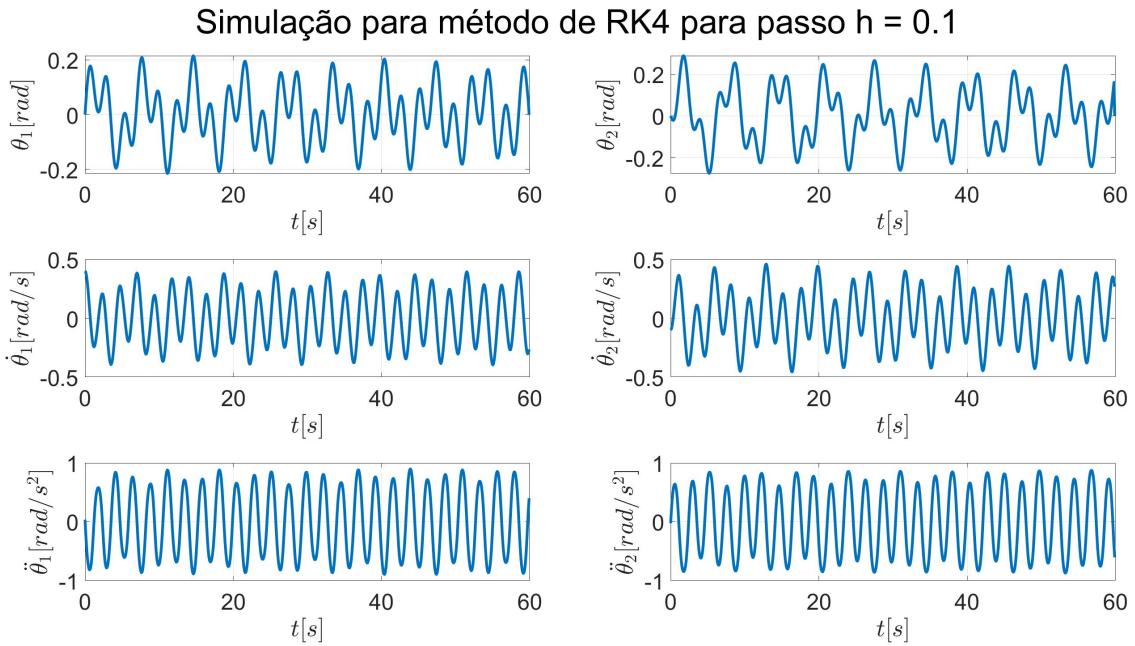


Figura 20: Método de Runge-Kutta de 4<sup>a</sup> ordem (passo  $h = 0,1$ ). Gráfico com subplots.

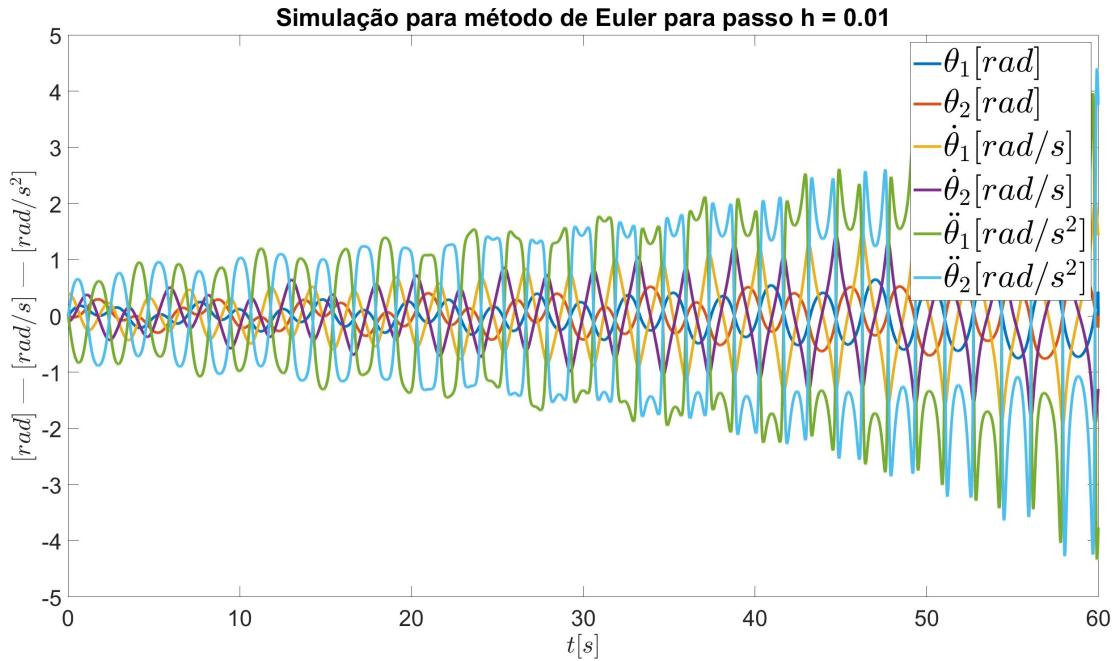


Figura 21: Método de Euler (passo  $h = 0,01$ )

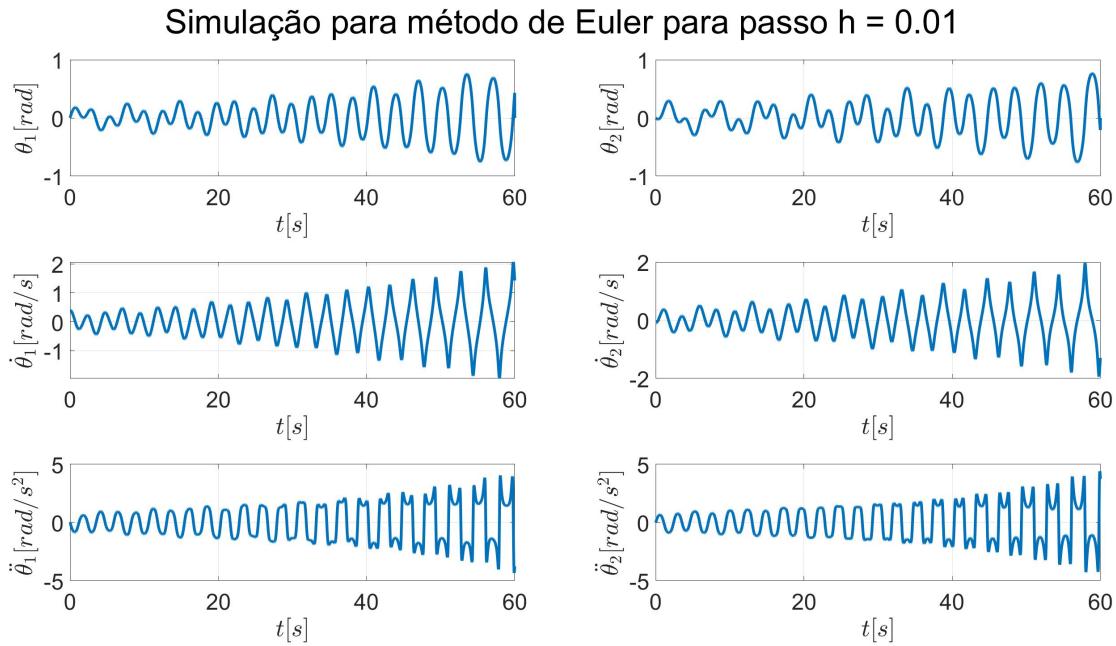


Figura 22: Método de Euler (passo  $h = 0,01$ ). Gráfico com subplots.

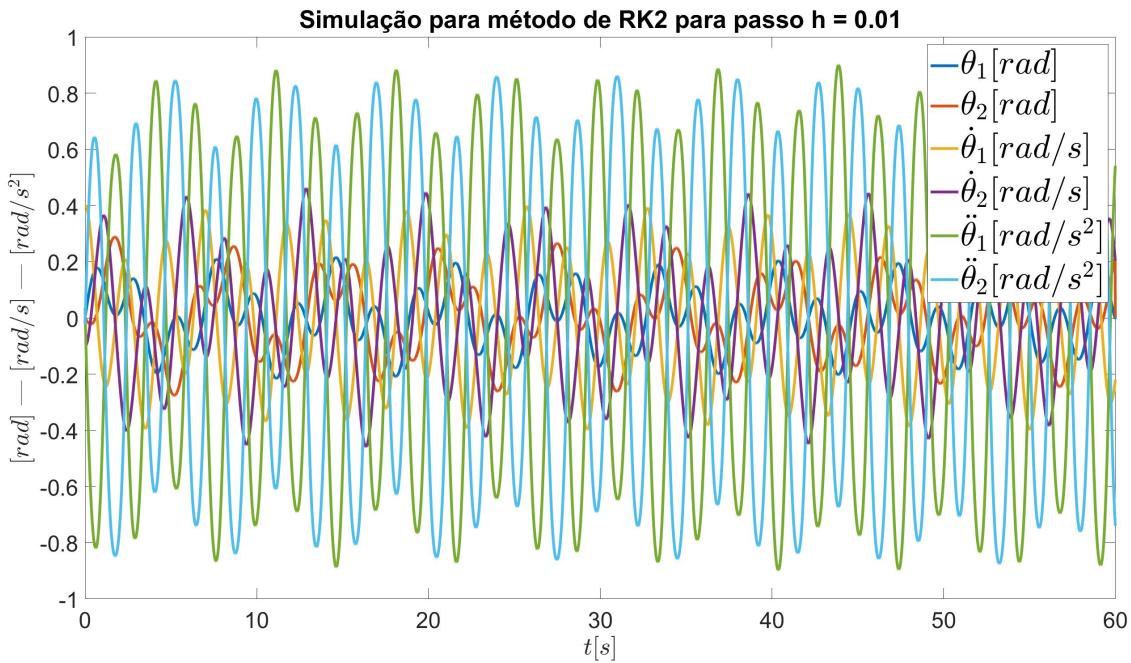


Figura 23: Método de Runge-Kutta de 2<sup>a</sup> ordem (passo  $h = 0,01$ )

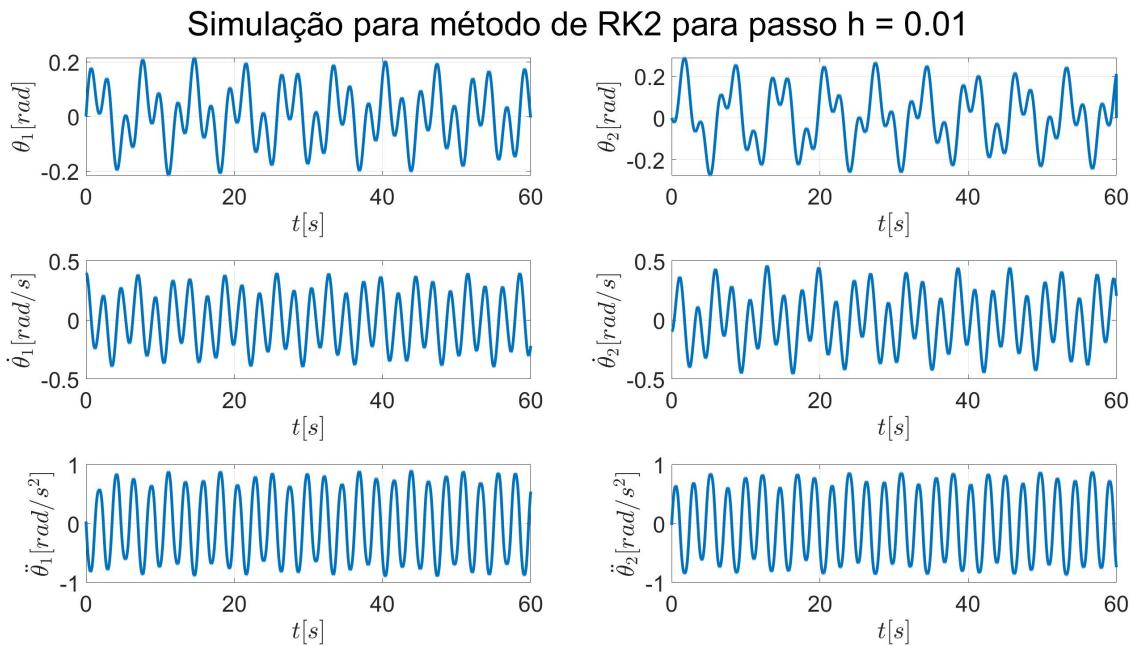


Figura 24: Método de Runge-Kutta de 2<sup>a</sup> ordem (passo  $h = 0,01$ ). Gráfico com subplots.

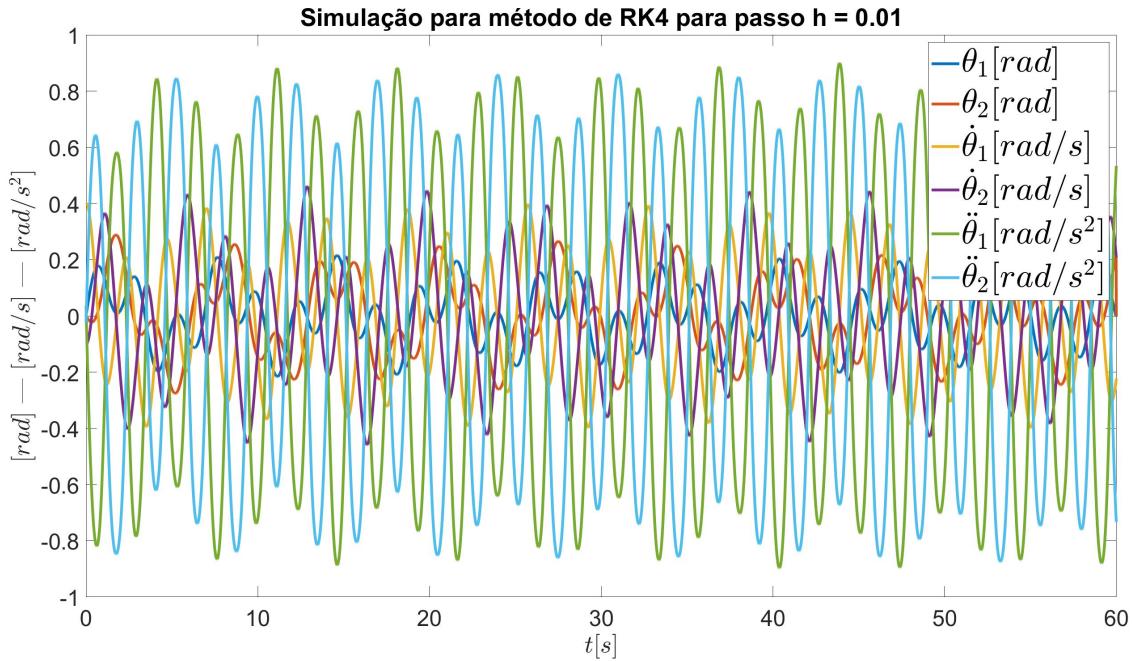


Figura 25: Método de Runge-Kutta de 4<sup>a</sup> ordem (passo  $h = 0,01$ )

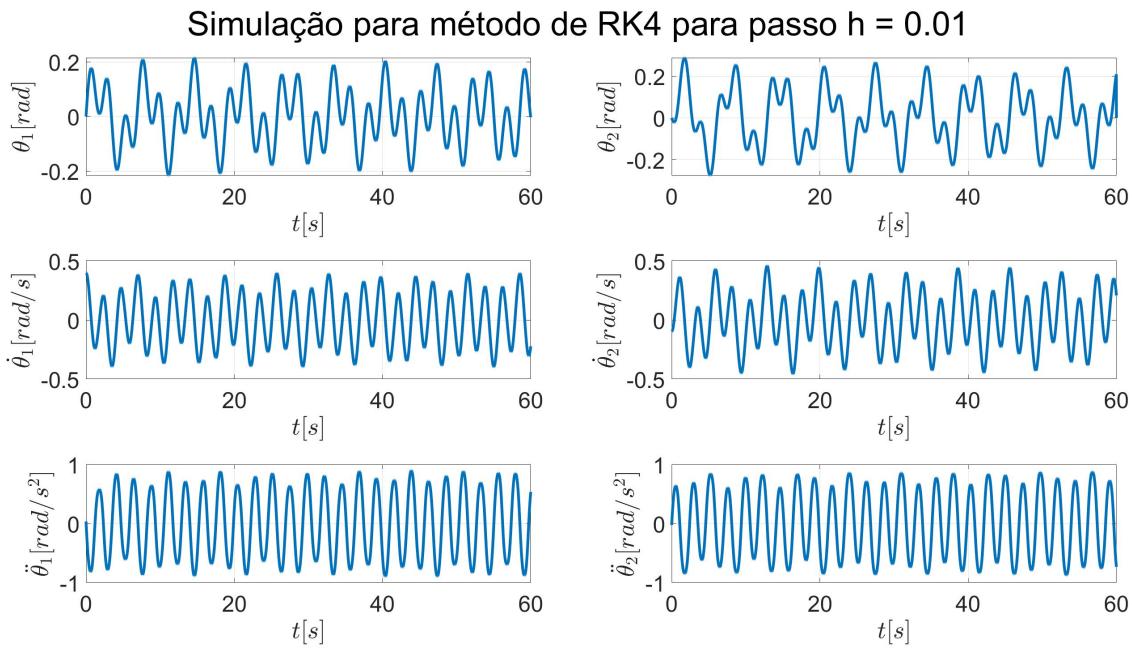


Figura 26: Método de Runge-Kutta de 4<sup>a</sup> ordem (passo  $h = 0,01$ ). Gráfico com subplots.

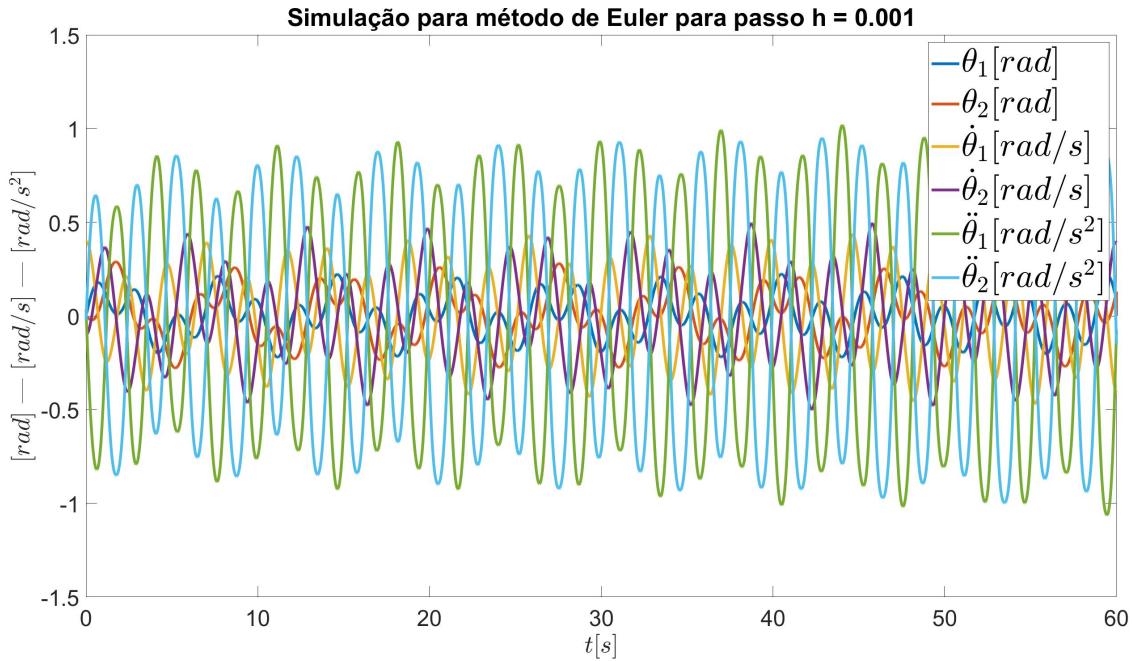


Figura 27: Método de Euler (passo  $h = 0,001$ )

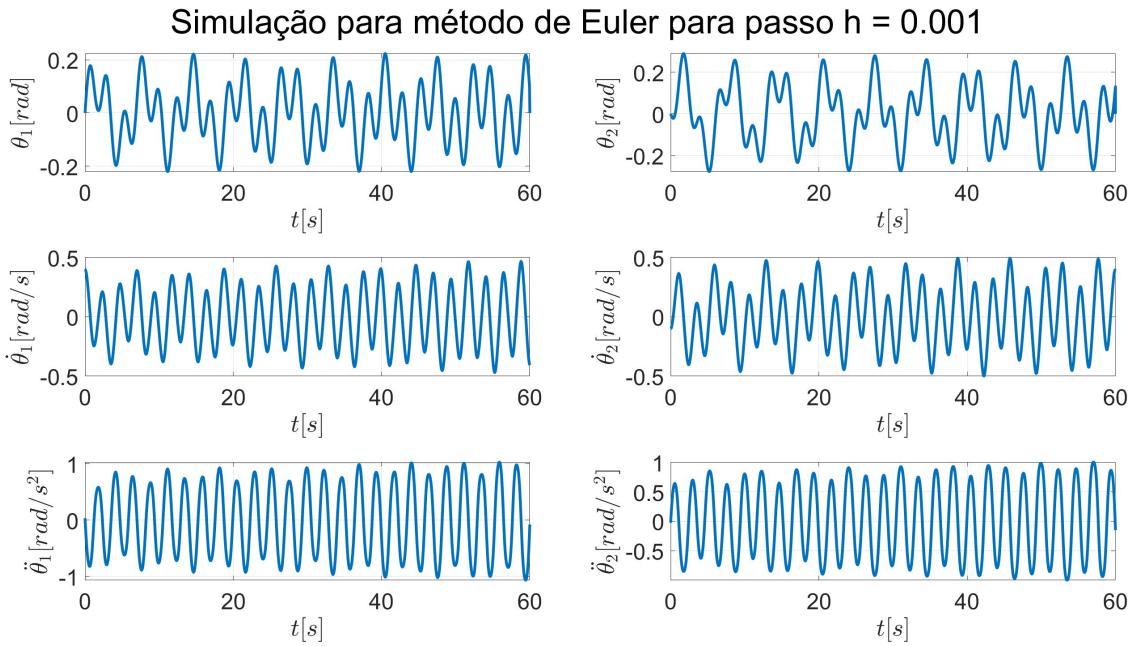


Figura 28: Método de Euler (passo  $h = 0,001$ ). Gráfico com subplots.

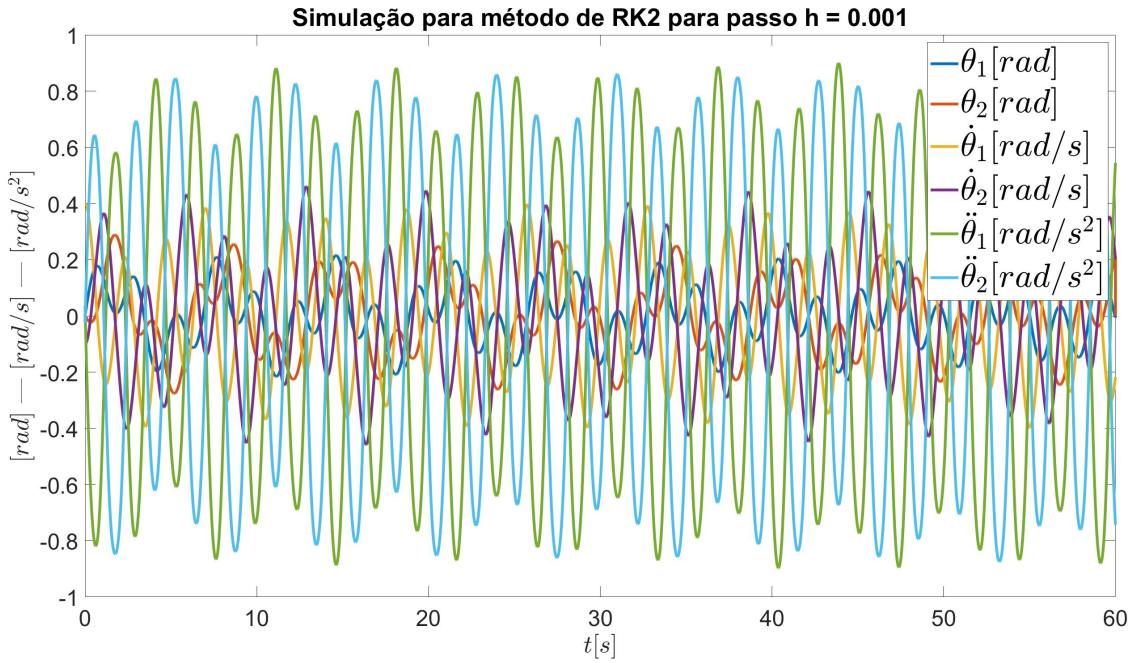


Figura 29: Método de Runge-Kutta de 2<sup>a</sup> ordem (passo  $h = 0,001$ )

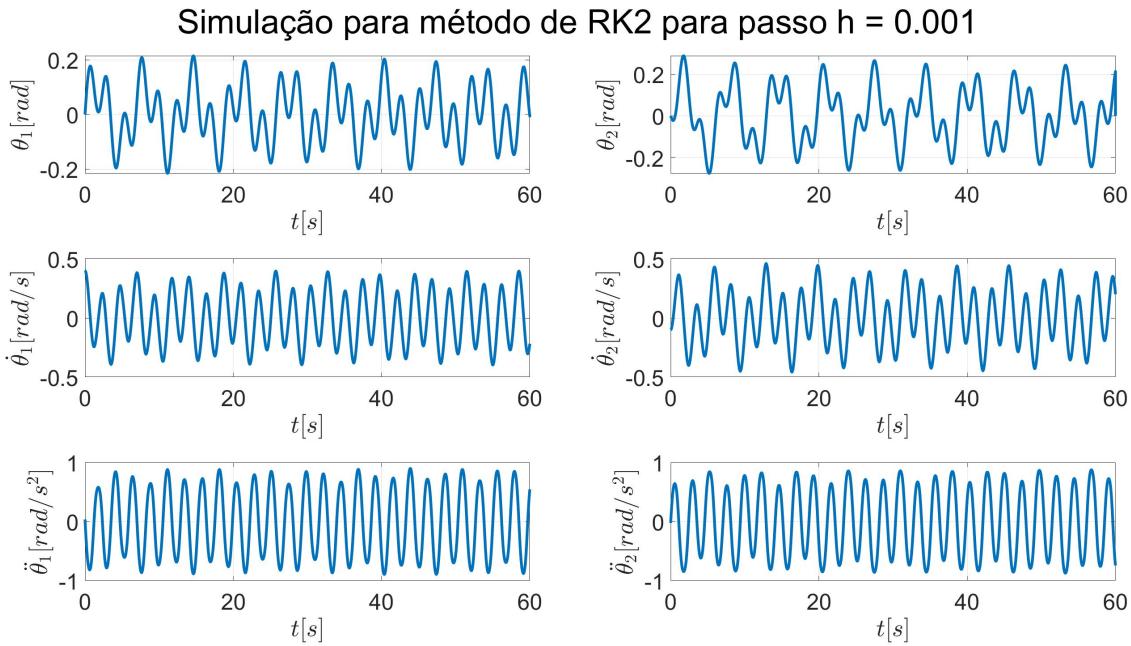


Figura 30: Método de Runge-Kutta de 2<sup>a</sup> ordem (passo  $h = 0,001$ ). Gráfico com subplots.

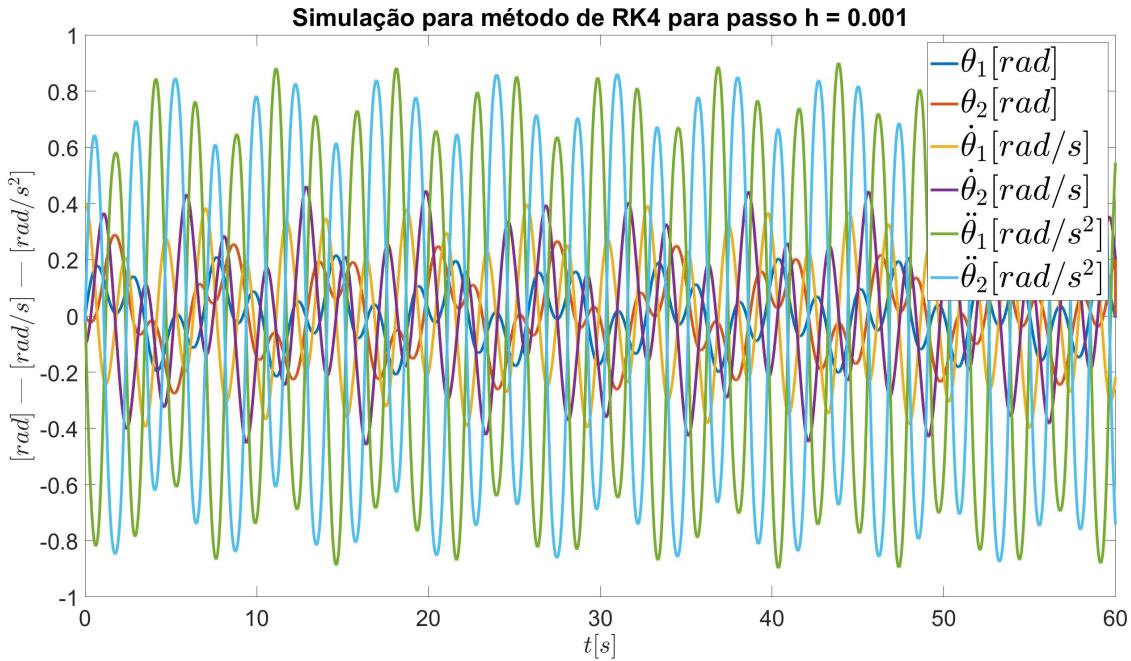


Figura 31: Método de Runge-Kutta de 4<sup>a</sup> ordem (passo  $h = 0,001$ )

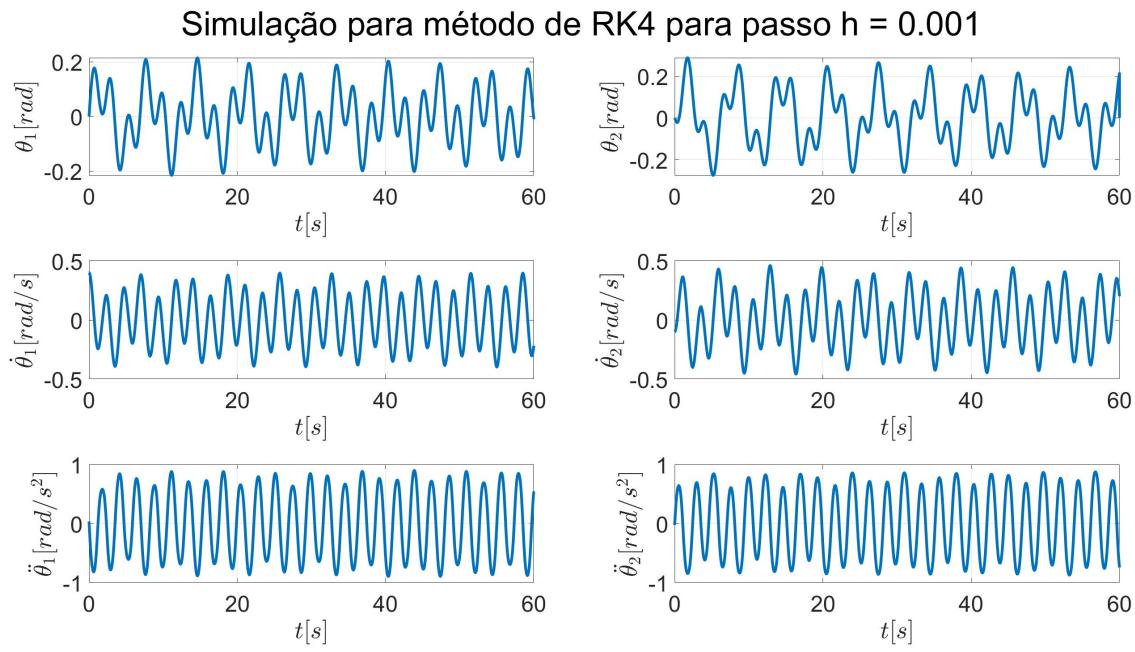


Figura 32: Método de Runge-Kutta de 4<sup>a</sup> ordem (passo  $h = 0,001$ ). Gráfico com subplots.

## 4.2 Discussões sobre a influência do passo

Percebe-se, pelos gráficos, que para passos maiores ( $h = 1$  e  $h = 0,5$ ), os métodos de Euler (figuras 3 e 9) e Runge-Kutta de 2<sup>a</sup> ordem (RK2)(figuras 5 e 11) divergem de uma solução de forma senoidal pois os valores das variáveis simuladas vão para o infinito, mas o método de Runge-Kutta de 4<sup>a</sup> ordem (RK4), por sua vez, (figuras 7 e 13) converge. Isto ocorre porque os erros global e local para o método de Euler são proporcionais a  $h$  e a  $h^2$ , respectivamente. Para o método de RK2 são proporcionais a  $h^2$  e a  $h^3$ . Enquanto isso, tais erros para o método de RK4 são proporcionais a  $h^4$  e a  $h^5$ , também respectivamente. À medida que formos diminuindo o passo, os métodos vão convergindo para a mesma solução já que os erros também vão exponencialmente diminuindo.

Nota-se assim que o método RK2 converge antes que o método de Euler (o método de RK2 converge para uma solução estável para  $0,1 \leq h \leq 0,5$  e o método de Euler apenas para o menor passo testado). Enfim, para um passo de  $h = 0,001$  temos que todos os métodos possuem solução bastante parecidas entre si (figuras 27, 29 e 31), comprovando graficamente que os erros de truncamento diminuem à medida que o passo utilizado é decrescido.

### 4.3 Perguntas

Para esta seção apenas foram consideradas simulações com o passo  $h = 0.001$ , visto que é o menor passo encontrado que conseguimos estabilidades para todos os métodos acima de resolução de equações diferenciais.

- Suponha que há um aumento da carga  $m_2$  para 1000 kg e analise o que ocorre com relação a estabilidade do veículo.

Quando aumentamos a carga  $m_2$  para 1000kg, mantendo as mesmas condições anteriores, as oscilações possuem uma amplitude similar à situação com  $m_2 = 650\text{kg}$  e vão aumentando ligeiramente ao longo do tempo como observado nas figuras 33 e 34.

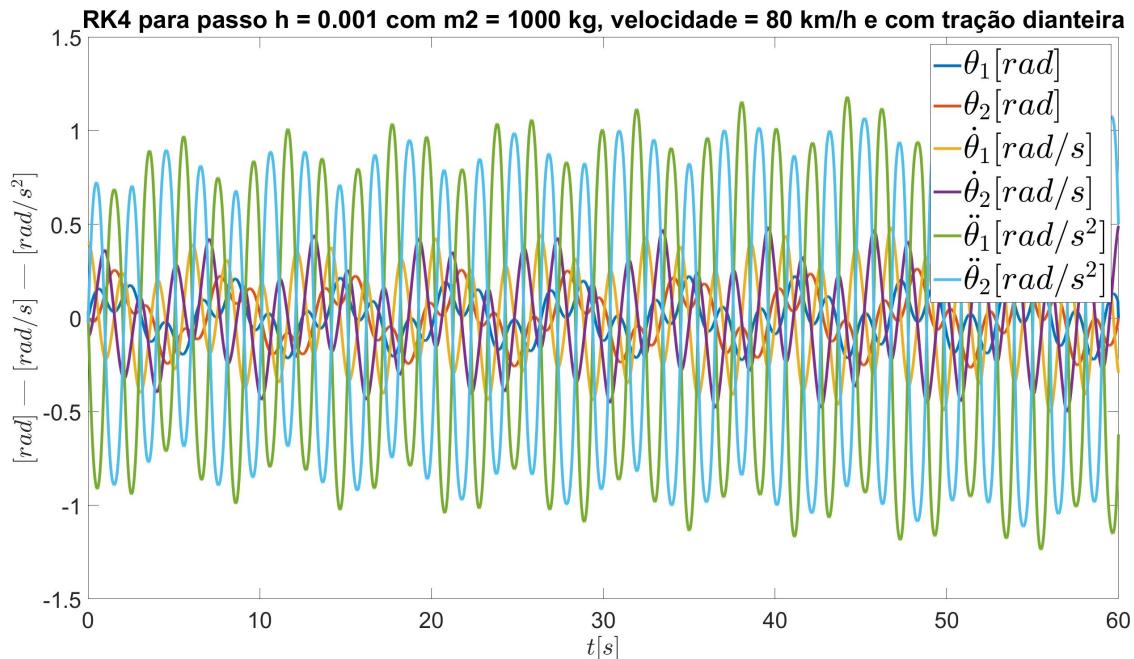


Figura 33: Método de Runge-Kutta de 4<sup>a</sup> ordem (passo  $h = 0,001$ ),  $m_2 = 1000\text{kg}$ .

RK4 passo  $h = 0.001$  com  $m_2 = 1000$  kg, velocidade = 80 km/h e com tração dianteira

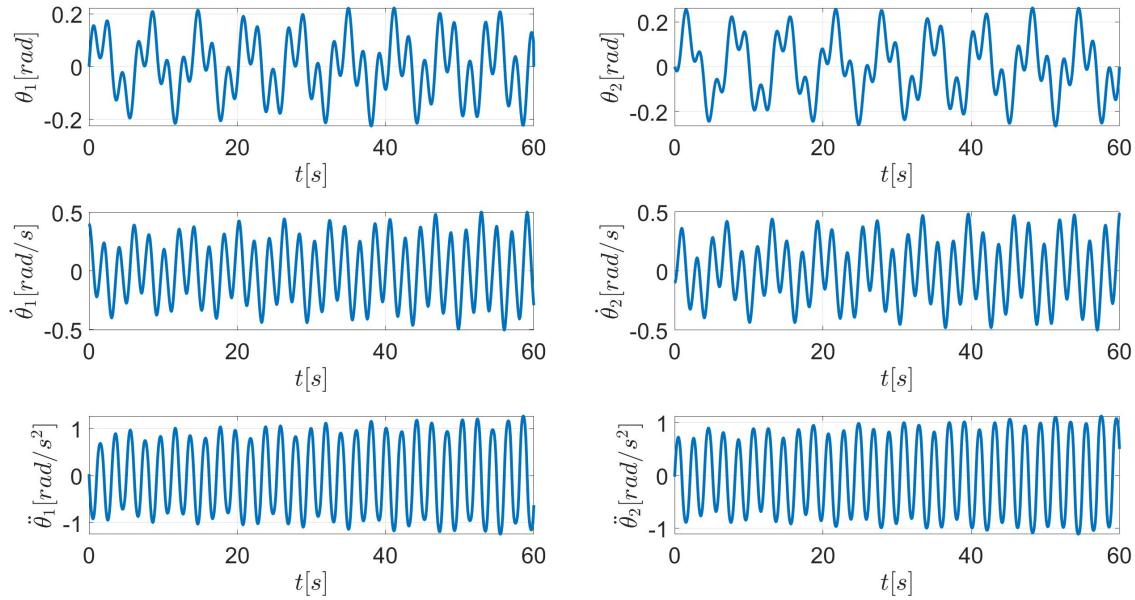


Figura 34: Método de Runge-Kutta de 4<sup>a</sup> ordem (passo  $h = 0,001$ ),  $m_2 = 1000\text{kg}$ . Gráfico com subplots.

Conclui-se pelas imagens que o aumento da carga  $m_2$  acarreta em um aumento da instabilidade do sistema, já que é um fator que multiplica  $A_0, A_1, B_0$  e  $B_1$ .

- Suponha que há uma redução da carga  $m_2$  para 200 kg e analise o que ocorre com relação a estabilidade do veículo.

Se reduzirmos a carga  $m_2$  para 200kg as oscilações ficam menores e diminuem ao longo do tempo, como visto na figura 35, indicando estabilidade do veículo. Assim, analisando a influência da variação da massa  $m_2$  vemos que deve-se torná-la o mais leve possível para que o carro oscile em menor amplitude.

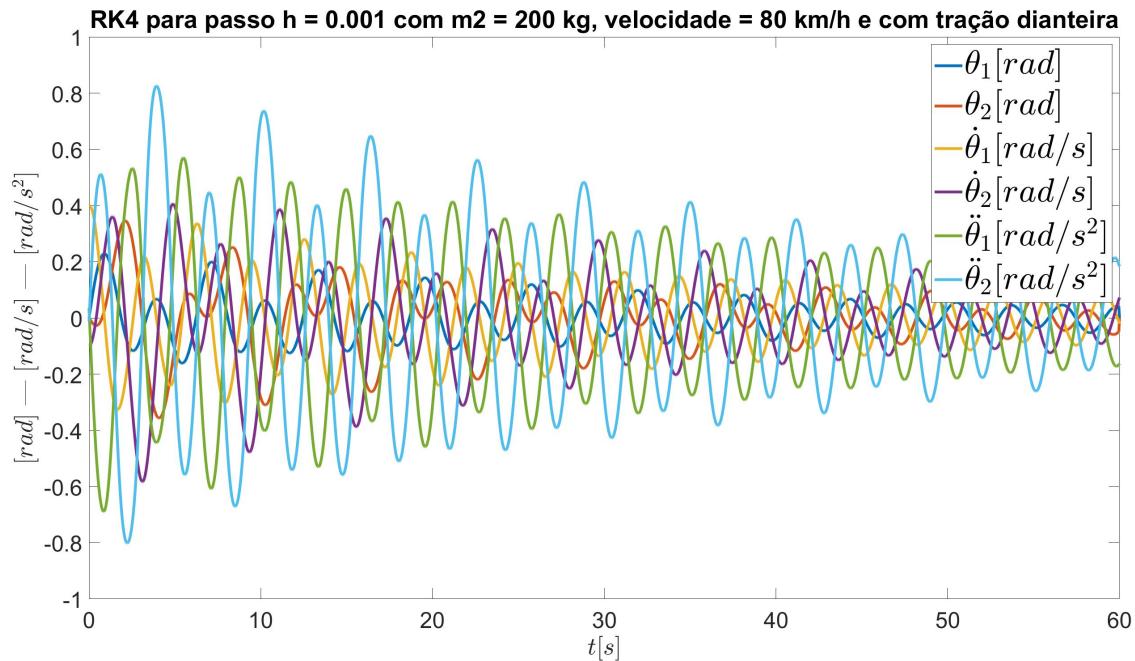


Figura 35: Método de Runge-Kutta de 4<sup>a</sup> ordem (passo  $h = 0,001$ ),  $m_2 = 200\text{kg}$ .

RK4 passo  $h = 0.001$  com  $m_2 = 200$  kg, velocidade = 80 km/h e com tração dianteira

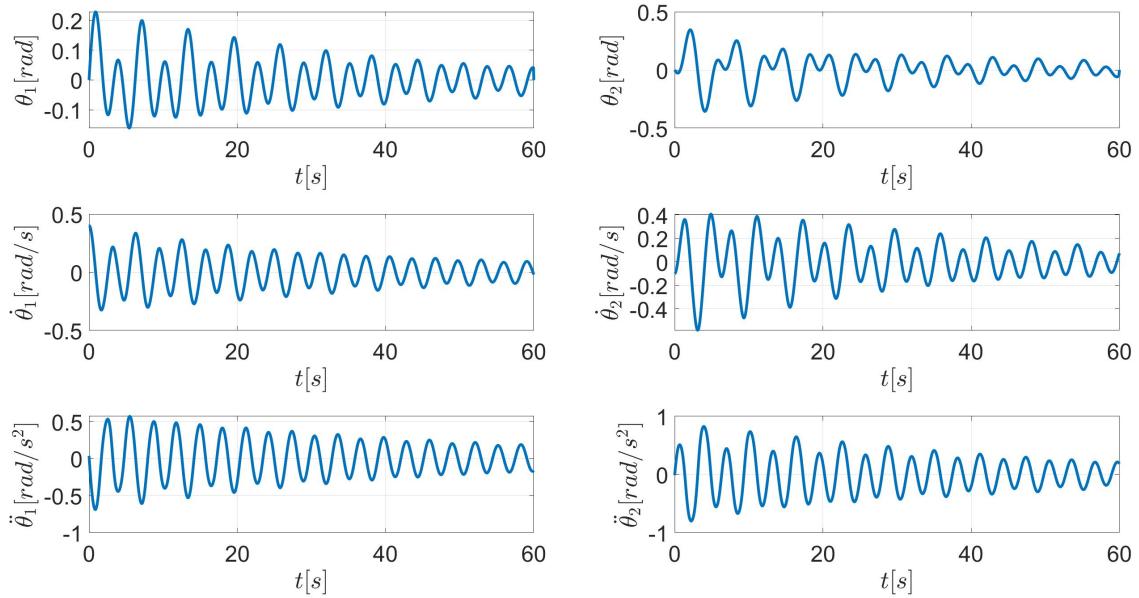


Figura 36: Método de Runge-Kutta de 4<sup>a</sup> ordem (passo  $h = 0,001$ ),  $m_2 = 200\text{kg}$ . Gráfico com subplots.

- Suponha que o motorista esteja a  $\text{Vel} = 120\text{km/h}$  e analise o que ocorre com relação a estabilidade do veículo. (Demais condições iguais às do item 1)

Ao aumentar a velocidade do veículo mantendo as condições iniciais, não são observadas grandes diferenças do comportamento em relação ao carro com velocidade de 80km/h. Portanto, infere-se que o parâmetro em questão não influencie bastante o sistema, já que os items que **Vel** influenciam são  $\dot{\theta}_1$  e  $\dot{\theta}_2$ , que não chegam a grandes valores (como pode-se ver na figura 38, ambos  $\dot{\theta}_1$  e  $\dot{\theta}_2$  oscilam entre aproximadamente 0,5 e -0,5  $\frac{\text{rad}}{\text{s}}$ ). Ao aumentar Vel em 50% tem-se que  $A_3$  diminui de -300 para  $-450 \frac{\text{kgm}^4}{\text{s}}$  e  $B_3$  de -60 para  $-90 \frac{\text{kgm}^3}{\text{s}}$ , o que perto de outras constantes que multiplicam variáveis não é muito relevante (por exemplo,  $A_0$  em seu menor valor [ $\theta_1 = \theta_2$ ] vale  $-2700 \text{ kgm}^3$ , e mesmo que  $\ddot{\theta}_1$  apenas oscile de 0,2 a  $-0,2 \frac{\text{rad}}{\text{s}^2}$  isso já equivale a valores muito mais significativos do que os causados por  $\text{Vel} = 120 \frac{\text{km}}{\text{h}}$ ).

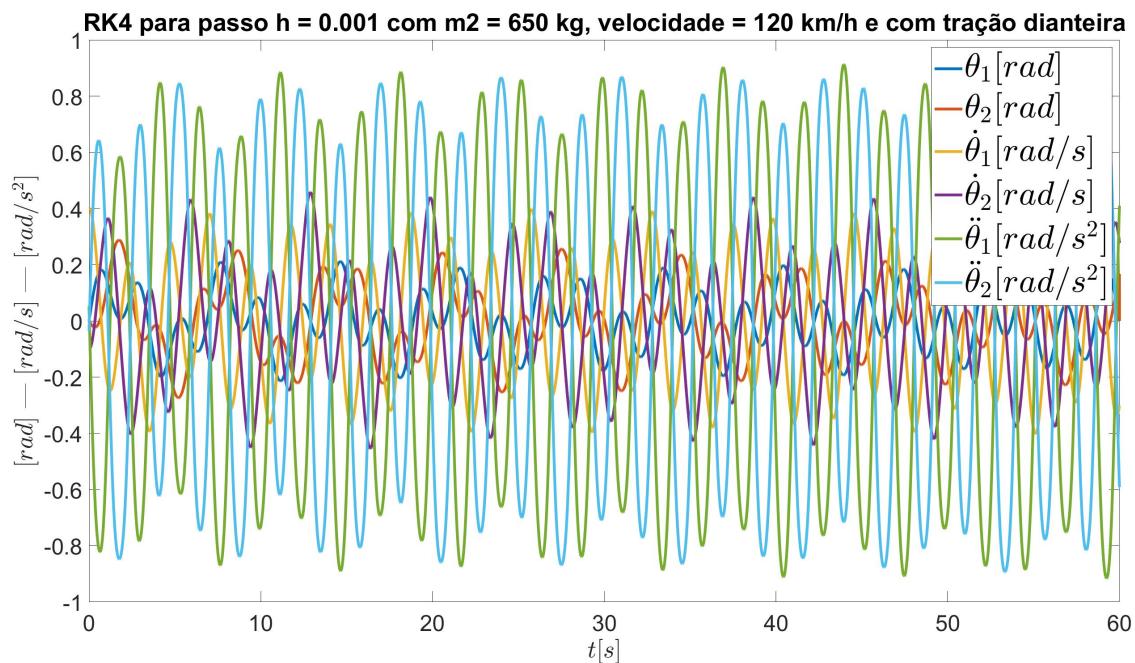


Figura 37: Método de Runge-Kutta de 4<sup>a</sup> ordem (passo  $h = 0,001$ ),  $\text{Vel} = 120\text{km/h}$ .

RK4 passo  $h = 0.001$  com  $m_2 = 650$  kg, velocidade = 120 km/h e com tração dianteira

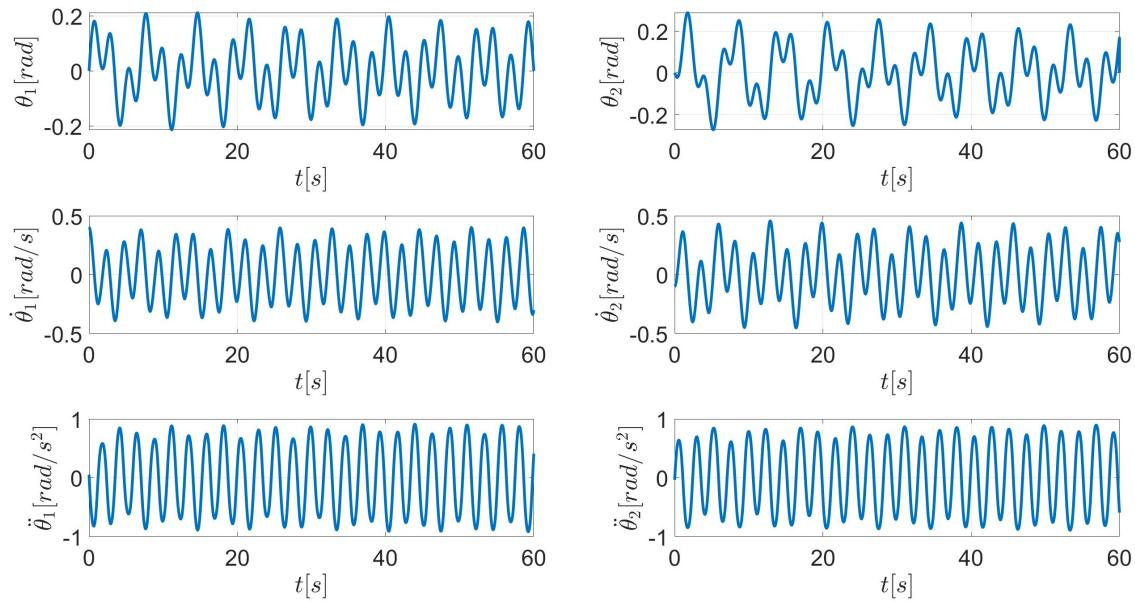


Figura 38: Método de Runge-Kutta de 4<sup>a</sup> ordem (passo  $h = 0,001$ ), Vel = 120km/h. Gráfico com subplots.

- Suponha que o veículo tenha tração traseira impondo  $F_1 = +0,5*m_1*gN$  e analise o que ocorre com relação a estabilidade do veículo. (Demais condições iguais às do item 1)

Adicionando tração traseira ao veículo observa-se uma perda de estabilidade, ilustrada pelas amplitudes das curvas, principalmente de  $\theta_1$  e  $\theta_2$ . Devido ao crescimento contínuo dessas variáveis percebe-se que a situação mais possível é a de que o veículo em questão tenha tombado. Enfim, conclui-se que a tração traseira prejudica a estabilidade do veículo.

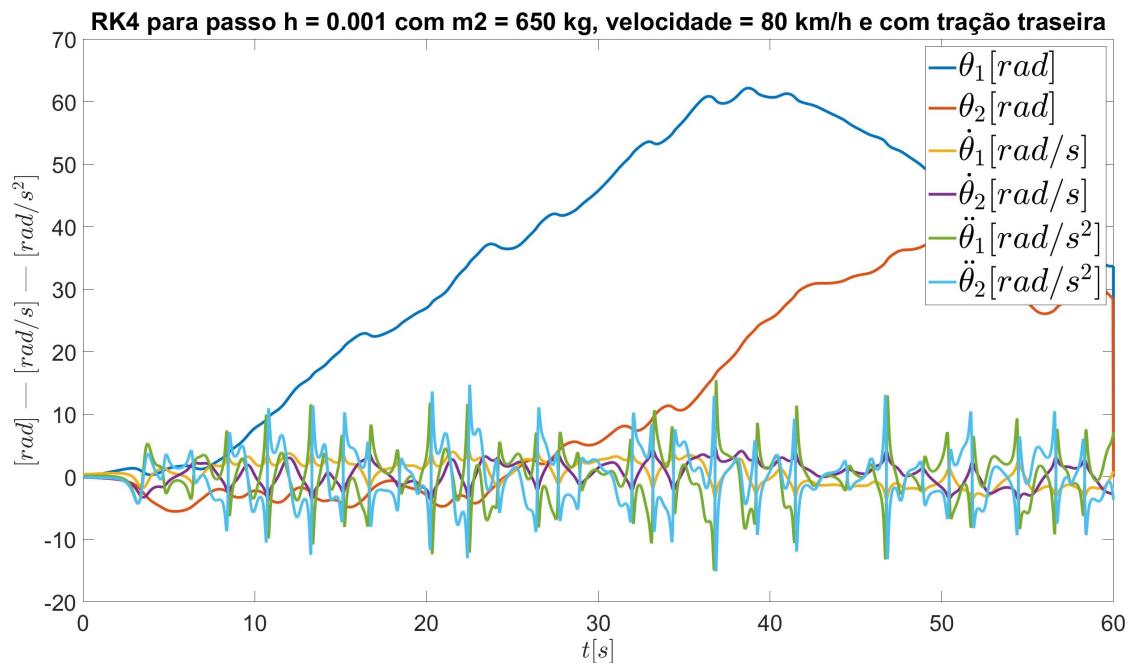


Figura 39: Método de Runge-Kutta de 4<sup>a</sup> ordem (passo  $h = 0,001$ ), tração traseira.

RK4 passo  $h = 0.001$  com  $m_2 = 650$  kg, velocidade = 80 km/h e com tração traseira

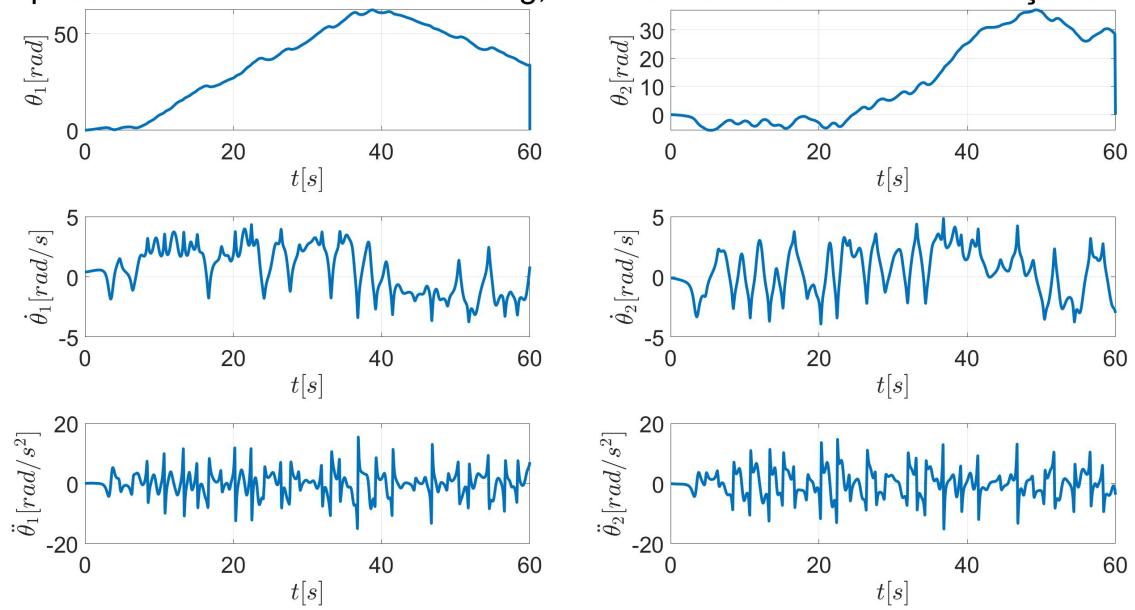


Figura 40: Método de Runge-Kutta de 4<sup>a</sup> ordem (passo  $h = 0,001$ ), tração traseira. Gráfico com subplots.

## 5 Conclusões

Foi possível criar, no MATLAB, três programas diferentes que simulassem os métodos de Euler, Runge-Kutta de 2<sup>a</sup> ordem e Runge-Kutta de 4<sup>a</sup> ordem para resolução numérica das equações diferenciais propostas no problema, além da função contendo as equações a serem resolvidas. Por meio de gráficos foi possível verificar a influência do passo  $h$  para cinco valores diferentes (1, 0.5, 0.1, 0.01 e 0.001) e perceber a partir de qual intervalo do valor do passo o método utilizado convergia para uma solução. Também, utilizando o menor valor de passo para o qual todos os métodos convergiam, foi possível identificar como os valores de  $m_2$ ,  $\dot{x}_d$  e  $F_1$  influenciam a solução encontrada. Enfim, o primeiro exercício programa foi um sucesso.