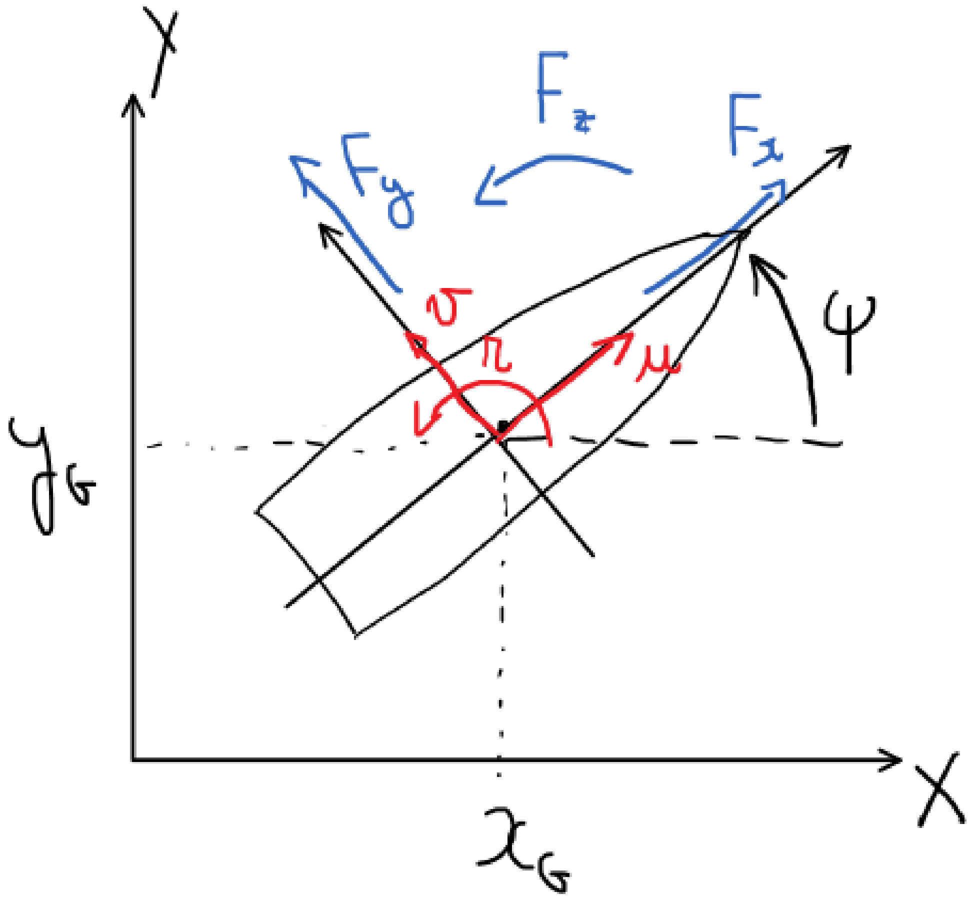
PMR3523 - Controle Moderno

Entrega 2 - Posicionamento Dinâmico de Embarcações

Ângelo Bianco Yanagita

Diôgo Lopes Cavalcante



Variáveis

Estados:	x_G, y_G: posição do navio (m)
	ψ: aproamento do navio (rad)
	u,v: velocidades nas direções longitudinal e tranversal (m/s)
	r: velocidade de guinada (rad/s)
Entradas:	Fx, Fy: Forças resultantes dos propulsores nas direções longitudinal e transversal (N)
	Fz: Momento de guinada resultante dos propulsores (N.m)
Saídas:	x_G, y_G,ψ

Equações do sistema:

$$
\begin{aligned}
&\dot{x}_G = u\cos\psi - v\sin\psi \\
&\dot{y}_G = u\sin\psi + v\cos\psi \\
&\dot{\psi} = r \\
&(M + M_{11})\dot{u} - (M + M_{22})vr - (Mx_G + M_{26})r^2 = F_x \\
&(M + M_{22})\dot{v} + (Mx_G + M_{26})\dot{r} + (M + M_{11})ur = F_y \\
&(I_z + M_{66})\dot{r} + (Mx_G + M_{26})(\dot{v} + ur) + (M_{22} - M_{11})uv = F_z
\end{aligned}
$$

As variáveis de estado, entrada e saída são as seguintes:

$$
\begin{aligned}
x &= \begin{bmatrix} x_g & y_g & \psi & u & v & r \end{bmatrix}^T \\
u &= \begin{bmatrix} F_x & F_y & F_z \end{bmatrix}^T \\
y &= \begin{bmatrix} x_g & y_g & \psi \end{bmatrix}^T
\end{aligned}
$$

Declaração das simbólicas a serem usadas

```
syms xg yg psi u v r Fx Fy Fz m m11 m22 m26 m66 izz
syms xg_dot yg_dot psi_dot u_dot v_dot r_dot
```

Definição dos membros direitos da equação de cada estado

```
eq1 = u*cos(psi) - v*sin(psi);
eq2 = u*sin(psi) + v*cos(psi);
eq3 = r;
eq4 = (Fx + (m*xg + m26)*r^2 + (m + m22)*v*r)/(m+m11);
eq5 = (Fy -(m*xg + m26)*r_dot - (m + m11)*u*r)/(m+m22);
eq6 = (Fz - (m*xg + m26)*(v_dot + u*r) - (m22 - m11)*u*v)/(izz + m66);
```

Definição das equações de saída:

```
eq7 = xg;
eq8 = yg;
eq9 = psi;
```

Linearização de acordo com as seguintes condições:

$$\dot{x}_G^* = \dot{y}_G^* = x_G^* = y_G^* = \psi^* = 0$$

$$u^* = v^* = r^* = \dot{u}^* = \dot{v}^* = \dot{r}^* = 0$$

e entradas nos esforços propulsores todas nulas.

```
taylor_xg_dot = taylor(eq1, [u, v, psi], 0 , 'Order', 2)
```

taylor_xg_dot = u

```
taylor_yg_dot = taylor(eq2, [u, v, psi], 0 , 'Order', 2)
```

taylor_yg_dot = v

```
taylor_psi_dot = taylor(eq3, r , 0 , 'Order', 2)
```

taylor_psi_dot = r

```
taylor_u_dot = taylor(eq4, [xg, r, v, Fx], 0 , 'Order', 2)
```

taylor_u_dot =

$$\frac{\mathrm{Fx}}{m+m_{11}}$$

```
taylor_v_dot = taylor(eq5, [xg, r_dot, u, r, Fy], 0 , 'Order', 2)
```

taylor_v_dot =

$$\frac{\mathrm{Fy}}{m+m_{22}} - \frac{m_{26}\, r_{\mathrm{dot}}}{m+m_{22}}$$

```
taylor_r_dot = taylor(eq6, [xg, v_dot, u, r, v, Fz], 0 , 'Order', 2)
```

taylor_r_dot =

$$\frac{\mathrm{Fz}}{\mathrm{izz}+m_{66}} - \frac{m_{26}\, v_{\mathrm{dot}}}{\mathrm{izz}+m_{66}}$$

feitas as expansões de taylor de cada membro direito, têm-se as equações linearizadas de estado

```
lin_xg_dot = taylor_xg_dot
```

lin_xg_dot = u

```
lin_yg_dot = taylor_yg_dot
```

lin_yg_dot = v

```
lin_psi_dot = taylor_psi_dot
```

lin_psi_dot = r

```
lin_u_dot = taylor_u_dot
```

lin_u_dot =

$$\frac{\mathrm{Fx}}{m+m_{11}}$$

```
lin_v_dot = v_dot == taylor_v_dot;

lin_r_dot = r_dot == taylor_r_dot;
```

Isolando as derivadas de v e r:

```
[lin_v_dot,lin_r_dot]=solve([lin_v_dot,lin_r_dot],[v_dot,r_dot])
```

lin_v_dot =

$$\frac{\mathrm{Fy}\,\mathrm{izz}+\mathrm{Fy}\,m_{66}-\mathrm{Fz}\,m_{26}}{-{m_{26}}^2+\mathrm{izz}\,m+\mathrm{izz}\,m_{22}+m\,m_{66}+m_{22}\,m_{66}}$$

lin_r_dot =

$$\frac{\mathrm{Fz}\,m-\mathrm{Fy}\,m_{26}+\mathrm{Fz}\,m_{22}}{-{m_{26}}^2+\mathrm{izz}\,m+\mathrm{izz}\,m_{22}+m\,m_{66}+m_{22}\,m_{66}}$$

Criação das Matrizes:

```
%{
syms nop

A=[nop nop nop nop nop nop;
   nop nop nop nop nop nop;
   nop nop nop nop nop nop;
   nop nop nop nop nop nop;
   nop nop nop nop nop nop;
   nop nop nop nop nop nop];

B=[nop nop nop;
   nop nop nop;
   nop nop nop;
   nop nop nop;
   nop nop nop;
   nop nop nop];

C=[nop nop nop nop nop nop;
   nop nop nop nop nop nop;
   nop nop nop nop nop nop];

vect_estados=[xg yg psi u v r];
vect_entradas=[Fx Fy Fz];
vect_saidas=[xg yg psi];



for i=1:6

    A(1,i)=collect(lin_xg_dot,vect_estados(i))
    A(2,i)=collect(lin_yg_dot,vect_estados(i));
```

```
    A(3,i)=collect(lin_psi_dot,vect_estados(i));
    A(4,i)=collect(lin_u_dot,vect_estados(i));
    A(5,i)=collect(lin_v_dot,vect_estados(i));
    A(6,i)=collect(lin_r_dot,vect_estados(i));

end

for i=1:3

    B(1,i)=collect(lin_xg_dot,vect_entradas(i));
    B(2,i)=collect(lin_yg_dot,vect_entradas(i));
    B(3,i)=collect(lin_psi_dot,vect_entradas(i));
    B(4,i)=collect(lin_u_dot,vect_entradas(i));
    B(5,i)=collect(lin_v_dot,vect_entradas(i));
    B(6,i)=collect(lin_r_dot,vect_entradas(i));

end

nop = 0;

display(A)
display(B)
display(C)
%}
```

Criação das Matrizes:

```
syms d

A = [0 0 0 1 0 0;
     0 0 0 0 1 0;
     0 0 0 0 0 1;
     0 0 0 0 0 0;
     0 0 0 0 0 0;
     0 0 0 0 0 0];

B = [0             0                 0;
     0             0                 0;
     0             0                 0;
     1/(m+m11)     0                 0;
     0             (izz+m66)/d       -m26/d;
     0             -m26/d            (m+m22)/d];

C = [1 0 0 0 0 0;
     0 1 0 0 0 0;
     0 0 1 0 0 0;
     0 0 0 0 0 0;
     0 0 0 0 0 0;
     0 0 0 0 0 0];

D = 0;
```

Usando os valores numéricos do enunciado:

```
m = 7240e3;
izz = 2750e6;
m11 = 640e3;
m22 = 6400e3;
```

```
m66 = 1560e6;
m26 = 7900e3;

display(A)
```

A = *6x6 double*

0	0	0	1	0	0
0	0	0	0	1	0
0	0	0	0	0	1
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

```
display(B)
```

B =

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{m+m_{11}} & 0 & 0 \\ 0 & \frac{\mathrm{izz}+m_{66}}{d} & -\frac{m_{26}}{d} \\ 0 & -\frac{m_{26}}{d} & \frac{m+m_{22}}{d} \end{pmatrix}$$

```
display(C)
```

C = *6x6 double*

1	0	0	0	0	0
0	1	0	0	0	0
0	0	1	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

```
display(D)
```

D = 0

Definindo tempo de simulação:

```
t = 0:0.001:10;
i = 1 % contador para os gráficos
```

i = 1

Definindo matriz de vasculhamento:

```
single_G=[0.05 0.1 0.25];
double_G=[0 0.05;
          0 0.1;
```

```
             0 0.25;
             0.05 0;
             0.05 0.05;
             0.05 0.1;
             0.05 0.25;
             0.1 0;
             0.1 0.05;
             0.1 0.1;
             0.1 0.25;
             0.25 0;
             0.25 0.05;
             0.25 0.1;
             0.25 0.25];
```

Gerando gráficos comparativos para cada entrada atuando sozinha com 100% do esforço de controle:

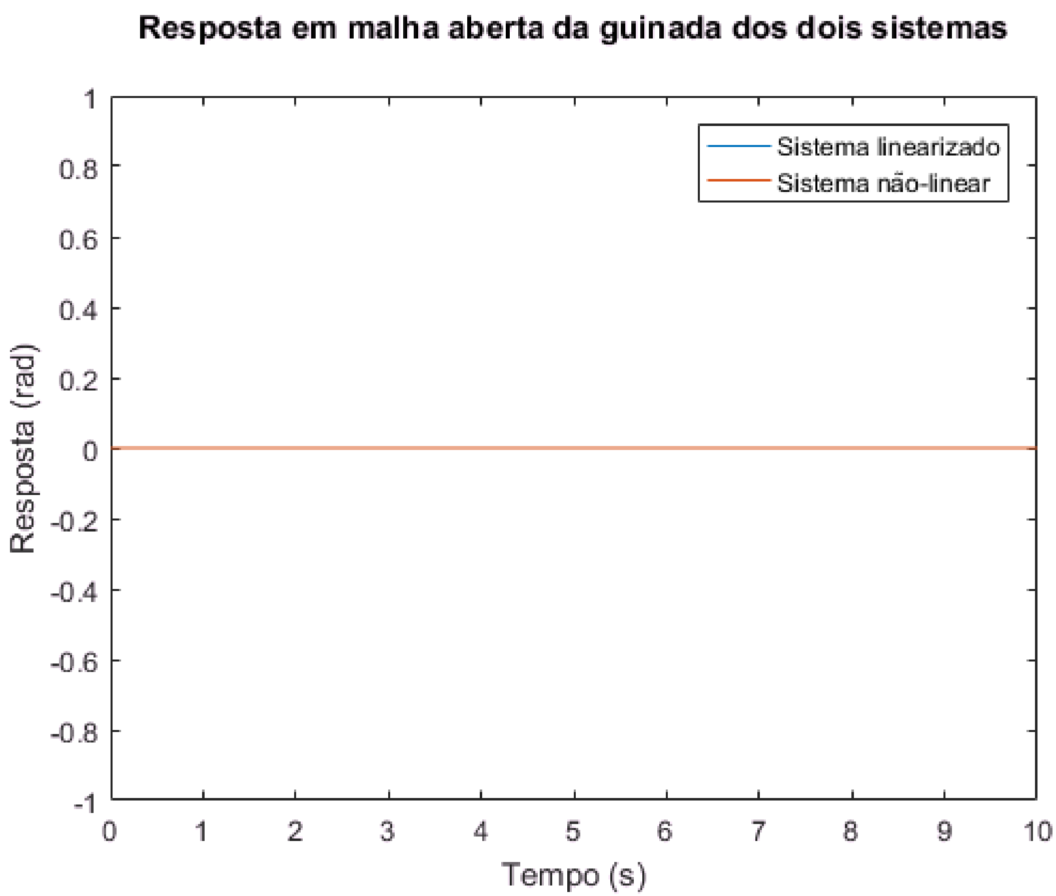
Fx:

(1) $F_x = 2.5 * 10^5 \ \ N; F_y = 0 ; M_z = 0$

```
fx = 2.5e5;
fy = 0;
mz = 0;

[xg_linear, yg_linear, psi_linear] = sim_linear(t);
[xg_non_linear, yg_non_linear, psi_non_linear] = sim_non_linear(t);

i=plota_graficos(psi_linear, yg_non_linear, xg_non_linear, xg_linear, yg_linear, psi_non_linea
```



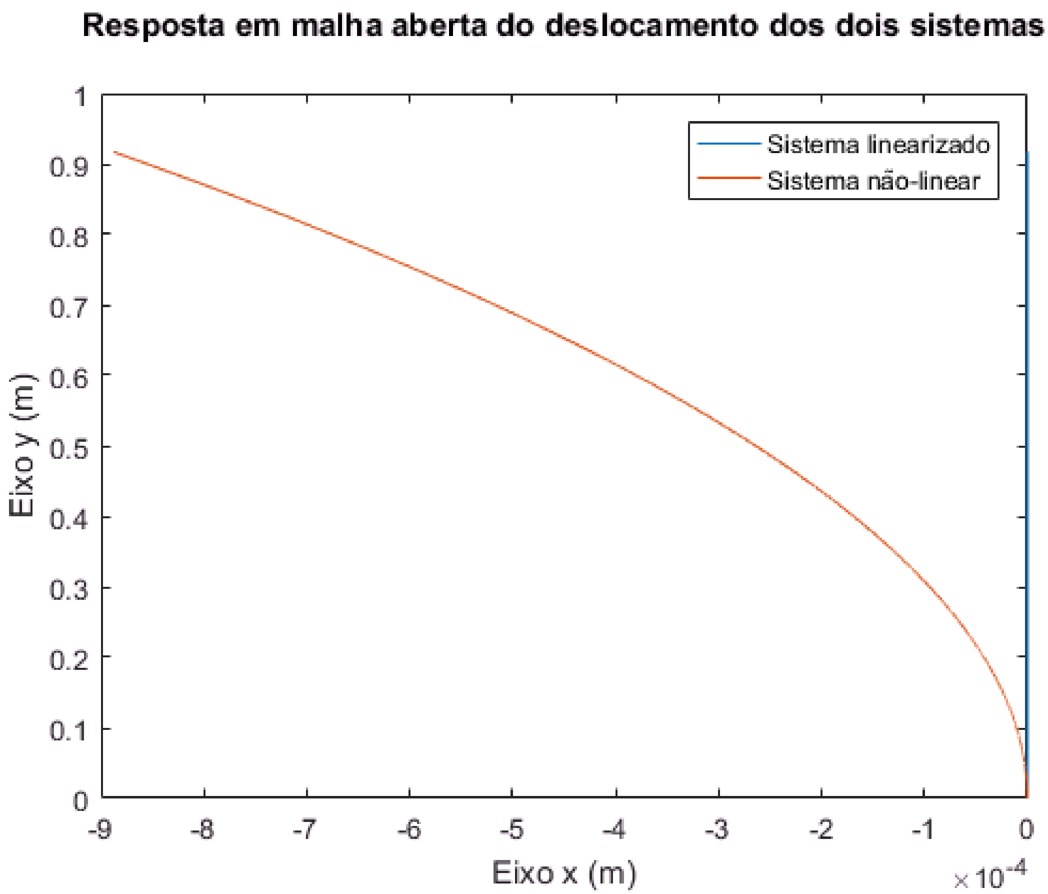
Fy:

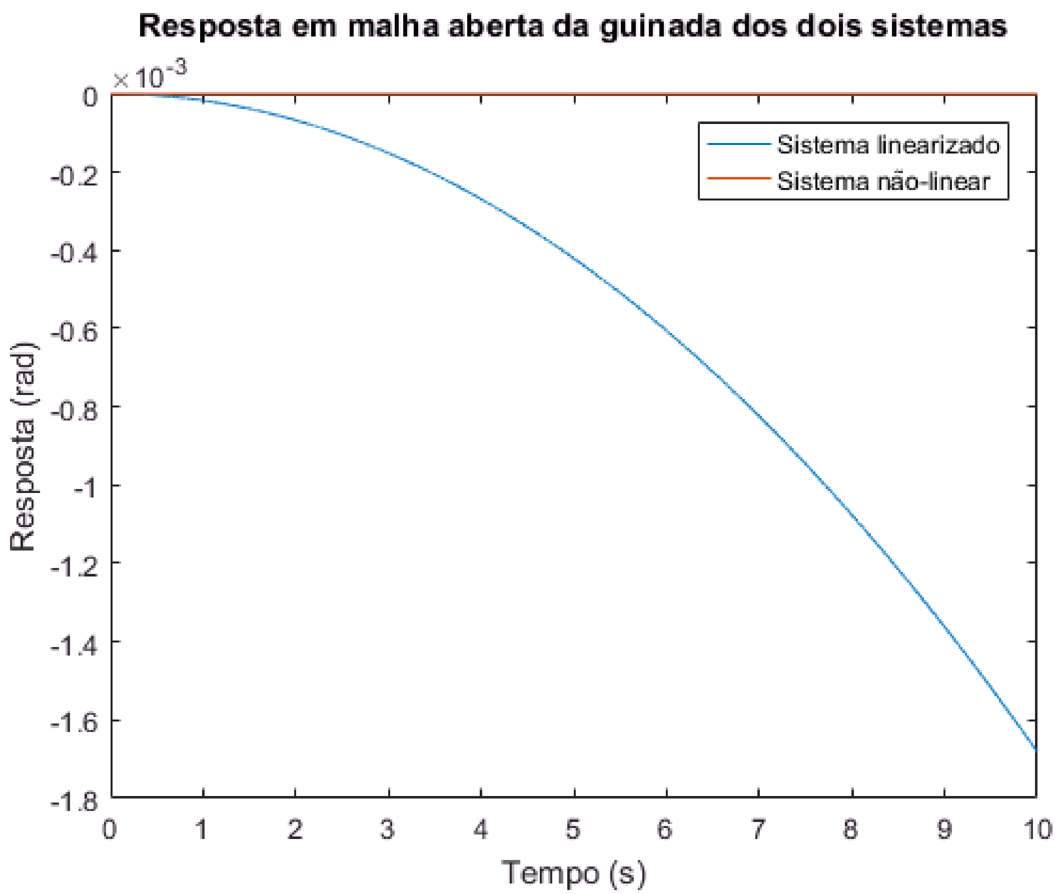
(2) $F_x = 0\ ; F_y = 2.5 * 10^5\ \ N; M_z = 0$

```
fx = 0;
fy = 2.5e5;
mz = 0;

[xg_linear, yg_linear, psi_linear] = sim_linear(t);
[xg_non_linear, yg_non_linear, psi_non_linear] = sim_non_linear(t);

i=plota_graficos(psi_linear, yg_non_linear, xg_non_linear, xg_linear, yg_linear, psi_non_linea
```





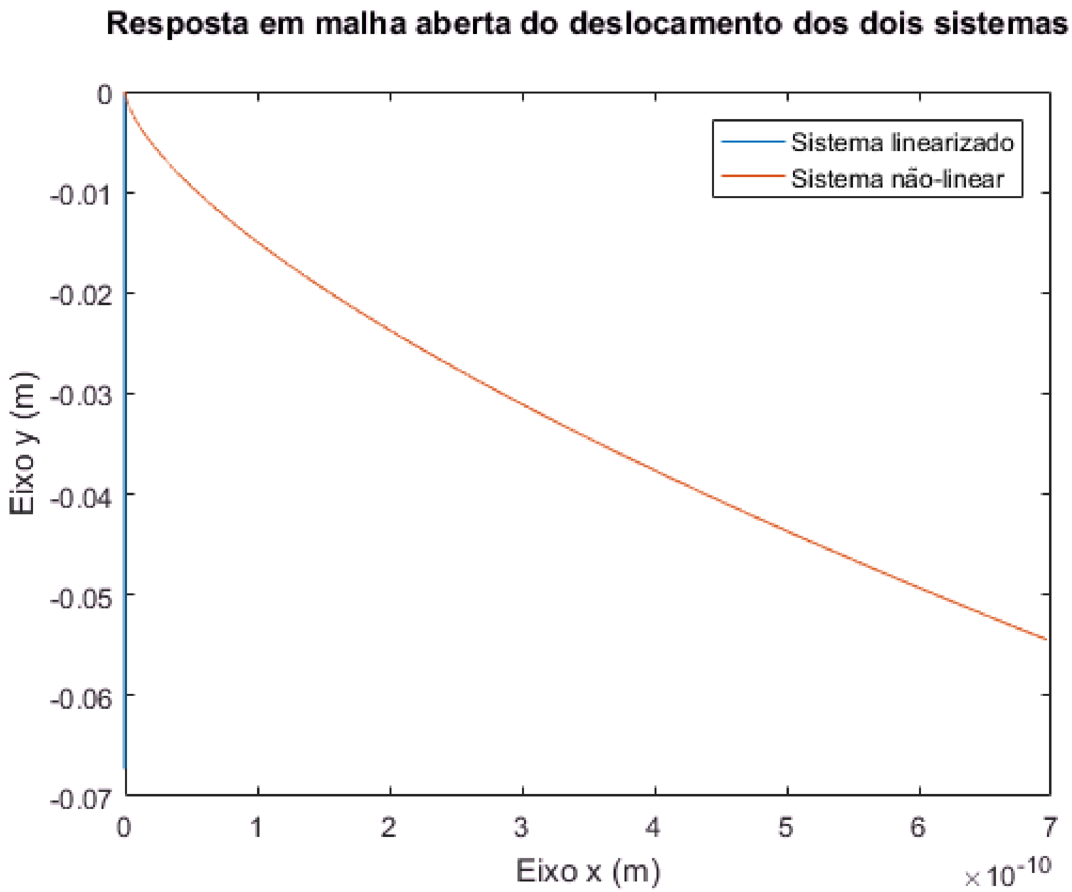
Fz:

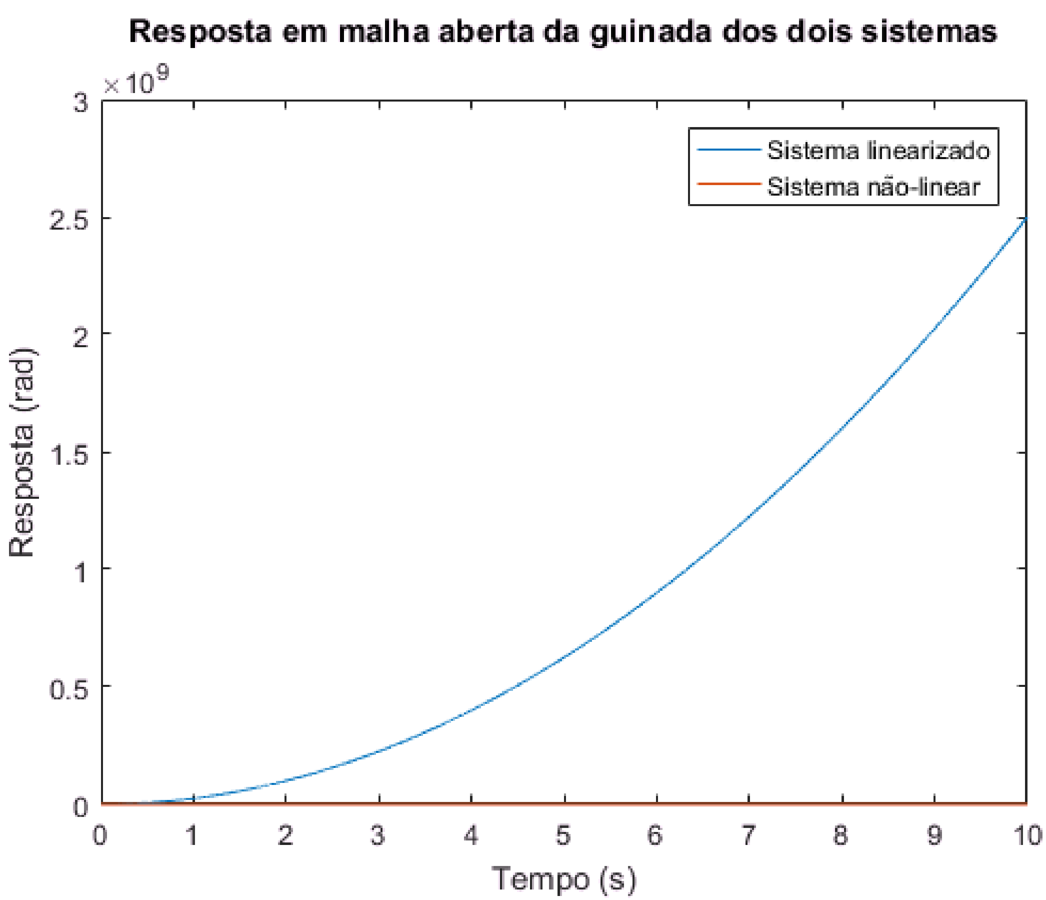
(3) $F_x = 0$; $F_y = 0$; $M_z = 1 * 10^7\ N.m$

```
fx = 0;
fy = 0;
mz = 1e7;

[xg_linear, yg_linear, psi_linear] = sim_linear(t);
[xg_non_linear, yg_non_linear, psi_non_linear] = sim_non_linear(t);

i=plota_graficos(psi_linear, yg_non_linear, xg_non_linear, xg_linear, yg_linear, psi_non_linea
```


Resposta em malha aberta do deslocamento dos dois sistemas
Sistema linearizado
Sistema não-linear
Eixo y (m)
Eixo x (m)
×10^-10
0
-0.01
-0.02
-0.03
-0.04
-0.05
-0.06
-0.07
0 1 2 3 4 5 6 7


Resposta em malha aberta da guinada dos dois sistemas
×10^9
Sistema linearizado
Sistema não-linear
Resposta (rad)
Tempo (s)
0
0.5
1
1.5
2
2.5
3
0 1 2 3 4 5 6 7 8 9 10

Podemos ver que para 100%, a aproximação linear não tem bom desempenho. Tentaremos reduzir o esforço

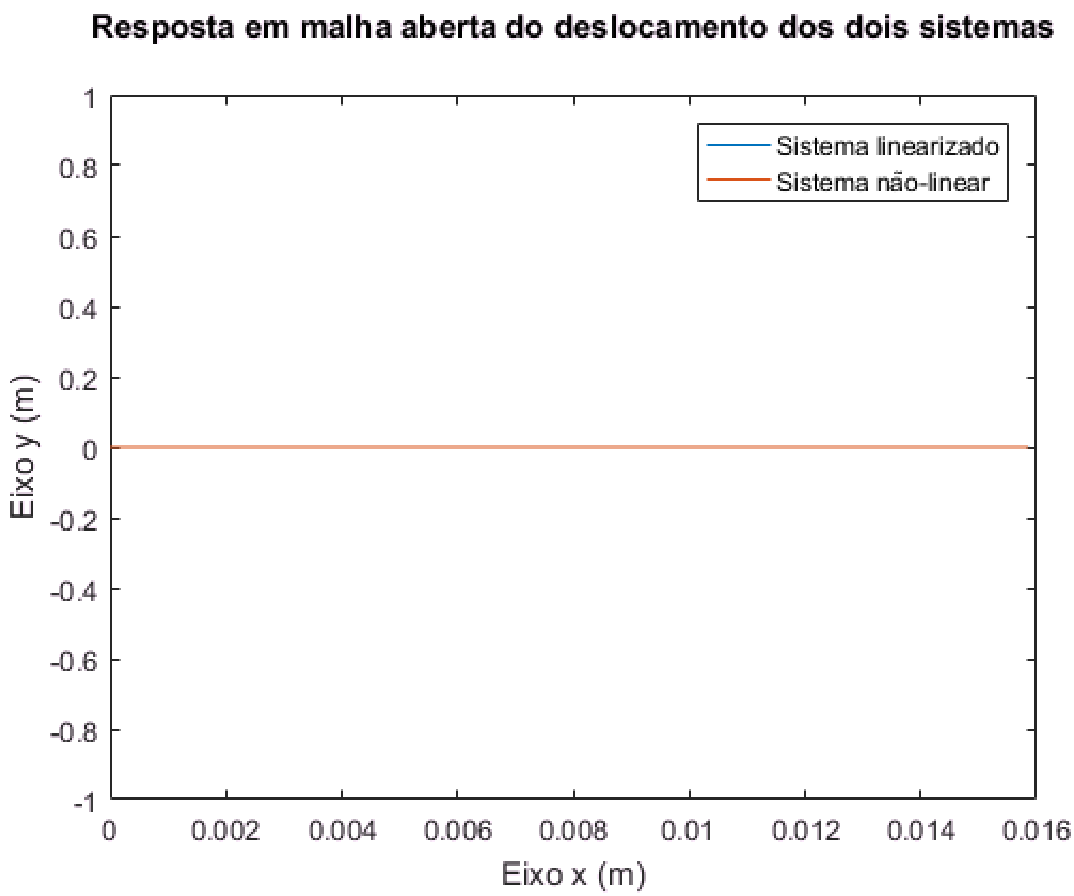
Gerando gráficos comparativos para cada entrada atuando sozinha com 25% do esforço de controle:

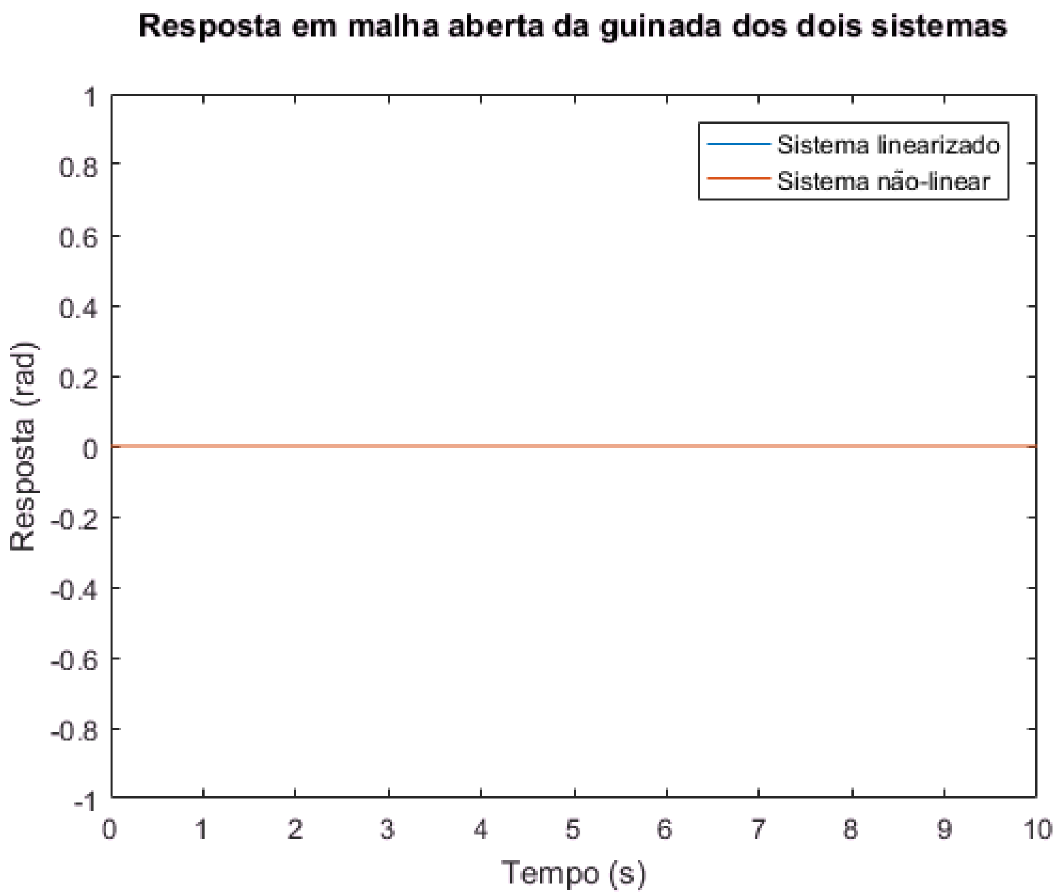
Fx:

```
fx = 0.01*2.5e5;
fy = 0;
mz = 0;

[xg_linear, yg_linear, psi_linear] = sim_linear(t);
[xg_non_linear, yg_non_linear, psi_non_linear] = sim_non_linear(t);

i=plota_graficos(psi_linear, yg_non_linear, xg_non_linear, xg_linear, yg_linear, psi_non_linea
```



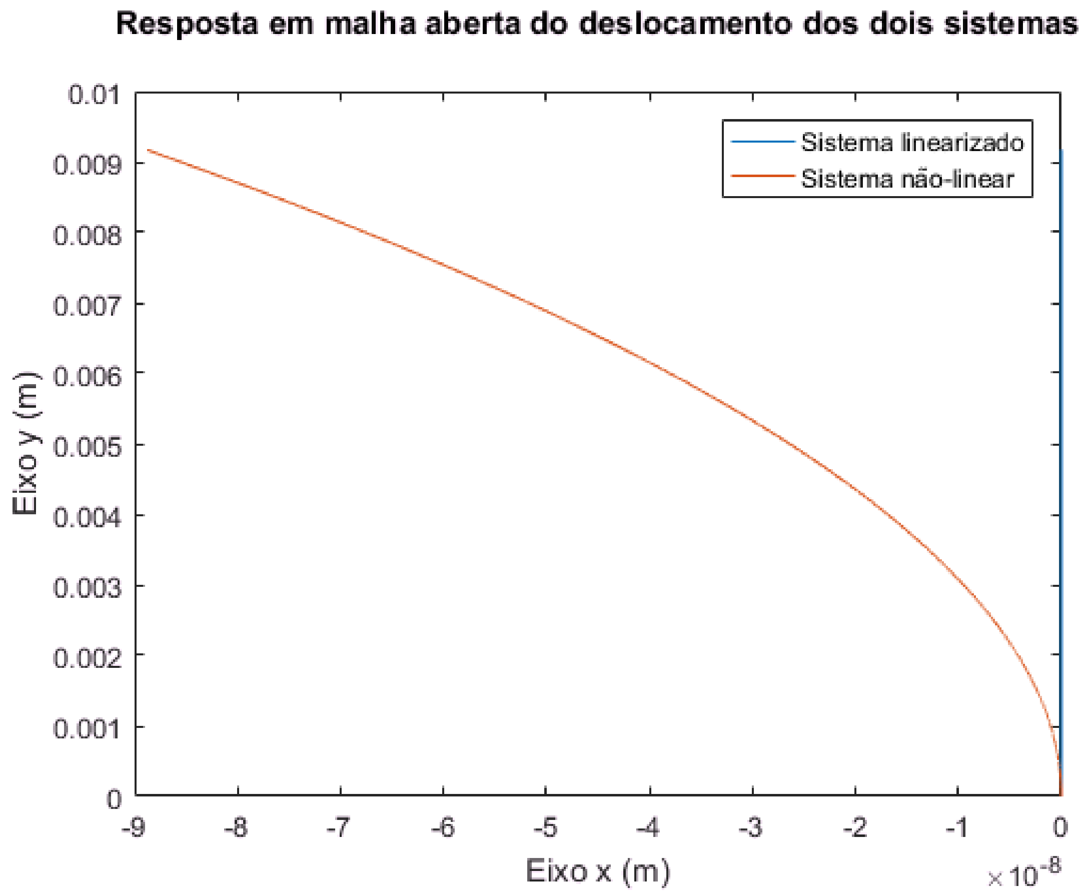


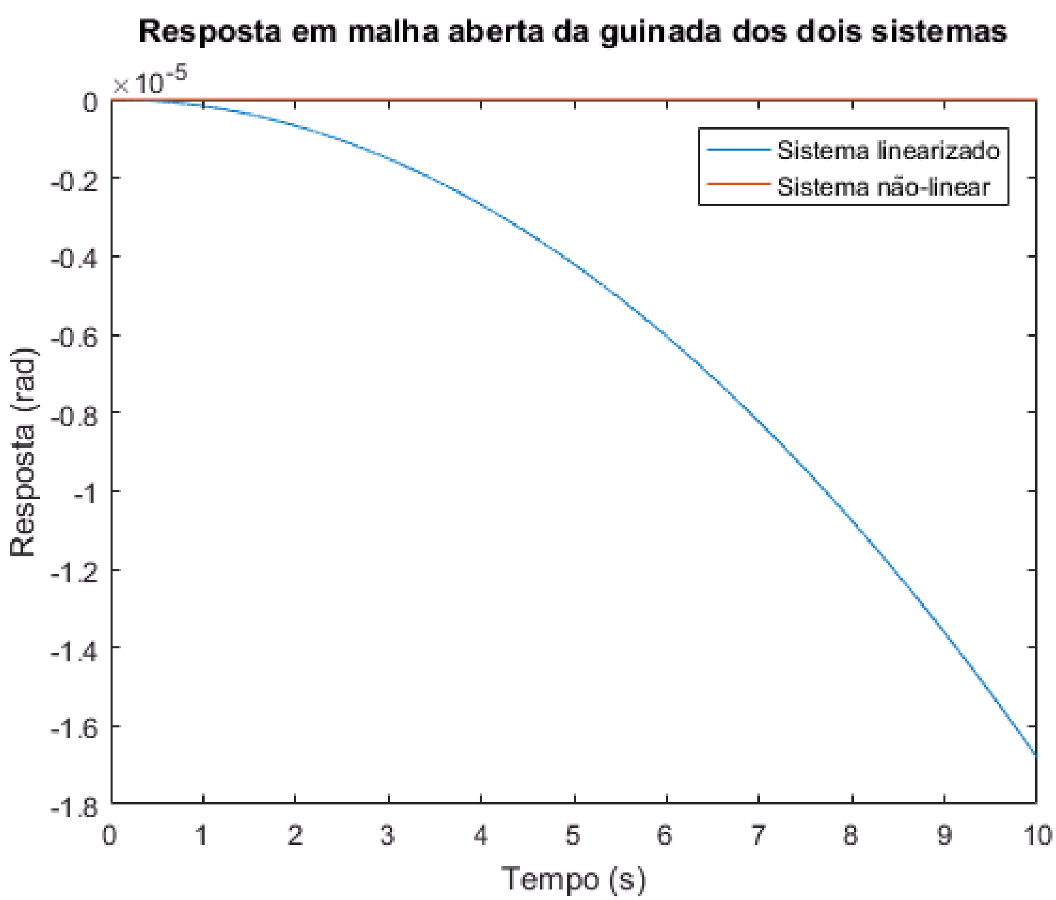
Fy:

```
fx = 0;
fy = 0.01*2.5e5;
mz = 0;

[xg_linear, yg_linear, psi_linear] = sim_linear(t);
[xg_non_linear, yg_non_linear, psi_non_linear] = sim_non_linear(t);

i=plota_graficos(psi_linear, yg_non_linear, xg_non_linear, xg_linear, yg_linear, psi_non_linea
```


Resposta em malha aberta do deslocamento dos dois sistemas
Sistema linearizado
Sistema não-linear
Eixo y (m)
Eixo x (m)
$\times 10^{-8}$
0.01
0.009
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0
-9
-8
-7
-6
-5
-4
-3
-2
-1
0

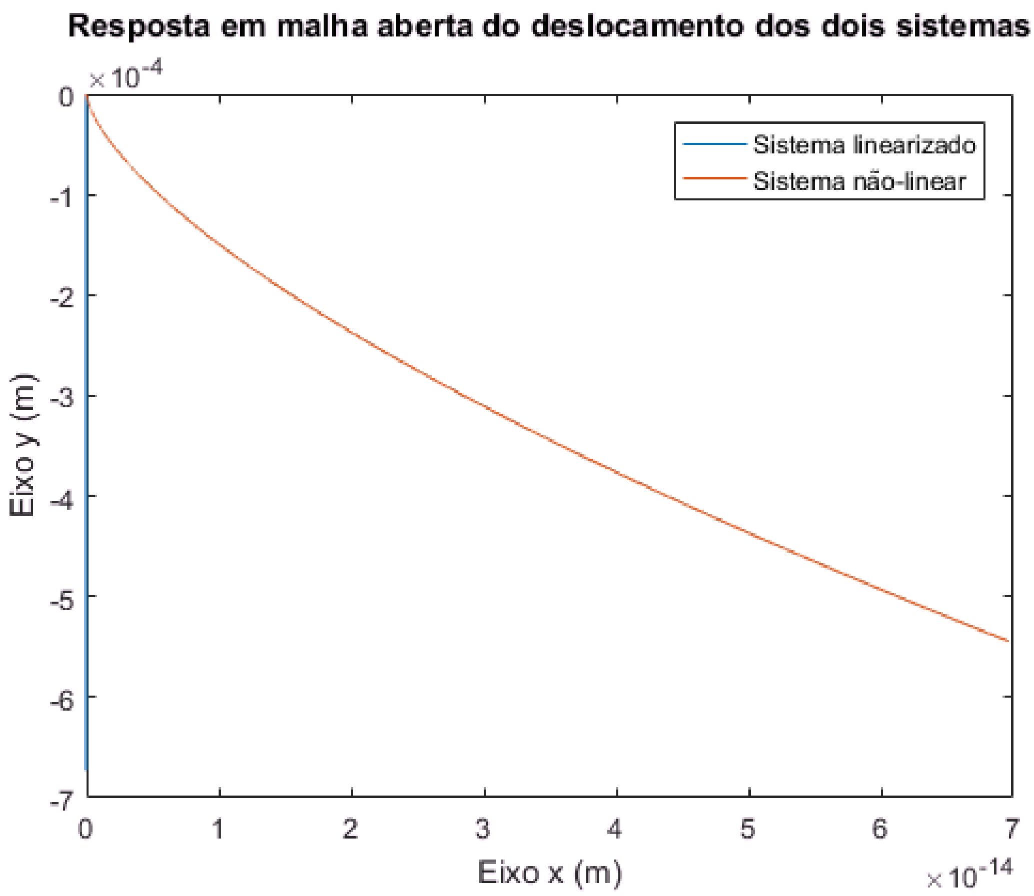

Resposta em malha aberta da guinada dos dois sistemas
$\times 10^{-5}$
Sistema linearizado
Sistema não-linear
Resposta (rad)
Tempo (s)
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
-1.4
-1.6
-1.8
0
1
2
3
4
5
6
7
8
9
10

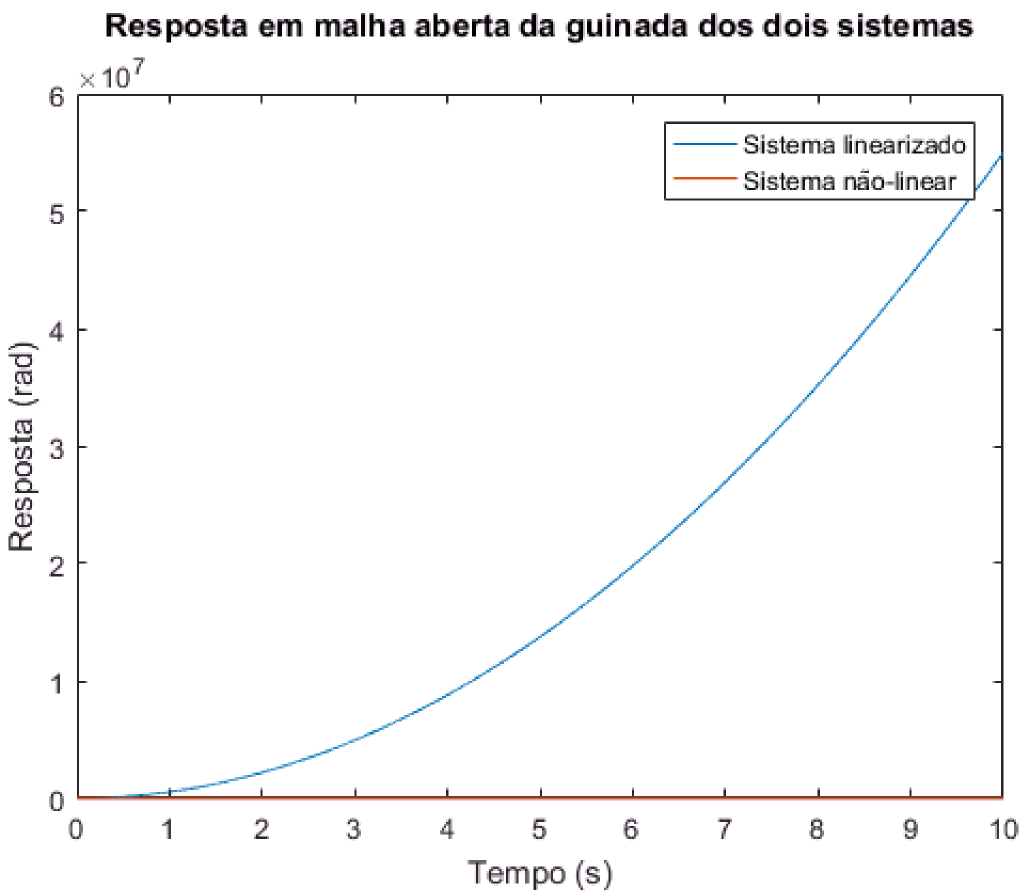
Fz:

```
fx = 0;
fy = 0;
mz = 0.01*1e7;

[xg_linear, yg_linear, psi_linear] = sim_linear(t);
[xg_non_linear, yg_non_linear, psi_non_linear] = sim_non_linear(t);

i=plota_graficos(psi_linear, yg_non_linear, xg_non_linear, xg_linear, yg_linear, psi_non_linea
```




Resposta em malha aberta da guinada dos dois sistemas
×10^7
6
5
4
3
2
1
0
Resposta (rad)
0
1
2
3
4
5
6
7
8
9
10
Tempo (s)
Sistema linearizado
Sistema não-linear