MODELICA
Elementos de Programação

PME3201 Laboratório de Simulações Numéricas

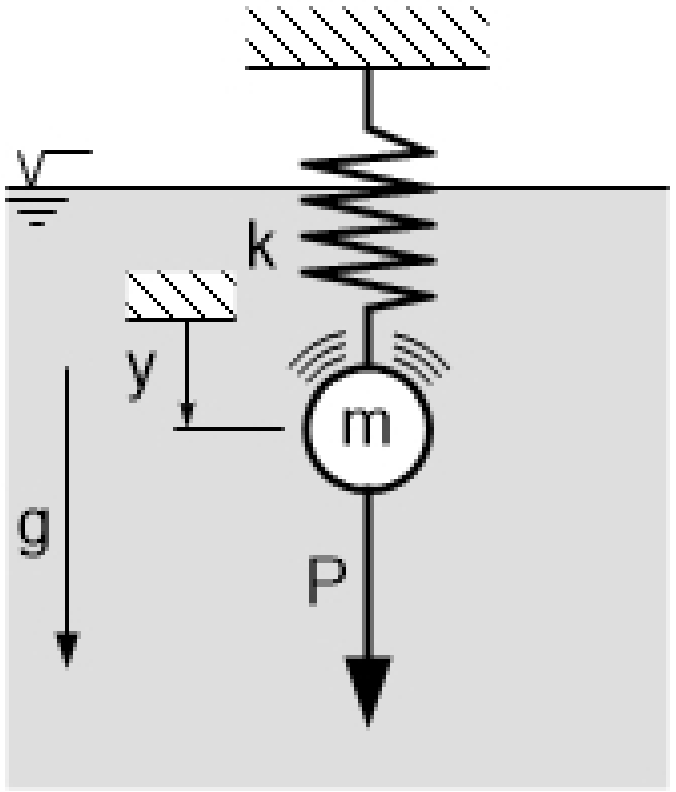
Prof. Ponge

Exemplos de Sistema Mecânico – Classes hereditárias

13.12.2017

Exemplo: sistema massa-mola-amortecedor

- Esfera em meio viscoso
- Parâmetros:
  - m = 10,0 kg
  - k = 1000,0 N/m
  - c = 50,0 kg/s
  - g = 10,0 m/s²



$$P = m \cdot g$$

$$F_k = k \cdot y$$

$$F_d = c \cdot v$$

$$m \cdot \frac{d\,v}{dt} = P - F_k - F_d$$

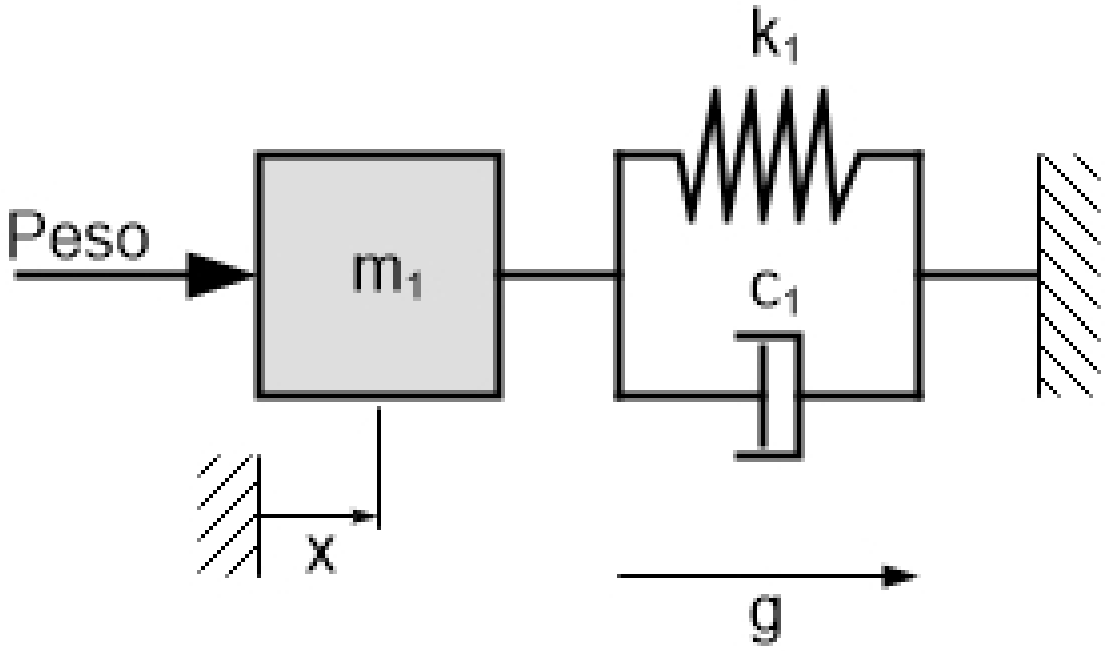
$$\frac{d\,y}{dt} = v$$

Biblioteca de Componentes

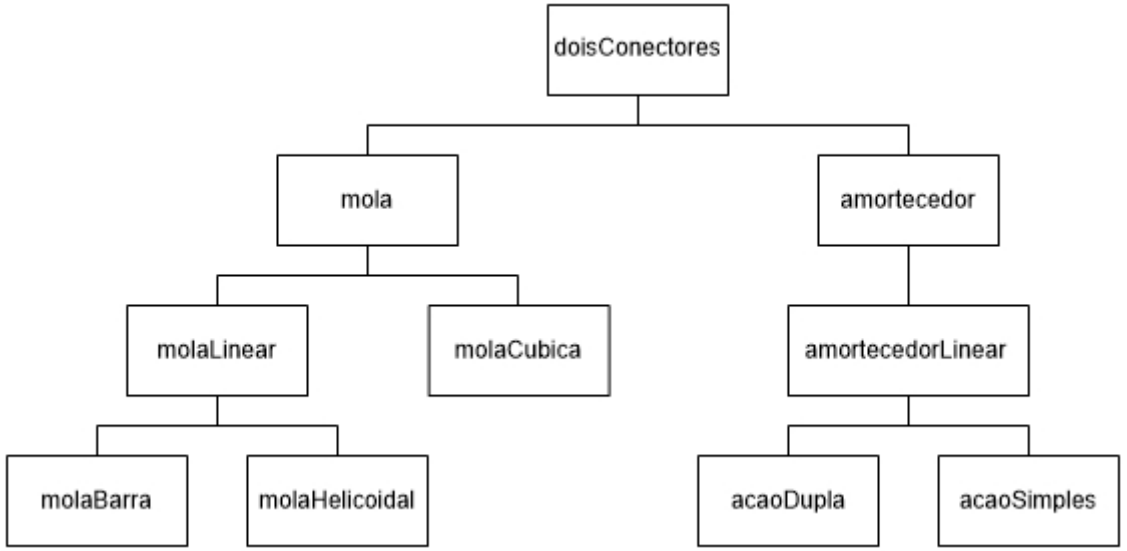
- Class connecter
  - acoplamento
- Class model
  - massa
  - mola
  - amortecedor
  - pontoFixo
  - forcaPeso
  - forcaHarmonica
- Função
  - funcaoHarmonica

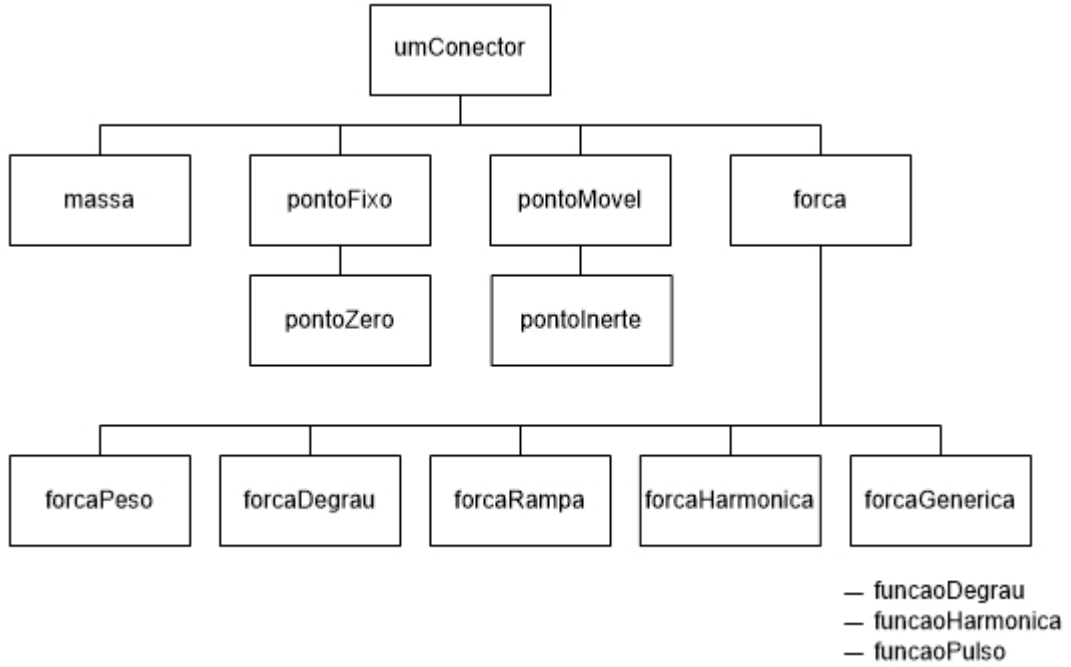
Sistema MKC com componentes da biblioteca

```
1  model SistemaMassaMolaAmortecedor "Sistema MKC com biblioteca de compoentes"
2    massa m1(m = 10.0);
3    mola k1(k = 1000.0);
4    amortecedor c1(c = 50.0);
5    pontoFixo p1;
6    forcaPeso Peso(m = 10.0);
7  equation
8    connect(Peso.b, m1.a);
9    connect(m1.b, k1.a);
10   connect(m1.b, c1.a);
11   connect(k1.b, p1.a);
12   connect(c1.b, p1.a);
13 end SistemaMassaMolaAmortecedor;
```

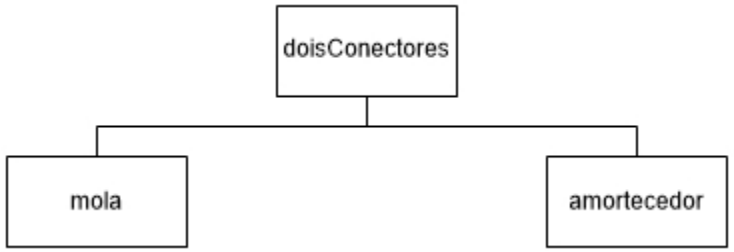


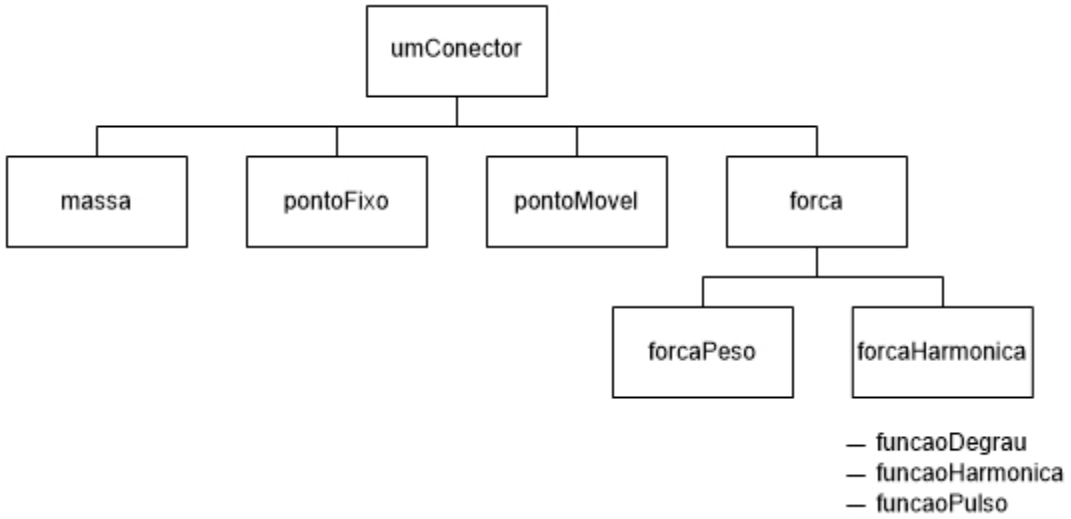
Estrutura de classes hereditárias





Estrutura de classes simplificada





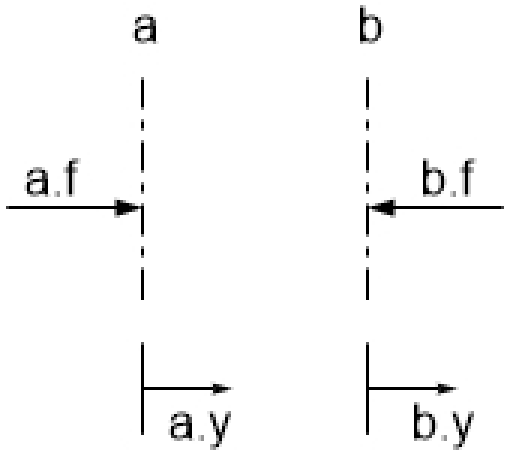
Acomplamento

```
1  connector acoplamento "Conector mecânico linear"
2    Real y;
3    flow Real f;
4  end acoplamento;
```

- Variável *entre - y*
  - Compatibilidade
  - tem valor igual para componentes que compartilham o mesmo conector
- Variável *através (flow) - f*
  - Continuidade (equilíbrio)
  - Somam para zero para componentes que compartilha o mesmo conector

doisConectores

```
1  partial model doisConectores
2    acoplamento a, b;
3    Real x, v, F, d;
4  equation
5    x = a.y;
6    F = a.f;
7    d = a.y - b.y;
8    v = der(d);
9    a.f + b.f = 0;
10 end doisConectores;
```



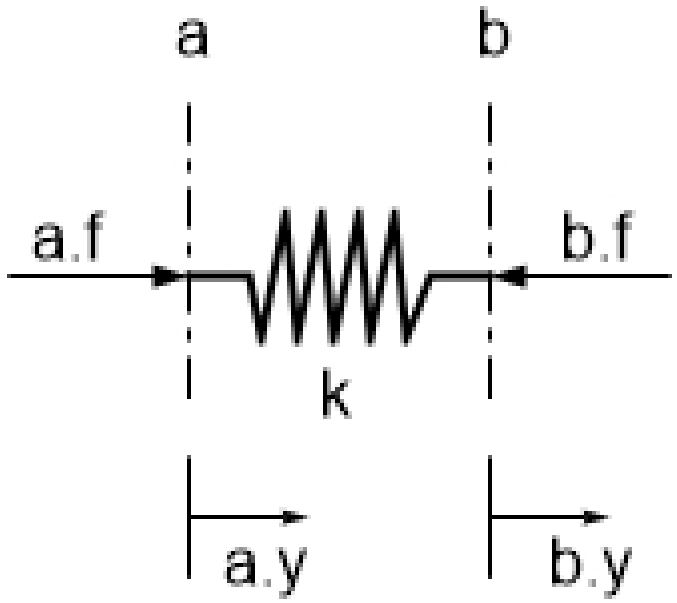
$$x = a.y$$

$$F = a.f \qquad a.f + b.f = 0$$

$$d = a.y - b.y \qquad v = der(d)$$

Mola

```
1  model mola
2    extends doisConectores;
3    parameter Real k = 1000.0;
4  equation
5    F = k * d "Lei de Hooke";
6  end mola;
```



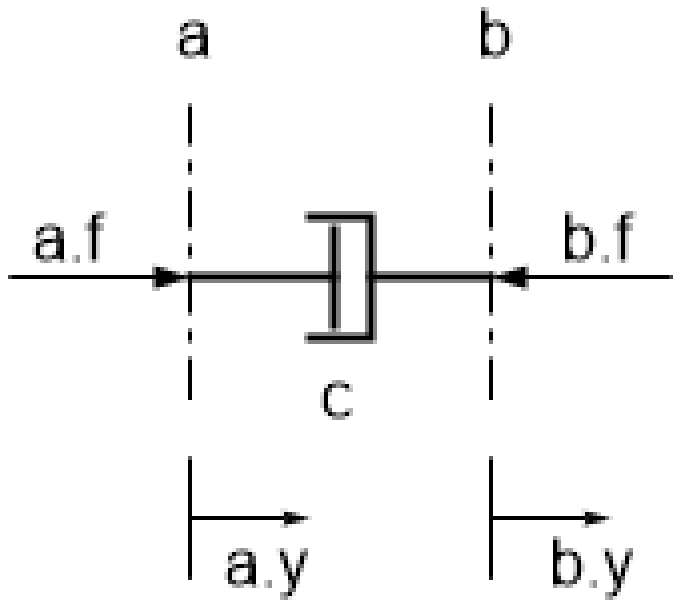
$$x = a.y \qquad F = k \cdot d$$

$$F = a.f \qquad a.f + b.f = 0$$

$$d = a.y - b.y$$

Amortecedor

```
1  model amortecedor
2    extends doisConectores;
3    parameter Real c = 100.0;
4  equation
5    F = c * v "Amortecimento Viscoso";
6  end amortecedor;
```



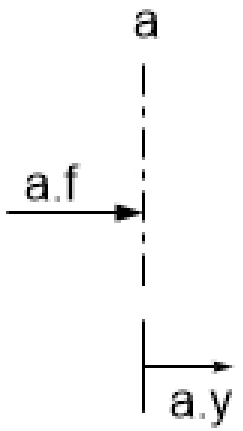
$$x = a.y \qquad F = c \cdot v$$

$$F = a.f \qquad a.f + b.f = 0$$

$$d = a.y - b.y \qquad v = der(d)$$

umConector

```
1  partial model umConector
2    acoplamento a;
3    Real x, v, F;
4  equation
5    x = a.y;
6    v = der(x);
7    F = a.f;
8  end umConector;
```



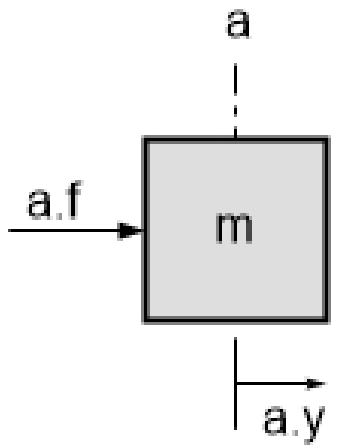
$$x = a.y$$

$$F = a.f$$

$$v = der(x)$$

Massa

```
1  model massa
2    extends umConector;
3    parameter Real m = 10.0;
4  equation
5    F = m * der(v);
6  end massa;
```



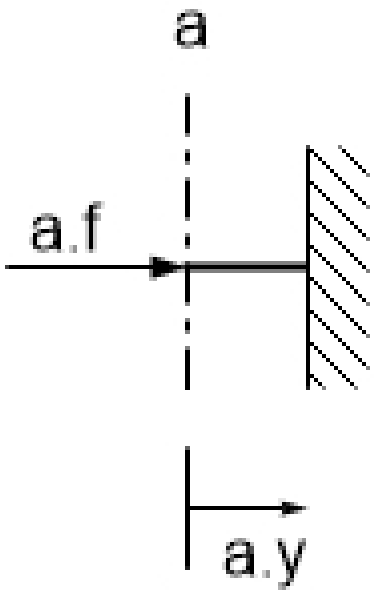
$$x = a.y \qquad F = m \cdot der(v)$$

$$F = a.f$$

$$v = der(x)$$

Ponto Fixo

```
1  model pontoFixo
2    extends umConector;
3    parameter Real X = 0.0;
4  equation
5    x = X;
6  end pontoFixo;
```



$$x = a.y \qquad x = 0$$

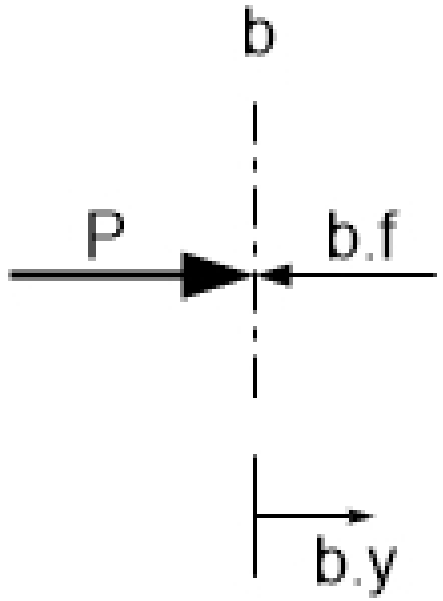
$$F = a.f$$

Força

```
1  partial model forca
2    extends umConector;
3    Real P;
4  equation
5    F + P = 0.0;
6  end forca;
```

Força Peso

```
1  model forcaPeso
2    extends forca;
3    parameter Real m = 10.0;
4    parameter Real g = 10.0;
5  equation
6    P = m * g;
7  end forcaPeso;
```

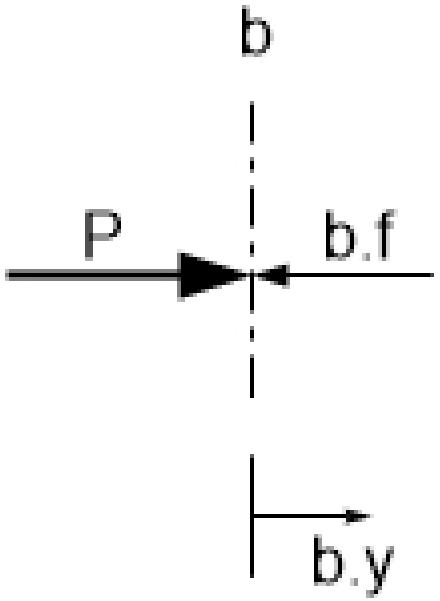


$$x = b.y \qquad P + F = 0$$

$$F = b.f$$

Força Harmônica

```
1  model forcaHarmonica
2    extends forca;
3    parameter Real Fo = 100.0;
4    parameter Real T = 2.0;
5    parameter Real to = 0.0;
6  equation
7    P = funcaoHarmonica(time, Fo, T, to);
8  end forcaHarmonica;
```



$$x = b.y \qquad P + F = 0$$

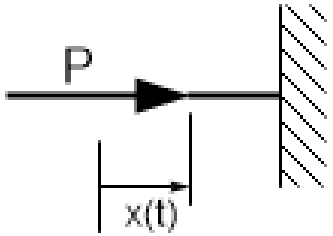
$$F = b.f$$

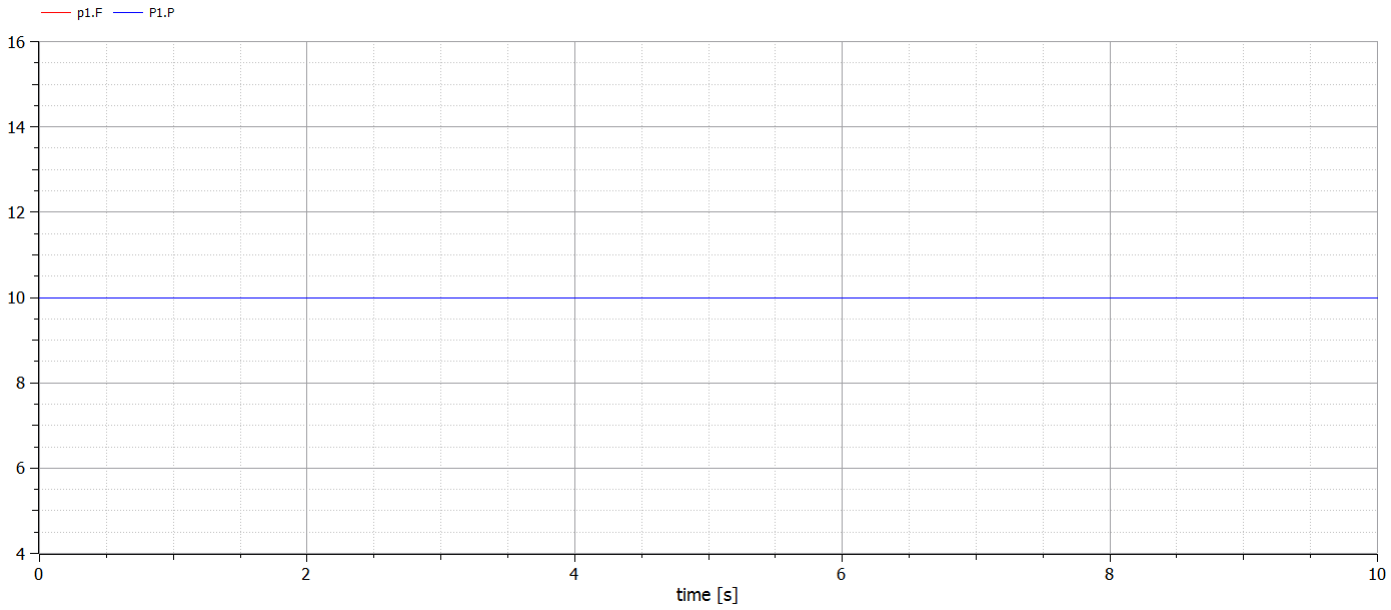
Função Harmônica

```
1  function funcaoHarmonica
2    input Real t;
3    input Real Fo;
4    input Real T;
5    input Real to;
6    output Real F;
7  protected
8    constant Real PI = Modelica.Constants.pi;
9  algorithm
10   F := Fo * sin(2 * PI * (t - to) / T);
11 end funcaoHarmonica;
```

Força aplicada ponto fixa

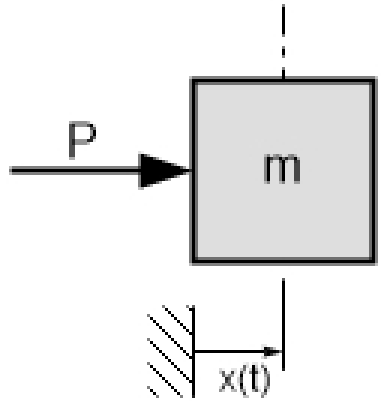
```
1  model forcaFixa
2    forcaPeso P1(m = 1.0);
3    pontoFixo p1;
4  equation
5    connect(P1.a, p1.a);
6  end forcaFixa;
```

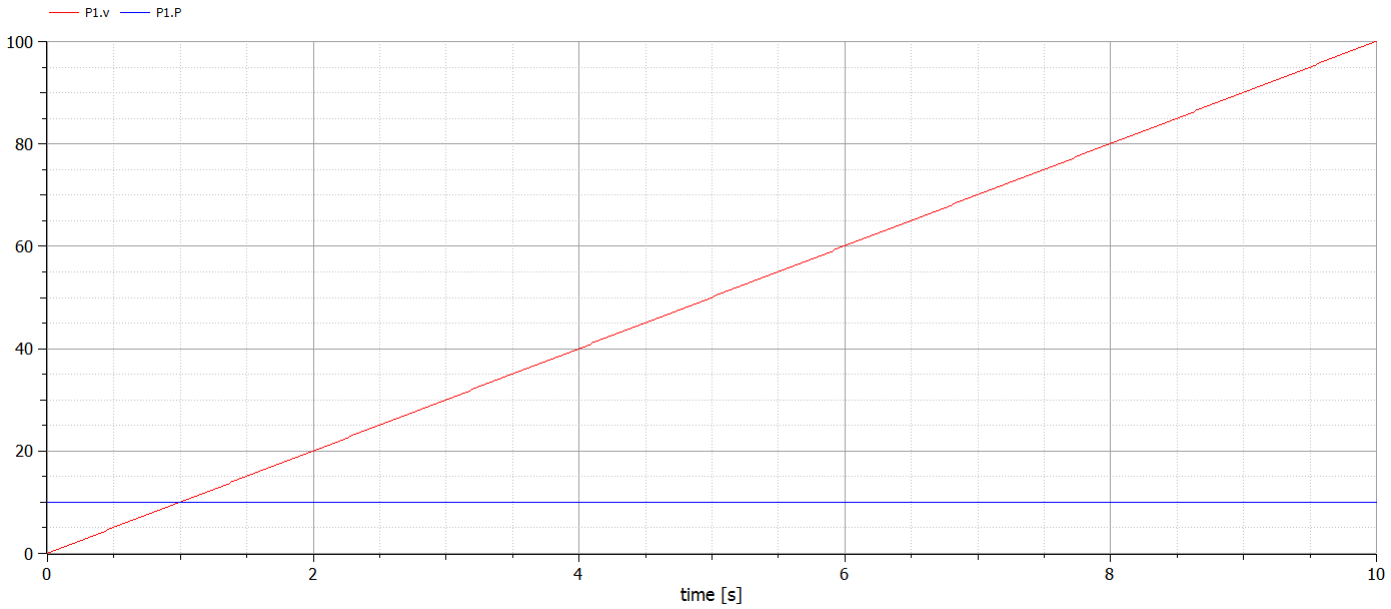




Força aplicada em uma massa

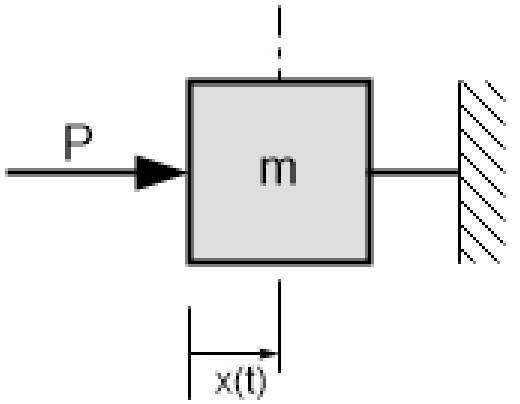
```
1  model forcaMassa
2    forcaPeso P1(m = 1.0);
3    massa m1(m = 1.0);
4  equation
5    connect(P1.a, m1.a);
6  end forcaMassa;
```

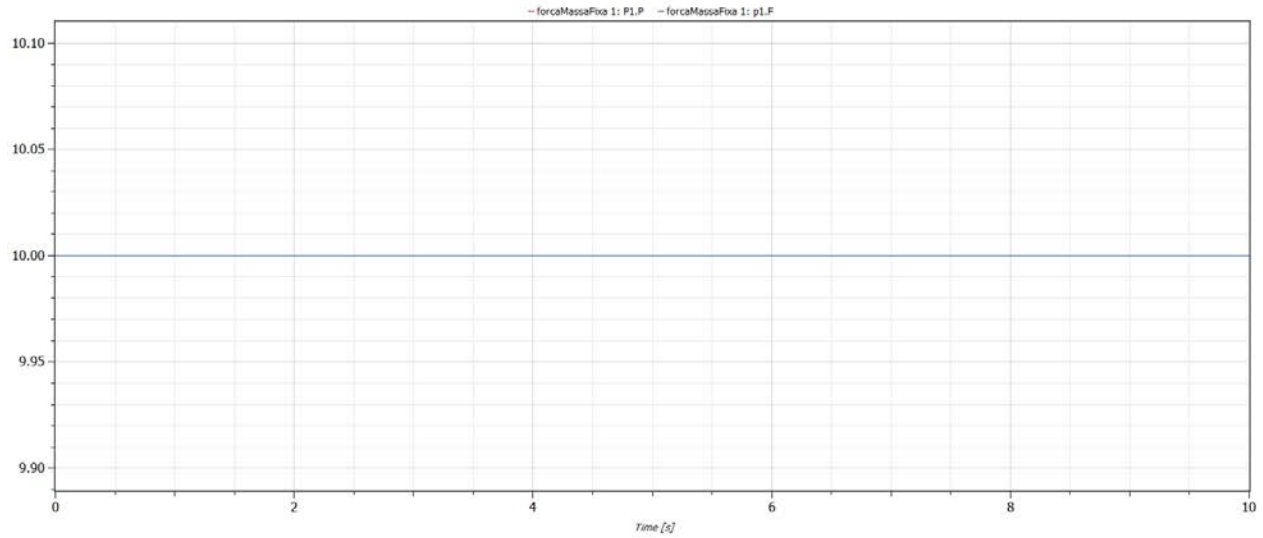




Força aplicada em massa fixa

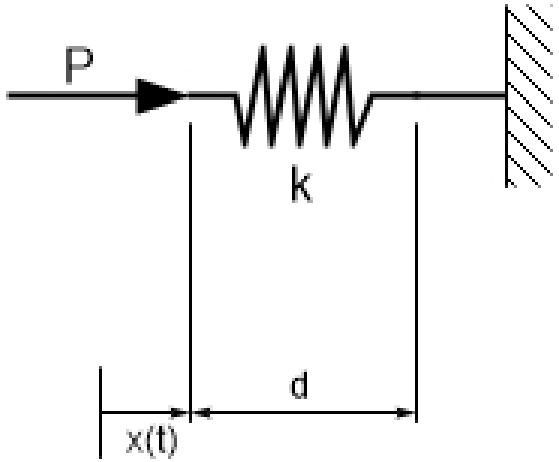
```
1  model forcaMassaFixa
2    forcaPeso P1(m = 1.0);
3    massa m1(m = 10.0);
4    pontoFixo p1;
5  equation
6    connect(P1.a, m1.a);
7    connect(m1.a, p1.a);
8  end forcaMassaFixa;
```

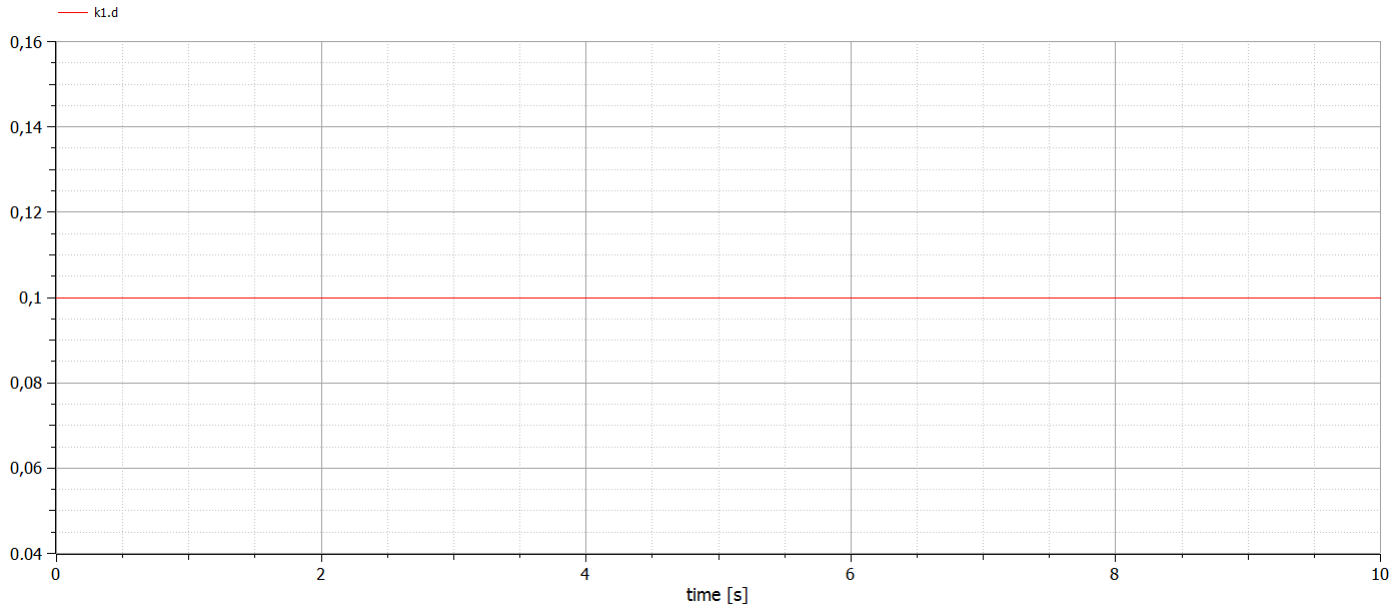




Força aplicada no extremo de uma mola

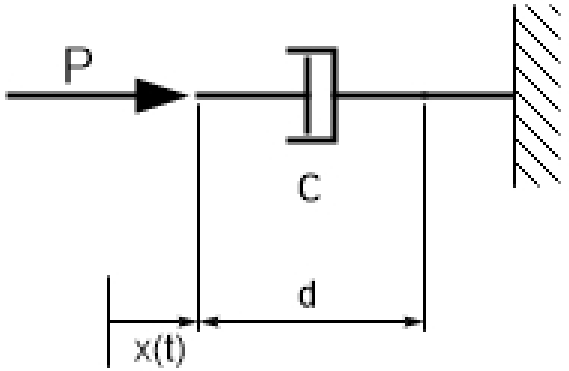
```
1  model forcaMola
2    mola k1(k = 1000.0);
3    pontoFixo p1;
4    forcaPeso P1(m = 10.0);
5  equation
6    connect(P1.a, k1.a);
7    connect(k1.b, p1.a);
8  end forcaMola;
```

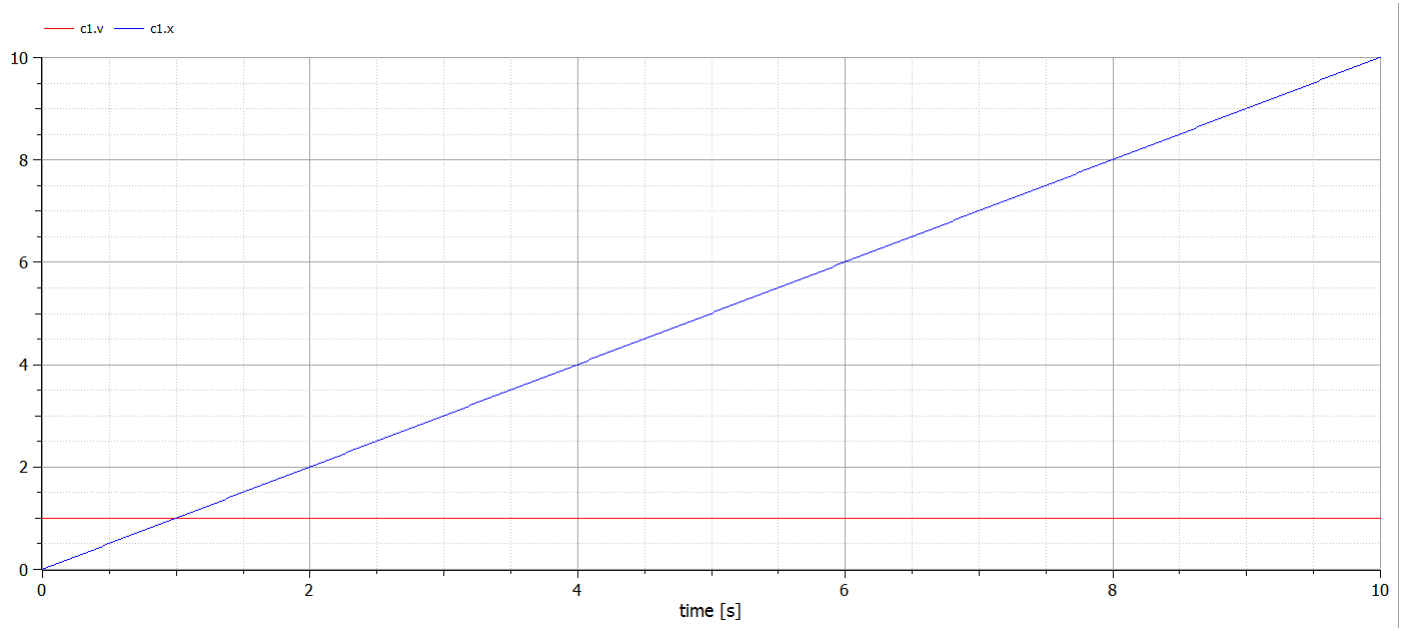




Força aplicada no extremo de um amortecedor

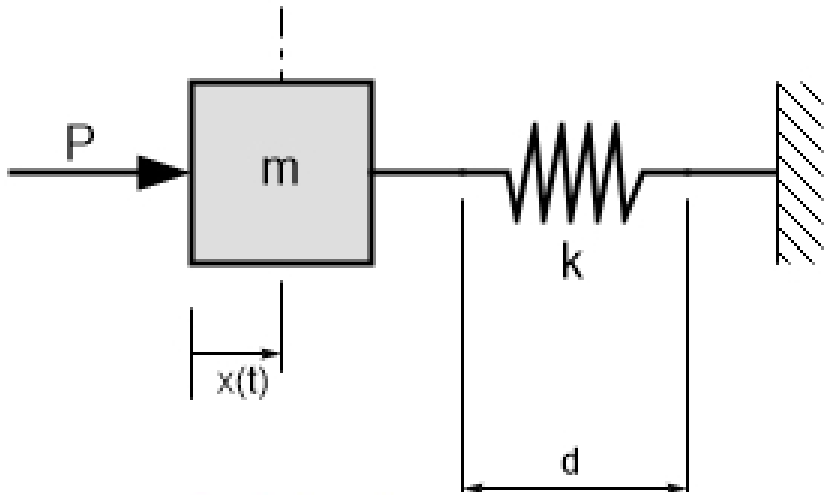
```
1  model forcaAmortecedor
2    amortecedor c1(c = 100.0);
3    pontoFixo p1;
4    forcaPeso P1(m = 10.0);
5  equation
6    connect(P1.a, c1.a);
7    connect(c1.b, p1.a);
8  end forcaAmortecedor;
```

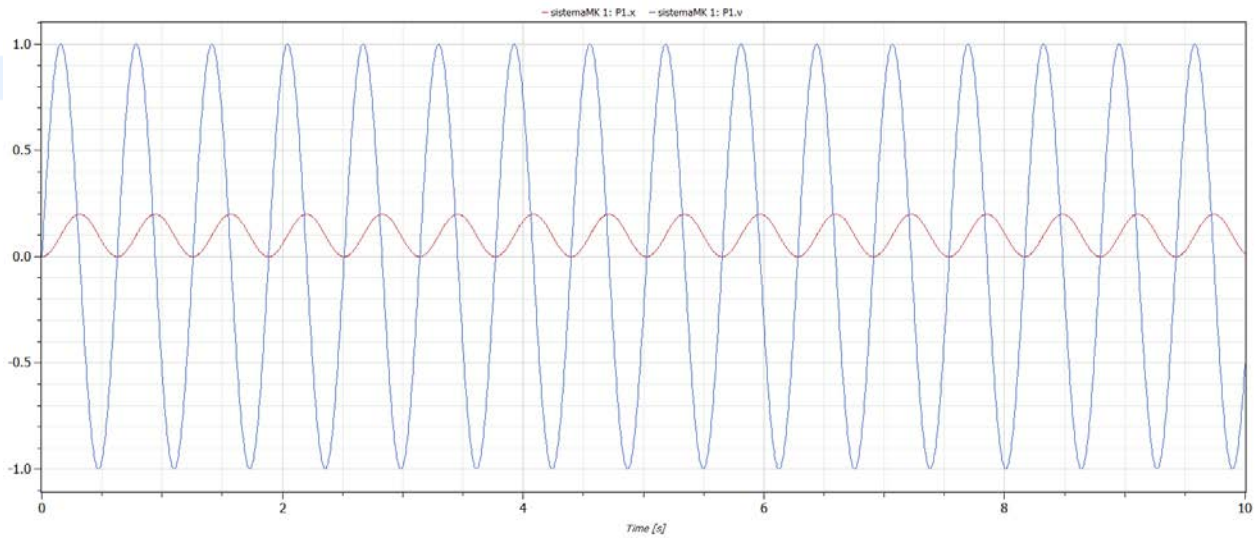




Sistema Massa-Mola com excitação degrau

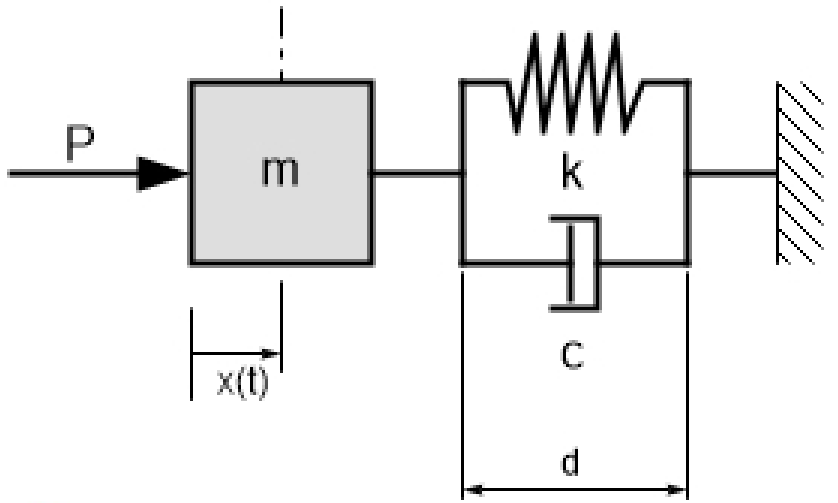
```
1  model sistemaMK
2    massa m1(m = 10.0);
3    mola k1(k = 1000.0);
4    pontoFixo p1;
5    forcaPeso P1(m = 10.0);
6    constant Real PI = Modelica.Constants.pi;
7    Real Wn = sqrt(k1.k / m1.m);
8    Real fn = Wn / 2 / PI;
9    Real T = 1 / fn;
10 equation
11   connect(P1.a, m1.a);
12   connect(m1.a, k1.a);
13   connect(k1.b, p1.a);
14 end sistemaMK;
```

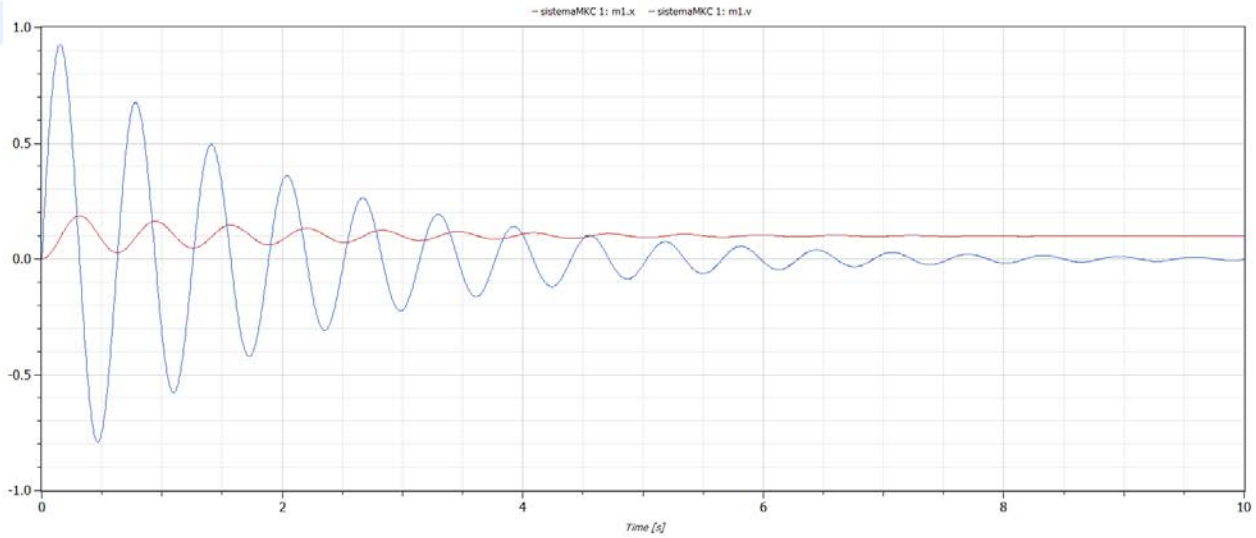




Sistema Massa-Mola-Amortecedor excitação degrau

```
1  model sistemaMKC
2    massa m1(m = 10.0);
3    mola k1(k = 1000.0);
4    amortecedor c1(c = 10.0);
5    pontoFixo p1;
6    forcaPeso P1(m = 10.0);
7  equation
8    connect(P1.a, m1.a);
9    connect(m1.a, k1.a);
10   connect(m1.a, c1.a);
11   connect(k1.b, p1.a);
12   connect(c1.b, p1.a);
13 end sistemaMKC;
```





Sistema Massa-Mola-Amortecedor excitação harmônica

```
1  model sistemaMKC_harmonico
2    massa m1(m = 10.0);
3    mola k1(k = 1000.0);
4    amortecedor c1(c = 10.0);
5    pontoFixo p1;
6    forcaHarmonica P1(Fo = 100.0, T = 4.0, to = 0.0);
7  equation
8    connect(P1.a, m1.a);
9    connect(m1.a, k1.a);
10   connect(m1.a, c1.a);
11   connect(k1.b, p1.a);
12   connect(c1.b, p1.a);
13 end sistemaMKC_harmonico;
```