Escola Politécnica - Universidade de São Paulo
PMR3401 - Mecânica Computacional para Mecatrônica
2020

Exercício Programa 1

Métodos de Runge-Kutta

Diego Jun Sato Kurashima - 10274231
Felipe Gomes de Melo D'Elia - 10340624

São Paulo , 30 de Março de 2020

1. Equacionamento matemático

1.1. Modelo físico

O modelo dinâmico do modelo a ser analisado foi apresentado no enunciado conforme a Figura 1 a seguir :

$$A_0\ddot{\theta_1} = A_1\dot{\theta}_1^2 + A_2\dot{\theta}_2^2 + A_3\dot{\theta}_1 \ + A_4\dot{\theta}_2 \ + A_5$$

$$A_0 = L_1^2.L_2.R.[m_2.cos(2\theta_1 - 2\theta_2) - 2m_1 - m_2]$$
$$A_1 = L_1^2.L_2.R.m_2.sin(2\theta_1 - 2\theta_2)$$
$$A_2 = 2L_1.L_2^2.R.m_2.sin(\theta_1 - \theta_2)$$
$$A_3 = -2L_2.\mu I_z.Vel$$
$$A_4 = -2L_1.\mu I_z.Vel.cos(\theta_1 - \theta_2)$$
$$A_5 = -RL_1\left[\mathrm{L}_{2_{\mathrm{eixo}}}F_2 sin(\theta_1 - 2\theta_2) + 2sin(\theta_1)\left(F_1L_2 \ + \frac{\mathrm{L}_{2_{\mathrm{eixo}}}F_2}{2}\right)\right]$$

$$B_0\ddot{\theta_2} = B_1\ddot{\theta_1} + B_2\dot{\theta}_1^2 + B_3\dot{\theta}_2 \ + B_4$$

$$B_0 = L_2^2.R.m_2$$
$$B_1 = -L_1.L_2.R.m_2.cos(\theta_1 - \theta_2)$$
$$B_2 = L_1.L_2.R.m_2.sin(\theta_1 - \theta_2)$$
$$B_3 = -\mu I_z.Vel$$
$$B_4 = \mathrm{L}_{2_{\mathrm{eixo}}} sin(\theta_2).R.F_2$$

$$\mathrm{L}_1 = 2 \ m; \qquad \mathrm{L}_2 = 2{,}5 \ m; \qquad \mathrm{L}_{2_{\mathrm{eixo}}} = 1{,}8 \ m;$$
$$\mathrm{m}_1 = 450 \ kg; \qquad \mathrm{m}_2 = 650 \ kg;$$
$$\mathrm{F}_1 = -0{,}5 \cdot \mathrm{m}_1 \cdot \mathrm{g} \ N; \qquad \mathrm{F}_2 = -0{,}5 \cdot \mathrm{m}_2 \cdot \mathrm{g} \ N$$
$$\mu I_z = 2{,}7 \ kg.m^2; \quad R = 0{,}3 \ m; \quad \mathrm{g} = 9{,}81 \ \frac{m}{s^2}; \quad \dot{x_d} = Vel = 80\mathrm{km/h},$$

Figura 1 - Modelo dinâmico do veículo

O modelo é uma equação diferencial ordinária (EDO) de segunda ordem que representa a dinâmica do veículo.

Para resolução do problema foi usado o ambiente MATLAB. O equacionamento foi implementado por meio de uma function veículo.m ou e também por meio de uma função anônima, conforme Figuras 2 e 3:

```
% A0 = ((L1^2)*L2*R*(m2*cos(2*y(1) - 2*y(3)) - 2*m1 - m2));
% A1 = ((L1^2)*L2*R*m2*sin(2*y(1) - 2*y(3)));
% A2 = (2*L1*(L2^2)*R*m2*sin(y(1)-y(3)));
% A3 = (-2*L2*uIz*veld);
% A4 = (-2*L1*uIz*veld*cos(y(1) - y(3)));
% A5 = (-R*L1*( L2eixo*F2*sin(y(1) - 2*y(3)) + 2*sin(y(1))*( F1*L2 + (1/2)*L2eixo*F2)));

% B0 = ((L2^2)*R*m2);
% B1 = (-L1*L2*R*m2*cos(y(1) - y(3)));
% B2 = (L1*L2*R*m2*sin(y(1) - y(3)));
% B3 = (-uIz*veld);
% B4 = (L2eixo*sin(y(3))*R*F2);

% f = @(t,y) [ y(2);
%             (1/A0)*( A1*(y(2)^2) + A2*(y(4)^2) + A3*y(2) + A4*y(4) + A5);
%              y(4);
%             (1/B0)*( B1*((1/A0)*( A1*(y(2)^2) + A2*(y(4)^2) + A3*y(2) ...
%                 + A4*y(4)+ A5)) + B2*(y(2)^2) + B3*y(4) + B4 )];
```

```
f = @(t,y) [y(2,:);
            (1./((L1.^2).*L2.*R.*(m2.*cos(2.*y(1,:) - 2.*y(3,:)) - 2.*m1 - m2)))...
              .*( ((L1.^2).*L2.*R.*m2.*sin(2.*y(1,:) - 2.*y(3,:))).*(y(2,:).^2) ...
              + (2.*L1.*(L2.^2).*R.*m2.*sin(y(1,:)-y(3,:))).*(y(4,:).^2) ...
              + (-2.*L2.*uIz.*veld).*y(2,:) ...
              + (-2.*L1.*uIz.*veld.*cos(y(1) - y(3,:))).*y(4,:) ...
              + (-R.*L1.*( L2eixo.*F2.*sin(y(1,:) - 2.*y(3,:))  ...
              + 2.*sin(y(1,:)).*( F1.*L2 + (1/2).*L2eixo.*F2))));
            y(4,:);
            (1./((L2.^2).*R.*m2)).*( (-L1.*L2.*R.*m2.*cos(y(1,:) - y(3,:)) )...
              .*( (1./((L1.^2).*L2.*R.*(m2.*cos(2.*y(1,:) - 2.*y(3,:)) - 2.*m1 - m2)))...
              .*( ((L1.^2).*L2.*R.*m2.*sin(2.*y(1,:) - 2.*y(3,:))).*(y(2,:).^2) ...
              + (2.*L1.*(L2.^2).*R.*m2.*sin(y(1,:)-y(3,:))).*(y(4,:).^2) ...
              + (-2.*L2.*uIz.*veld).*y(2,:) ...
              + (-2.*L1.*uIz.*veld.*cos(y(1) - y(3,:))).*y(4,:) ...
              + (-R.*L1.*( L2eixo.*F2.*sin(y(1,:) - 2.*y(3,:)) ...
              + 2.*sin(y(1,:)).*( F1.*L2 + (1/2).*L2eixo.*F2))))) ...
              + (L1.*L2.*R.*m2.*sin(y(1,:) - y(3,:))).*(y(2,:).^2) ...
              + (-uIz.*veld).*y(4,:) + (L2eixo.*sin(y(3,:)).*R.*F2))];
```

Figura 2 : Função anônima do modelo do veículo.

```
function [f] = veiculo(L1, L2, L2eixo, m1, m2, uIz, R, g, veld, F1, F2)
% Variáveis de Estado:
% thetha1 = y(1)
% thetha1' = y(2)
% thetha2 = y(3)
% thetha2' = y(4)
f = @(t,y)[    y(2,:);
                (1./((L1.^2).*L2.*R.*(m2.*cos(2.*y(1,:)-2.*y(3,:))-2.*m1-m2)))...
              .*(((L1.^2).*L2.*R.*m2.*sin(2.*y(1,:)-2.*y(3,:))).*(y(2,:).^2) ...
              + (2.*L1.*(L2.^2).*R.*m2.*sin(y(1,:)-y(3,:))).*(y(4,:).^2) ...
              + (-2.*L2.*uIz.*veld).*y(2,:)+(-2.*L1.*uIz.*veld.*cos(y(1)-y(3,:))).*y(4,:) ...
              + (-R.*L1.*(L2eixo.*F2.*sin(y(1,:)-2.*y(3,:))+2.*sin(y(1,:)).*(F1.*L2+(1/2).*L2eixo.*F2))));
               y(4,:);
                (1./((L2.^2).*R.*m2)).*((-L1.*L2.*R.*m2.*cos(y(1,:)-y(3,:)))...
              .*((1./((L1.^2).*L2.*R.*(m2.*cos(2.*y(1,:)-2.*y(3,:))-2.*m1-m2)))...
              .*(((L1.^2).*L2.*R.*m2.*sin(2.*y(1,:)-2.*y(3,:))).*(y(2,:).^2)...
              + (2.*L1.*(L2.^2).*R.*m2.*sin(y(1,:)-y(3,:))).*(y(4,:).^2)...
              + (-2.*L2.*uIz.*veld).*y(2,:)+(-2.*L1.*uIz.*veld.*cos(y(1)-y(3,:))).*y(4,:)...
              + (-R.*L1.*( L2eixo.*F2.*sin(y(1,:)-2.*y(3,:))...
              + 2.*sin(y(1,:)).*(F1.*L2 +(1/2).*L2eixo.*F2)))))...
              + (L1.*L2.*R.*m2.*sin(y(1,:)-y(3,:))).*(y(2,:).^2)...
              + (-uIz.*veld).*y(4,:)+(L2eixo.*sin(y(3,:)).*R.*F2))   ];
end
```

Figura 3 : function veiculo.m

1.2. Método de Euler

Equacionamento

$$y_{n+1} = y_n + hf(t_n, y_n)$$

Implementação no MATLAB

```
%% Método de Euler
function [X, Y] = euler_method(f,h,yi,xi,xf)
% f : função do tipo y' = f(x, y)
% h : passo
% yi, xi : condições iniciais
% xf : condição de parada

i = 1;
y(:,i) = yi;
x(:,i) = xi;
steps = (xf-xi)/h;

% Laço Principal do Método de Euler
for i=1:steps
    y(:,i+1) = y(:,i) + h*f(x(:,i),y(:,i));
    x(:,i+1) = x(:,i) + h;
end
Y = y;
X = x;
end
```

Figura : function euler_method.m que implementa o Método de Euler

1.3. Método de Runge-Kutta 2ª ordem

Solução Analítica

$$y_{i+1} = y_i + hk_2$$

$$k_1 = f(x_i, y_i)$$

$$k_2 = f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_1\right)$$

Implementação Numérica no MATLAB

```
%% Método de Rungwe-Kutta de 2 ordem
function [X, Y] = runge_kutta_2(f,h,yi,xi,xf)
% f : função do tipo y' = f(x, y)
% h : passo
% yi, xi : condições iniciais
% xf : condição de parada
i = 1;
y(:,i) = yi;
x(:,i) = xi;
steps = (xf-xi)/h;

% Laço Principal
for i=1:steps
    k1 = f(x(:,i),y(:,i));
    k2 = f(x(:,i) + h*(1/2), y(:,i) +h*k1*(1/2) );
    y(:,i+1) = y(:,i) + h*k2;
    x(:,i+1) = x(:,i) + h;
end
 Y = y;
 X = x;
end
```

Figura : function runge_kutta_2.m que implementa o Método de RK2.

1.4. Método de Runge-Kutta 4ª ordem

Solução Analítica:

$$y_{n+1} = y_n + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$
$$t_{n+1} = t_n + h$$

$$k_1 = f\left(t_n, y_n\right)$$
$$k_2 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right)$$
$$k_3 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right)$$
$$k_4 = f\left(t_n + h, y_n + hk_3\right)$$

Implementação Numérica no MATLAB

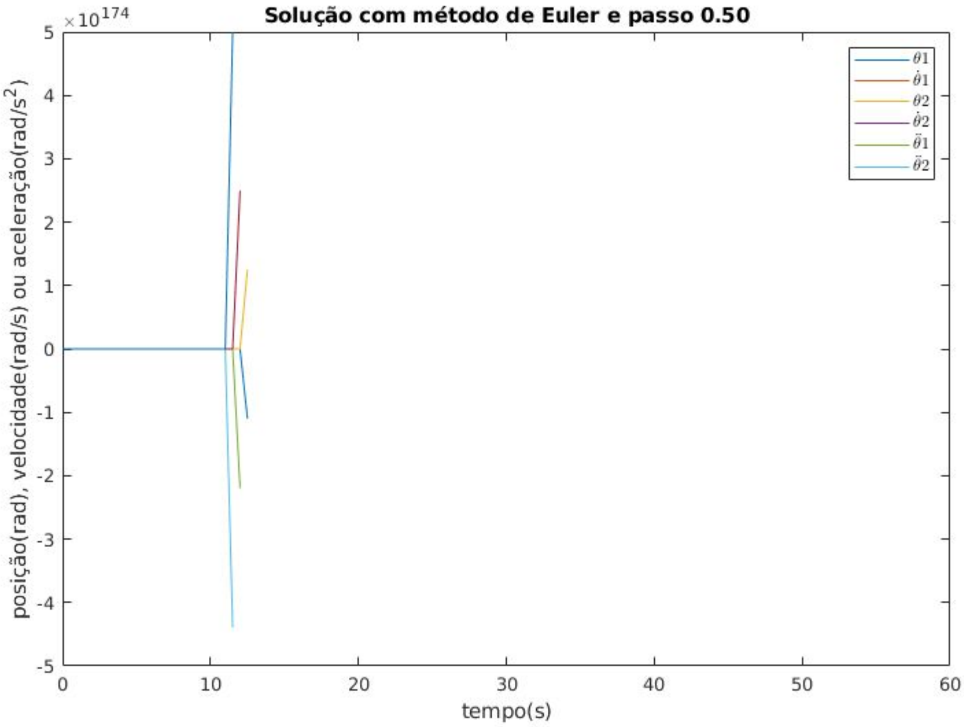
```
%% Método de Runge-Kutta de 4 ordem
function [X, Y] = runge_kutta_4(f,h,yi,xi,xf)
% f : função do tipo y' = f(x, y)
% h : passo
% yi, xi : condições iniciais
% xf : condição de parada
i = 1;
y(:,i) = yi;
x(:,i) = xi;
steps = (xf-xi)/h;
% Laço Principal
for i=1:steps
    k1 = f(x(:,i),y(:,i));
    k2 = f(x(:,i) + h*(1/2), y(:,i) +h*k1*(1/2) );
    k3 = f(x(:,i) + h*(1/2), y(:,i) +h*k2*(1/2) );
    k4 = f(x(:,i) + h, y(:,i) +h*k3 );
    y(:,i+1) = y(:,i) + (h/6)*(k1 + 2*k2 + 2*k3 + k4);
    x(:,i+1) = x(:,i) + h;
end
Y = y;
X = x;
end
```

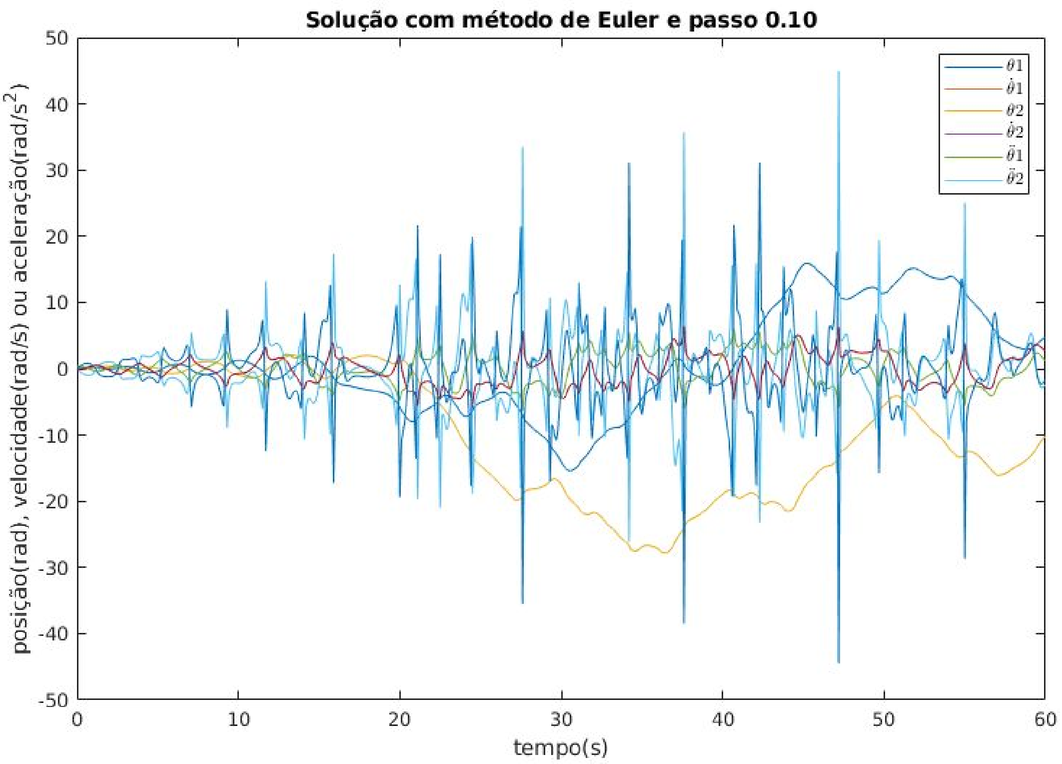
Figura : function runge_kutta_4.m que implementa o Método de RK4.

2. Questão 1

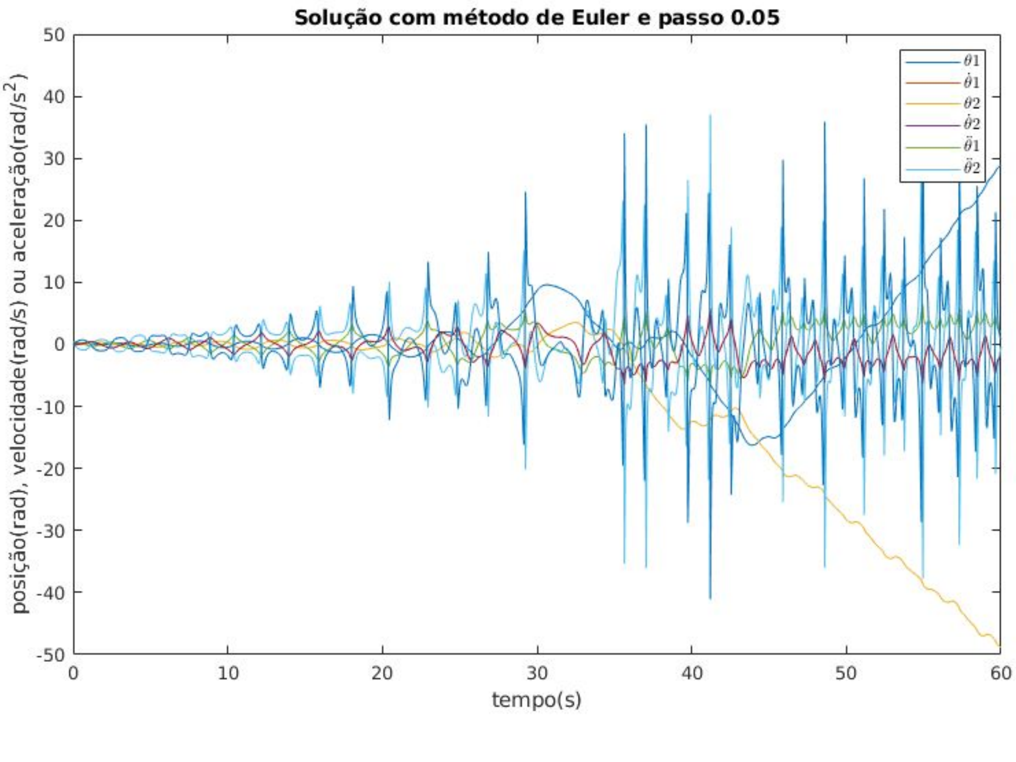
A Questão 1 trata-se da análise da solução dos Métodos Numéricos implementados para o problema do veículo proposto pelo enunciado. A análise é feita na eficiência do método e na influência do passo “h” em cada função.

2.1. Item a) - Método de Euler

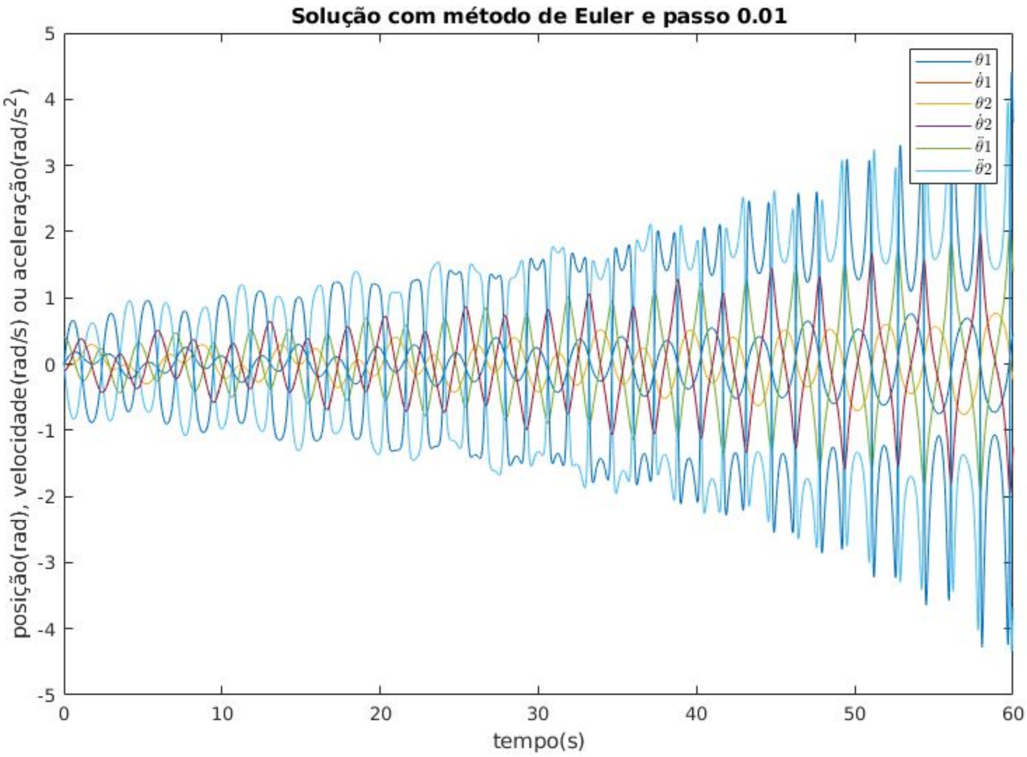




Solução com método de Euler e passo 0.10
posição(rad), velocidade(rad/s) ou aceleração(rad/s²)
tempo(s)
50
40
30
20
10
0
-10
-20
-30
-40
-50
0
10
20
30
40
50
60
θ1
θ̇1
θ2
θ̇2
θ̈1
θ̈2



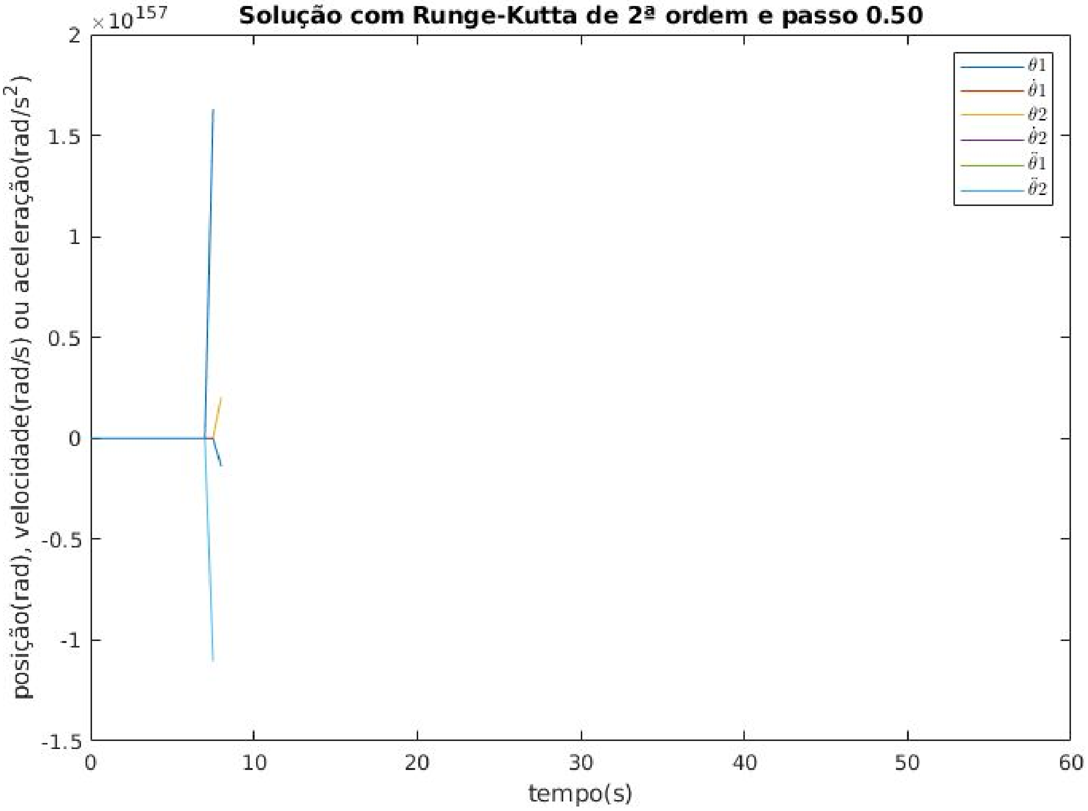
Solução com método de Euler e passo 0.05
posição(rad), velocidade(rad/s) ou aceleração(rad/s²)
tempo(s)
50
40
30
20
10
0
-10
-20
-30
-40
-50
0
10
20
30
40
50
60
θ1
θ̇1
θ2
θ̇2
θ̈1
θ̈2

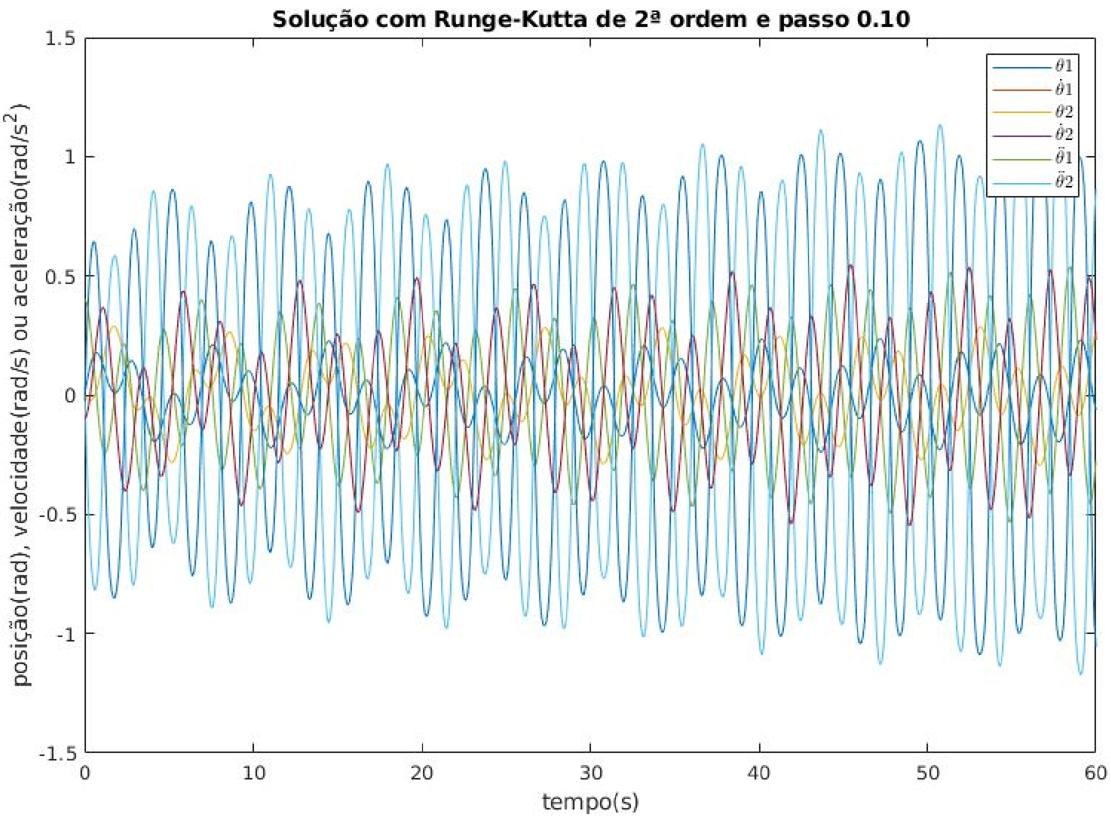


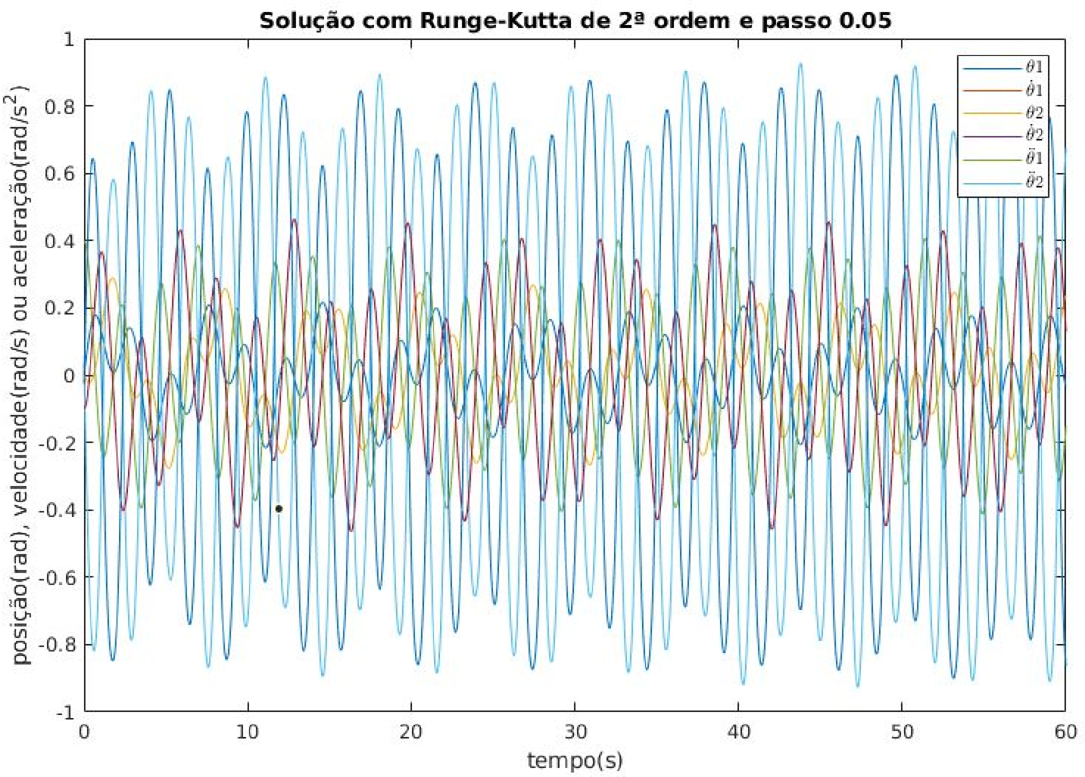
Discussão

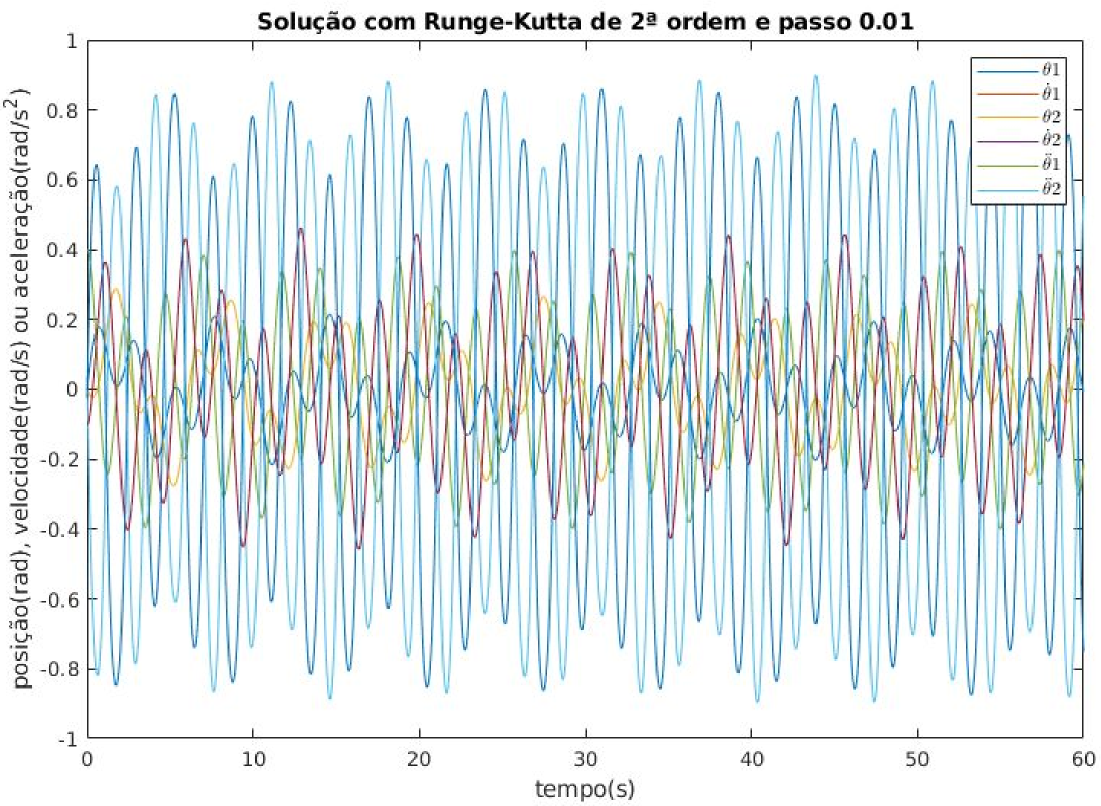
Podemos observar que o método de Euler não converge para nenhum dos passos escolhidos, o que nos mostra que a simplicidade do método se reflete em baixa robustez

2.2. Item b) - Método de Runge-Kutta de 2ª ordem





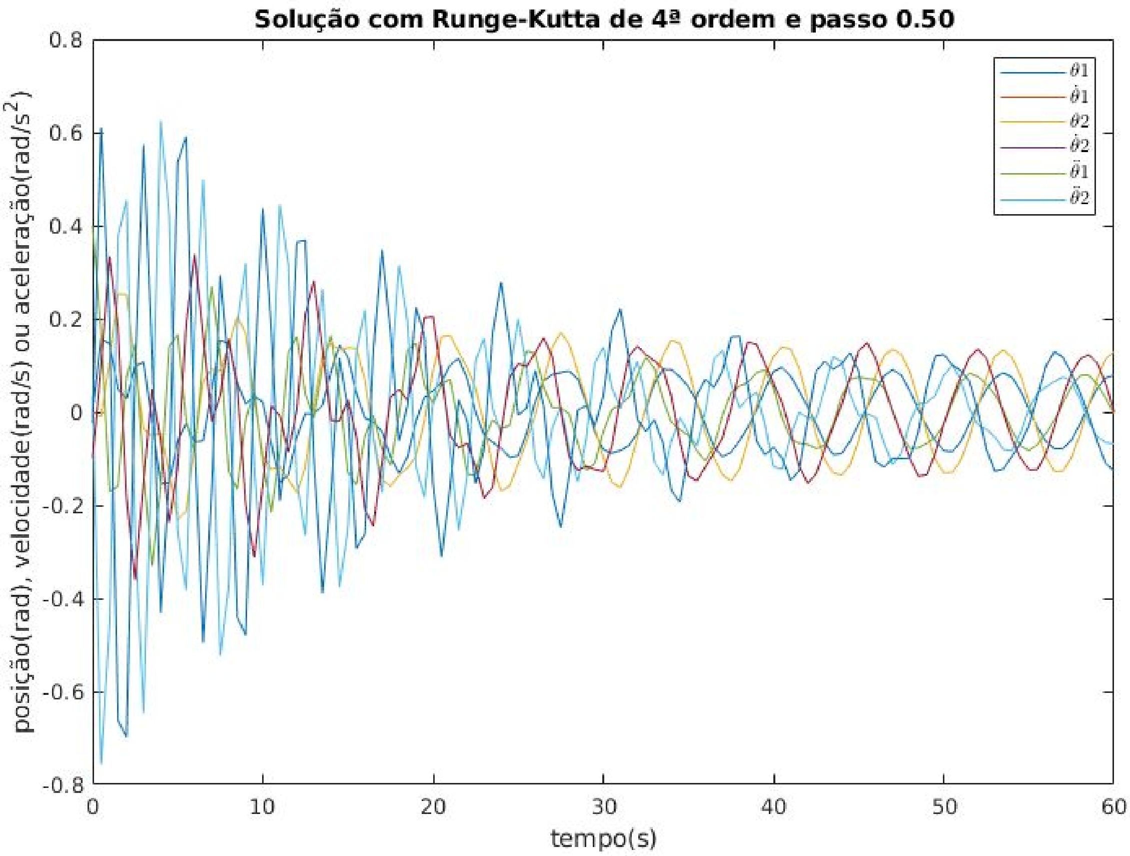

Solução com Runge-Kutta de 2ª ordem e passo 0.05
posição(rad), velocidade(rad/s) ou aceleração(rad/s^2)
tempo(s)
$\theta 1$
$\dot{\theta} 1$
$\theta 2$
$\dot{\theta} 2$
$\ddot{\theta} 1$
$\ddot{\theta} 2$

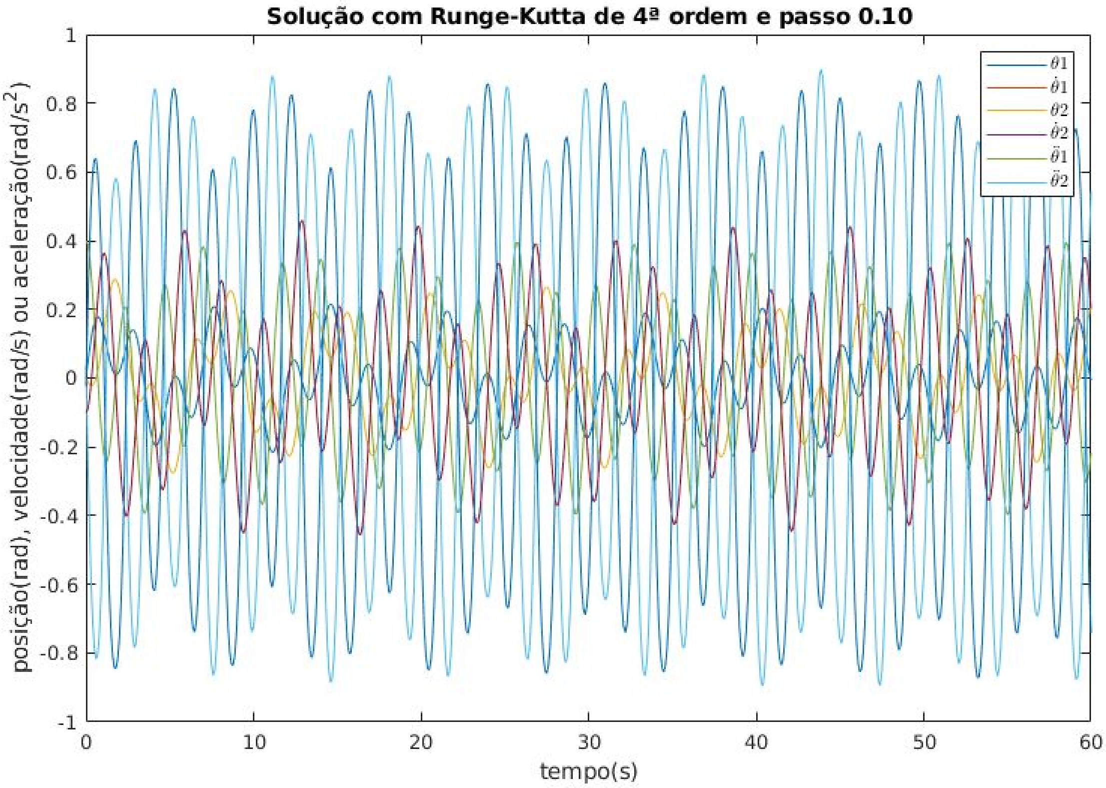

Solução com Runge-Kutta de 2ª ordem e passo 0.01
posição(rad), velocidade(rad/s) ou aceleração(rad/s^2)
tempo(s)
$\theta 1$
$\dot{\theta} 1$
$\theta 2$
$\dot{\theta} 2$
$\ddot{\theta} 1$
$\ddot{\theta} 2$

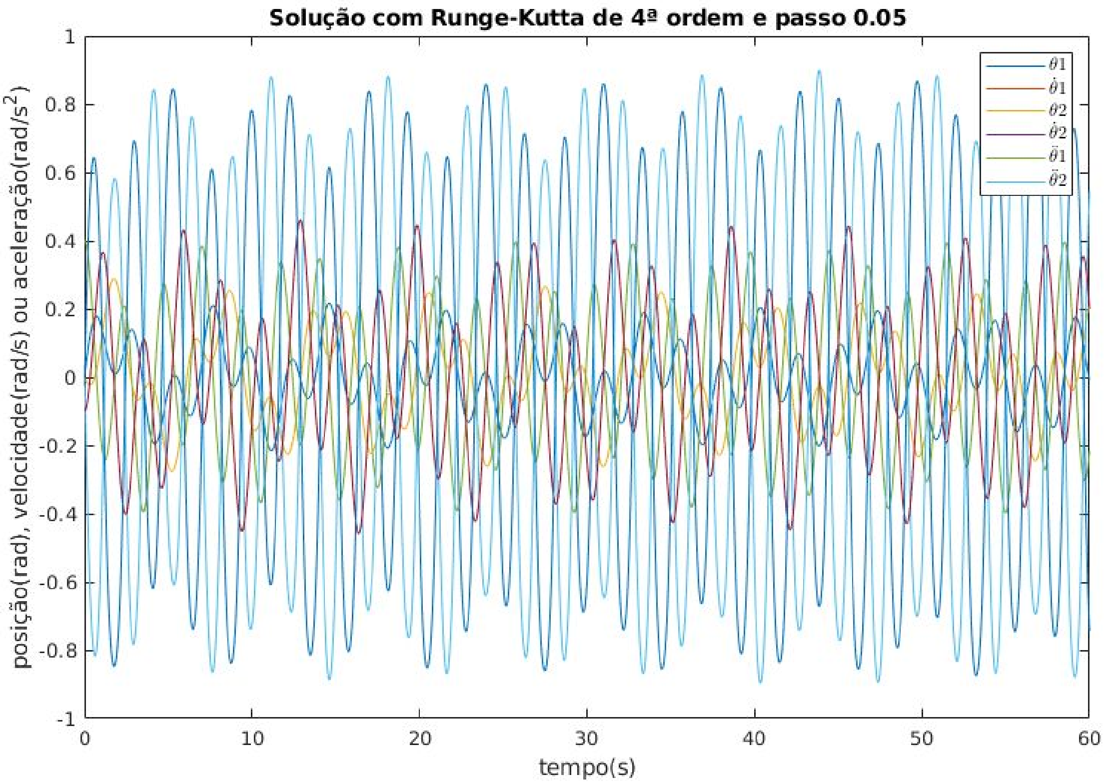
Discussão

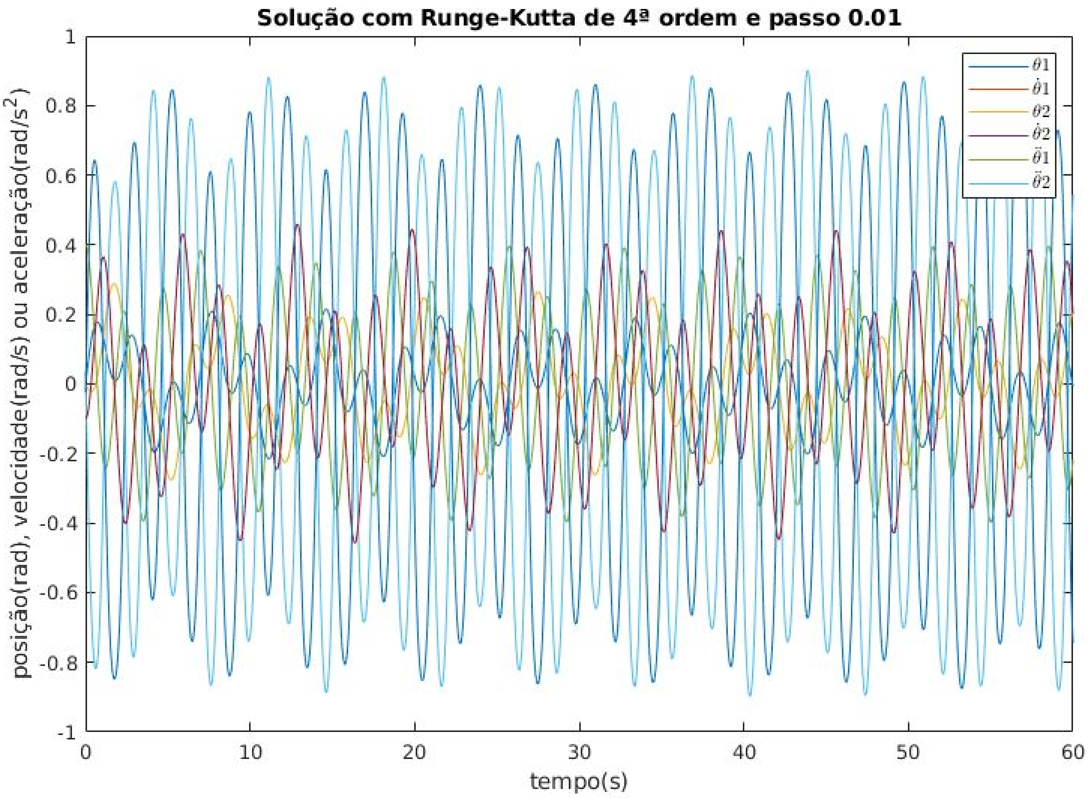
Utilizando passo 0.50, observamos que o método diverge. Com passo 0.10, conseguimos observar a forma da solução, mas também um aumento sensível do erro numérico em instantes futuros da solução. Com passos menores do que 0.05 podemos observar a convergência da solução sem problemas perceptíveis.

2.3. Item c) - Método de Runge-Kutta de 4ª ordem




Solução com Runge-Kutta de 4ª ordem e passo 0.10
posição(rad), velocidade(rad/s) ou aceleração(rad/s^2)
tempo(s)
$\theta 1$
$\dot{\theta} 1$
$\theta 2$
$\dot{\theta} 2$
$\ddot{\theta} 1$
$\ddot{\theta} 2$
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
10
20
30
40
50
60


Solução com Runge-Kutta de 4ª ordem e passo 0.05
posição(rad), velocidade(rad/s) ou aceleração(rad/s^2)
tempo(s)
$\theta 1$
$\dot{\theta} 1$
$\theta 2$
$\dot{\theta} 2$
$\ddot{\theta} 1$
$\ddot{\theta} 2$
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
10
20
30
40
50
60



Discussão

Com passou 0.50, nenhum dos métodos converge para a solução esperada. O método de Runge-Kutta de 4ª ordem é capaz de convergir com passo 0.10, o que não foi observado nos dois métodos anteriores, o que nos mostra como esse método é robusto e eficiente na resolução de equações diferenciais.

2.4. Conclusões

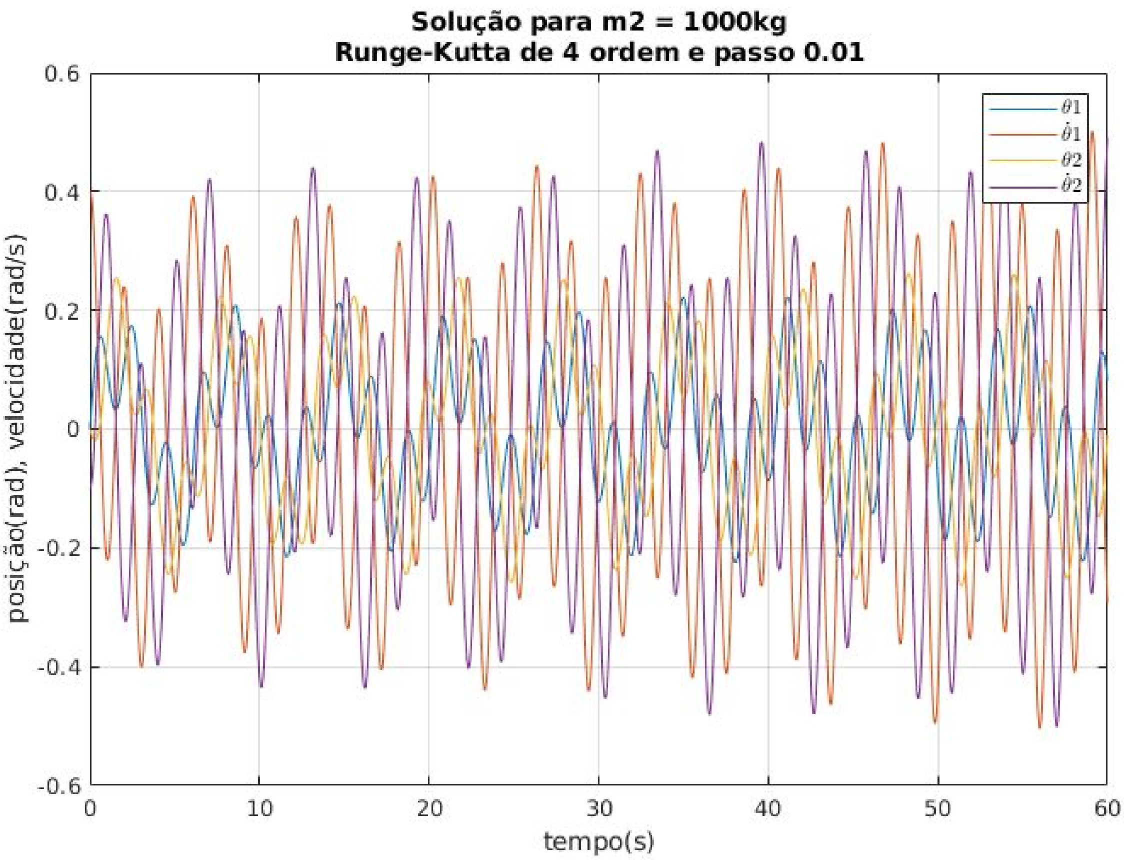
A comparação pode ser feita em duas frentes, o método utilizado e o passo escolhido. Em geral, para o problema em específico, o Método de Euler foi menos eficiente pois sua solução não convergiu para um grande número de passos em comparação aos métodos RK2 e RK4 de Runge-Kutta, que se mostraram métodos mais robustos. O RK4 em especial, por ser mais preciso que o RK2 apresentou melhores resultados.

Analisando o passo, foi possível perceber que quanto mais fino, a precisão do cálculo foi melhor quanto menor o ajuste, entretanto, um menor passo leva ao ao aumento da carga do cálculo computacional, o que leva a um processamento muito mais lento dependendo do quanto fino fô-lo.

3. Questão 2

Para a Questão 2 foi utilizado o método de **Runge-Kutta de 4ª ordem** com passo **h = 0.01**, pois, conforme concluído na Questão 1, apresentou melhor precisão computacional.

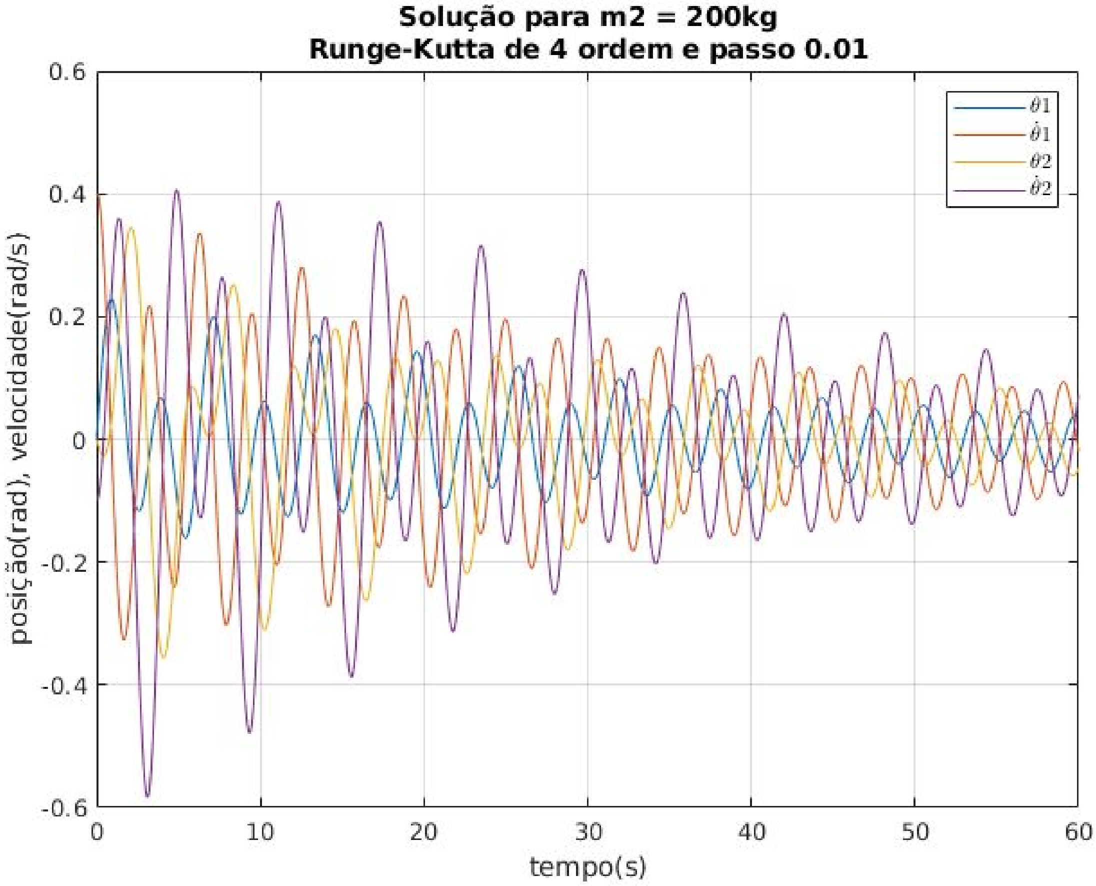
3.1. Item a) - $m_2 = 1000\ kg$



Discussão

Pelo gráfico, é possível perceber que o movimento do veículo para aumento de massa **m_2 = 1000 kg** mantém-se oscilatório, o que indica que o movimento não se estabiliza. A amplitude do movimento também se alterou pouco em relação às condições da Questão 1.

3.2. Item b) - $m_2 = 200\ kg$

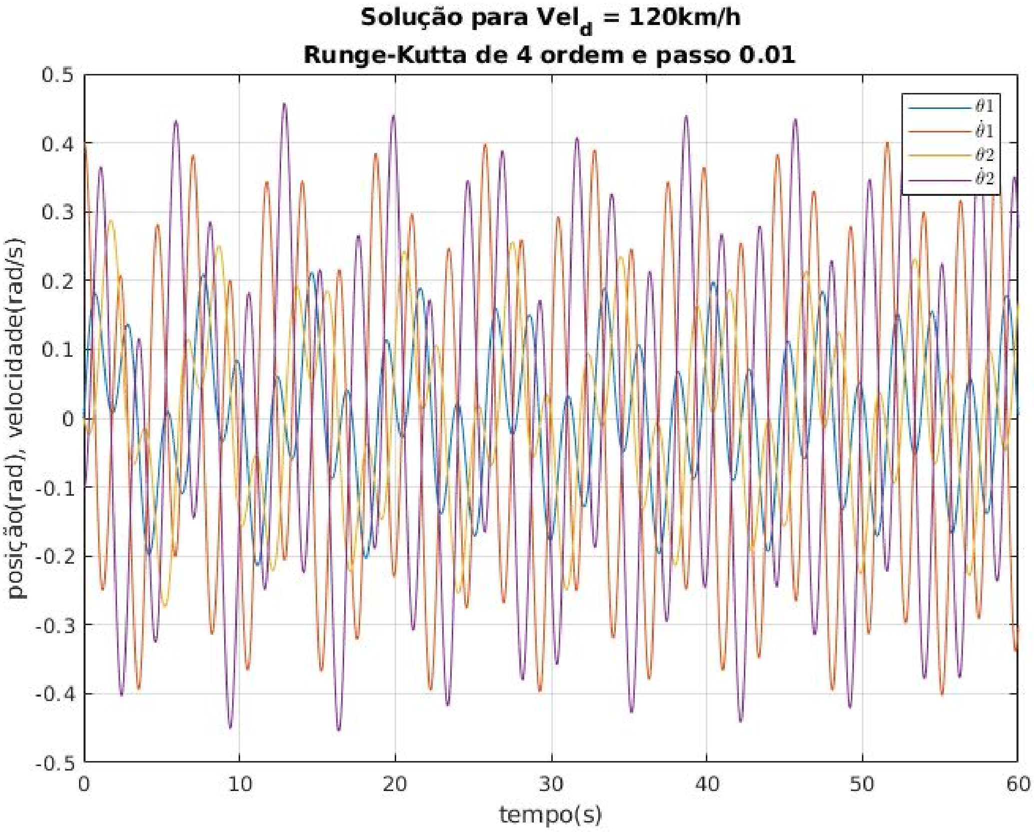


Discussão

O carro estabiliza com a diminuição de massa para $\mathbf{m_2 = 200\ kg}$. A amplitude do movimento das articulações diminui com o tempo. O que torna o movimento do veículo mais seguro.

Comparando os itens a) e b) podemos perceber a importância em respeitar os limites de segurança de carga no reboque, bem como o impacto na distribuição das massas.

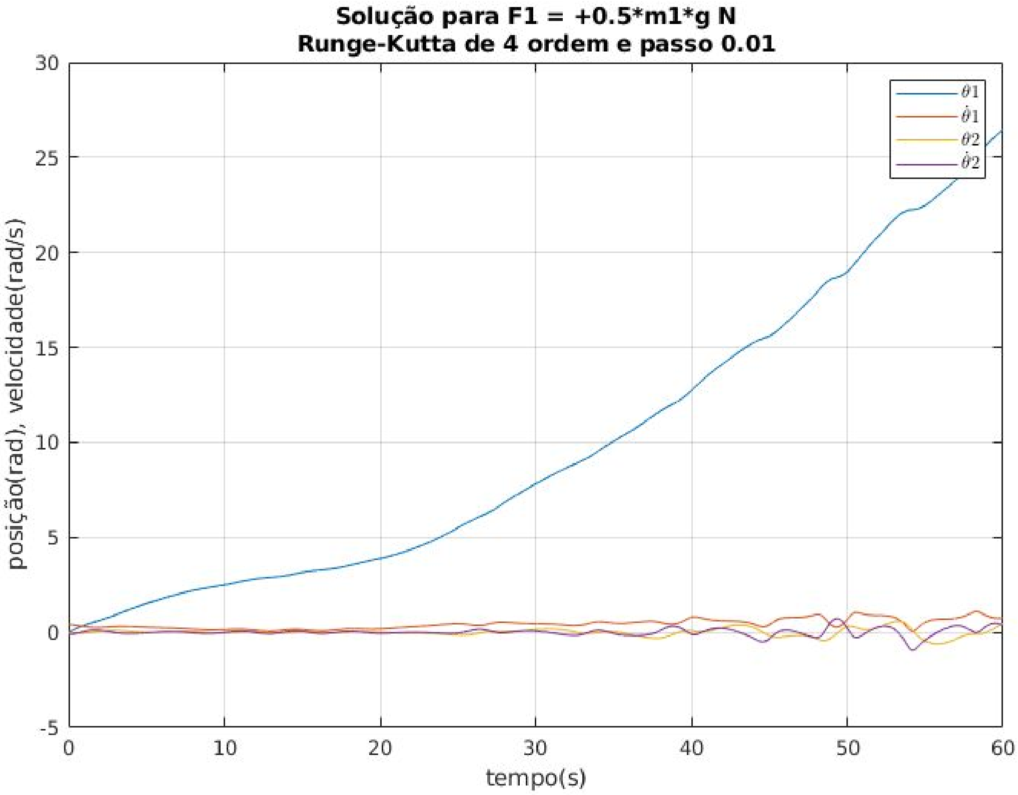
3.3. Item c) - $vel = 120\ km/h$



Discussão

Não ocorreu mudança significativa com a mudança de velocidade do veículo para **vel$_d$ = 120 km/h**. É possível que a velocidade de 120 km/h apresente apenas mudanças muito pequenas.

3.4. Item d) - Tração dianteira



Discussão

Considerando tração positiva na roda $\mathbf{F_1 = 0.5^*m_1^*g}$, o movimento do reboque veículo desestabiliza totalmente. Nesse caso, a posição angular crescente do primeiro reboque indica que ele inicia uma rotação de infinitamente. O que é coerente, pois a tração positiva é favorável a desestabilização do veículo e à oscilação deste.

É necessário frisar que nesse caso específico, o modelo não impõe restrição geométrica física para a articulação do reboque, que explica o fato da solução numérica ter fornecido um movimento circular infinito para essa articulação em específico.

3.5. Conclusões

Primeiramente, comparando os Métodos de Integração Numérica observou-se a importância de analisar o erro atrelado à cada método pois a solução poderá convergir (ou não convergir) à valores com erros muito elevados. O passo h de cada laço de iteração do método também influencia na resposta, estando muito atrelado à precisão instantânea em cada ponto da solução. No geral, o Método RK4 por apresentar maior precisão, apresentou resultados mais próximos e foi usado para solução do problema do veículo.

Analisando o veículo, com base nas respostas anteriores é possível perceber que a distribuição de massa do veículo afeta significativamente o movimento das articulações dos reboques, sendo que concentração de massa no extremo traseiro do veículo torna o veículo instável conforme comparação entre os itens a) e b) da Questão2).

No caso específico, a velocidade do veículo não afetou significamente a oscilação do reboque. É possível que para esse sistema em específico, essa variação ainda não seja o suficiente para fornecer mudanças drásticas, mas o mesmo não pode ser dito em relação à outros sistemas. Parâmetros, principalmente os inerciais como massa, irão definir o quanto a velocidade do veículo afeta o sistema.