2019.1-ExameFinal-Gabarito

July 18, 2019

1 Questão 1

1.1 Massa equivalente

Tomando como coordenada generalizada o deslocamento vertical da massa à esquerda, que vamos chamar de x, precisamos calcular a massa e amortecimento equivalentes em relação à esta coordenada.

Para calcular a massa equivalente, vamos usar a equivalência de energias cinéticas (ou podemos simplestemente usar os momentos de inércia de massa, mas isto é mais elegante.)

Chamando de l_l e l_r os comprimentos da barra à esquera e direita do pivô, respectivamente, temos que para um ângulo de rotação da barra θ pequeno,

$$y_l = l_l\theta, \quad y_r = l_r\theta,$$

então

$$\frac{y_l}{l_l} = \frac{y_r}{l_r}$$

ou

$$y_r = y_l\frac{l_r}{l_l}.$$

A energia cinética da massa à direita é então

$$T_r = \frac{1}{2}m_r\dot{y}_r^2 = \frac{1}{2}m_r\dot{y}_l^2\left(\frac{l_r}{l_l}\right)^2 = \frac{1}{2}m_r\left(\frac{l_r}{l_l}\right)^2\dot{y}_l^2,$$

É óbvio então que massa equivalente na coordenada y_l é

$$m_{\text{eq}} = m_r\left(\frac{l_r}{l_l}\right)^2\dot{y}_l^2.$$

Podemos calcular já esta valor,

```
In [87]: ll = 0.15
         lr = 0.25
         ml=2
         mr=1
         meq = mr*(lr/ll)**2
         print(meq)
```

```
2.77777777777778
```

A massa total na coordenada generalizada y_l é então,

```
In [88]: m = ml+meq
         print(m)
```

```
4.77777777777778
```

Podemos então calcular a frequência natural,

```
In [89]: k = 1720e3
         wn = sqrt(k/m)
         print(wn)
```

```
600.000000000000
```

1.2 Amortecimento equivalente

Podemos calcular o amortecimento equivalente considerando que ele é transformado da mesma forma que uma mola na mesma configuração, ou através da equivalência de potências instantâneas. Vamos usar a última opção só para não dizer que nunca fizemos isto. Quem fez pela analogia com molas está correto também, claramente.

A potência instantânea dissipada no amortecedor é $F_v\dot{y}$, onde F_v é a força viscosa, $F_v = c\dot{y}$, assim, a potência instantânea é dada por $c\dot{y}^2$. No caso, a potência no amortecedor é

$$H = c\dot{y}_r^2 = c\dot{y}_l^2\left(\frac{l_r}{l_l}\right)^2 = \frac{1}{2}c\left(\frac{l_r}{l_l}\right)^2\dot{y}_l^2,$$

então claramente o amortecimento equivalente é

$$c_{\text{eq}} = c\left(\frac{l_r}{l_l}\right)^2.$$

No caso então,

```
In [90]: c = 2750
         ceq = c*(lr/ll)**2
         print(ceq)
```

```
7638.88888888889
```

O amortecimento crítico é $c_c = 2m\omega_n$, então

```
In [91]: cc = 2*m*wn
         print(cc)
```

```
5733.33333333333
```

E a razão de amortecimento, $\zeta = c/c_c$, é

```
In [92]: zeta = ceq/cc
         print(zeta)
```

```
1.33236434108527
```

1.3 Deslocamento

O sistema é, portanto, claramente superamortecido e temos que escolher a resposta adequada!

Temos no formulário que neste caso o deslocamento é dado por

$$x(t) = C_1 e^{\left(-\zeta+\sqrt{\zeta^2-1}\right)\omega_n t} + C_2 e^{\left(-\zeta-\sqrt{\zeta^2-1}\right)\omega_n t},$$

com

$$C_1 = \frac{x_0\omega_n\left(\zeta+\sqrt{\zeta^2-1}\right)+\dot{x}_0}{2\omega_n\sqrt{\zeta^2-1}}, \quad C_2 = \frac{-x_0\omega_n\left(\zeta-\sqrt{\zeta^2-1}\right)-\dot{x}_0}{2\omega_n\sqrt{\zeta^2-1}}.$$

Neste problema em particular, o deslocamento inicial é nulo, e as fórmulas para C_1 e C_2 simplificam-se para

$$C_1 = \frac{\dot{x}_0}{2\omega_n\sqrt{\zeta^2-1}}, \quad C_2 = \frac{-\dot{x}_0}{2\omega_n\sqrt{\zeta^2-1}}.$$

Temos tudo o que é necessário para calcular estas expressões.

```
In [93]: v0 = -25
         sqz = sqrt(zeta**2-1)
         C1 = v0/(2*wn*sqz)
         C2 = -C1
         print(C1)
         print(C2)
```

```
-0.0236621037772585
0.0236621037772585
```

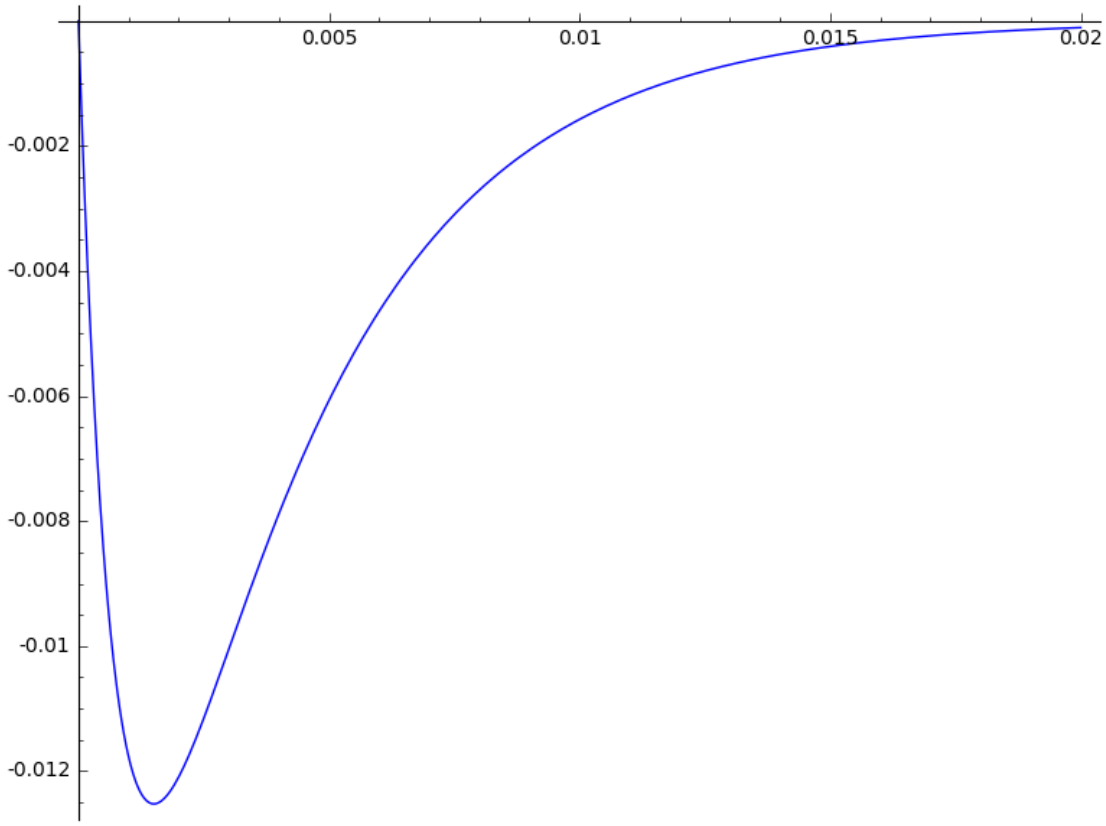
```
In [94]: var('t')
         x(t) = C1*exp((-zeta+sqz)*wn*t) + C2*exp((-zeta-sqz)*wn*t)
         show(x(t))
```

```
-0.0236621037772585*e^(-271.147740924597*t) + 0.0236621037772585*e^(-1327.68946837773*t)
```

Vamos plotar esta função para ver como fica (qualquer esquema razoável que **não tenha oscilação** e tenha o jeitão de um movimento superamortecido é aceitável na prova!)

```
In [95]: plot(x(t), (t, 0, 0.02))
```

Out[95]:



No tempo requisitado, o deslocamento é

```
In [96]: print(x(0.01))
```

```
-0.00157203596039829
```

aproximadamente 1,6 milímetros.

2 Questão 2

2.1 RIgidez Equivalente

Em primeiro lugar vamos calcular a rigidez equivalente da viga, considerando como coordenada generalizada o deslocamento vertical da massa concentrada.

Como a deflexão máxima é dada por

$$y(L) = \frac{WL^3}{3EI},$$

apelando para a definição imediata de rigidez $k = F/x$, podemos escrever

$$k_{\text{eq}} = \frac{3EI}{L^3}.$$

```
In [97]: L = 0.65
         E = 210e9
         b = 0.015
         h = 0.015
         I = b*h**3/12
         print(I)
         k = 3*E*I/L**3
         print(k)
```

```
4.21875000000000e-9
9677.96995903505
```

2.2 Frequência natural

A frequência natural é então

```
In [98]: m = 0.25
         wn = sqrt(k/m)
         print(wn)
         fn = wn/(2*N(pi))
         print(fn)
         taun = 1/fn
         print(taun)
```

```
196.753347712663
31.3142678583488
0.0319343247788367
```

2.3 Razão de amortecimento

Podemos usar o decremento logarítimico para calcular a razão de amortecimento.

```
In [99]: x1=1
         x251=0.80
         n = 250
         delta = ln(x1/x251)/n
         print(delta)
         zeta = delta/(2*N(pi))
         print(zeta)
```

```
0.000892574205256839
0.000142057596842946
```

2.4 Força aplicada

Claramente, como o sistema é linear e estamos interessados no regime permanente, podemos usar a Série de Fourier da função para calcular a resposta. É claro que na prova só é necessário calcular dois ou três harmônicos, como estou fazendo computacionalmente aqui posso exagerar no número de termos.

É dado no enunciado que o período do seno retificado é 0.04 segundos, com isto podemos calcular a frequência fundamental ω_0,

```
In [100]: T = 0.18
          w0 = 2*N(pi)/T
          print(w0)
```

```
34.9065850398866
```

Esta é uma série em cossenos apenas, cujos termos (sem as constantes, depois colocamos) são dados por,

```
In [101]: var('n, t')
          term(n, t) = cos (n*w0*t)/(4*n^2-1)
          show(term(1,t))
          show(term(2,t))
          show(term(3,t))
```

```
1/3*cos(34.9065850398866*t)


1/15*cos(69.8131700797732*t)


1/35*cos(104.719755119660*t)
```

e assim vai.

Isto server apenas para ilustração, para para calcular a resposta é apenas a amplitude de cada termo, que é dada pela expressão

$$A_n = \frac{4A}{\pi(4n^2-1)}.$$

```
In [102]: A=1.5
          An(n) = 4*A/(N(pi)*(4*n^2-1))
          show(An(n))
```

```
6.00000000000000/(12.5663706143592*n^2 - 3.14159265358979)
```

Vamos calcular um bom número de termos, lembrando que na prova 2 ou três são suficientes.

```
In [103]: nterm = 11
          nterm +=1
          Ans = [An(n) for n in range(1, nterm)]
          print(Ans)
```

```
[0.636619772367581, 0.127323954473516, 0.0545674090600784, 0.0303152272555991, 0.0192915082535
```

2.5 Resposta

Do formulário, sabemos que a resposta é dada por (os termos em seno são nulos para esta função)

$$x_p(t) = \frac{a_0}{2k} + \sum_{n=1}^{\infty} \frac{a_n/k}{\sqrt{(1-n^2r^2)^2 + (2\zeta nr)^2}} \cos(n\omega t - \varphi_j),$$

Vamos calcular o fator de amplificação para cada termo,

```
In [104]: r = w0/wn
          Mn(n) = 1/sqrt((1-(n*r)^2)**2+(2*zeta*n*r)**2)
          show(Mn(n))
          Mns = [Mn(n) for n in range(1, nterm)]
          print(Mns)
```

```
1/sqrt((-0.0314753425642045*n^2 + 1)^2 + (2.54073507961484e-9)*n^2)
```

```
[1.03249823426619, 1.14403564920711, 1.39524124228980, 2.01452650923825, 4.69226738213223, 7.5
```

Para calcular os coeficientes da série, precisamos calcular a_n/k,

```
In [105]: anks = [ an/k for an in Ans]
          print(anks)
```

```
[0.0000657803005240012, 0.0000131560601048003, 5.63831147348582e-6, 3.13239526304768e-6, 1.993
```

Os coeficientes da série são estes termos multiplicados pelo fator de amplificação de cada termo,

```
In [106]: cks = [ak*mk for ak, mk in zip (anks, Mns)]
          print(cks)
```

```
[0.0000679180441405304, 0.0000150510017630029, 7.86680470468319e-6, 6.31029329482187e-6, 9.353
```

Lembrem-se que temos que adicionar o termo constante!

```
In [107]: a0 = 2*A/N(pi)
          print(a0)
          c0 = a0/(2*k)
          print(c0)
```

```
0.954929658551372
0.0000493352253930009
```

Os termos tem frequência diferente então não podem ser somados diretamente, mas, é razoável, como uma estimativa de engenharia, somar as amplitudes com o termo constante para termos um valor que erra pela segurança.

```
In [121]: A = c0+sum(cks)
          print(A)
```

```
0.000169600992837461
```

O que vem daqui para a frente obviamente não é necesário para a prova, é apenas para ilustração.

Vamos plotar a resposta para ver com o que parece. Precisamos do ângulo de fase de cada componente para construir a resposta exata. A fórmula é

$$\phi_j = \arctan \frac{2\zeta nr}{1 - n^2 r^2},$$

que não é dada no formulário, mas não era estritamente necessária para uma resposta aproximada.

```
In [124]: phijs = [ arctan2(2*zeta*n*r, 1-n^2*r^2) for n in range(1, nterm)]
          print(phijs)
```

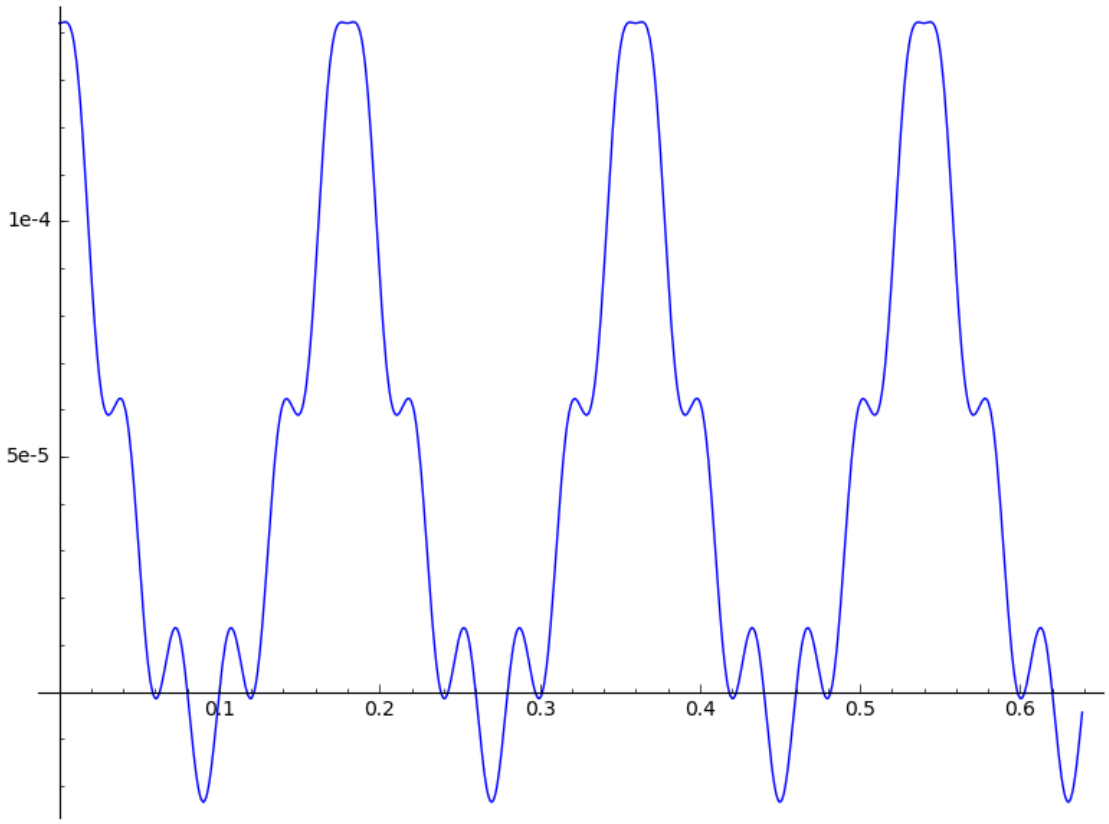
```
[0.0000520438012595188, 0.000115331846758211, 0.000210984356250357, 0.000406174525572233, 0.00
```

Como o amortecimento é muito baixo, a resposta de cada componente ou está praticamente em fase, se o harmônico está abaixo da frequência natural, ou praticamente em oposição de fase, para frequências acima da frequência natural.

```
In [130]: var('t')
          y(t) = c0
          for n, (c, phi)  in enumerate(zip(cks, phijs)):
              y(t) = y(t) + c*cos((n+1)*w0*t-phi)
          show(y)
          plot(y(t), (t, 0, 20*taun), plot_points=1000)
```

```
t |--> (1.45476551215813e-7)*cos(383.972435438753*t - 3.14139523165698) + (2.30305388909191e-7)
```

Out[130]:



A forma da resposta não é exatamente óbvia.

3 Questão 3

3.1 Equações de movimento

Vamos tomar como coordenadas generalizadas os deslocamentos x_1, do carro, e x_2, do cilindro.

Vamos usar as equações de Euler-Lagrange para determinar as equações de movimento deste sistema. A energia cinética total do sistema é

$$T = \frac{1}{2}m_1\dot{x}_1^2 + \frac{1}{2}m_2\dot{x}_3^2 + \frac{1}{2}J_0\dot{\theta}^2,$$

onde $\dot{\theta}$ é a velocidade angular do cilindro. A velocidade angular do cilindro depende da *diferença* entre as velocidades de translação do cilindro e do carro, $\dot{\theta} = (\dot{x}_1 - \dot{x}_2)/R$. Colocando isto na expressão acima,

$$T = \frac{1}{2}m_1\dot{x}_1^2 + \frac{1}{2}m_2\dot{x}_2^2 + \frac{1}{2}J_0\frac{(\dot{x}_1 - \dot{x}_2)^2}{R^2},$$

A energia potencial é dada por

$$V = \frac{1}{2}k_1x_1^2 + \frac{1}{2}k_2x_2^2.$$

As equações de EL são

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{q}_j} - \frac{\partial T}{\partial q_j} + \frac{\partial V}{\partial q_j} = Q_j,$$

com $q_1 = x_1$ e $q_2 = x_2$, e $Q_1 = F(t)$, já que ela já está projetada nesta coordenada.
Para x_1,

$$\frac{\partial T}{\partial \dot{x}_1} = m_1\dot{x}_1 + \frac{J_0}{R^2}(\dot{x}_1 - \dot{x}_2),$$

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{x}_1} = m_1\ddot{x}_1 + \frac{J_0}{R^2}(\ddot{x}_1 - \ddot{x}_2),$$

$$\frac{\partial T}{\partial x_1} = 0,$$

$$\frac{\partial V}{\partial x_1} = k_1 x_1.$$

Para x_2,

$$\frac{\partial T}{\partial \dot{x}_2} = m_2\dot{x}_2 - \frac{J_0}{R^2}(\dot{x}_1 - \dot{x}_2),$$

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{x}_2} = m_2\ddot{x}_2 - \frac{J_0}{R^2}(\ddot{x}_1 - \ddot{x}_2),$$

$$\frac{\partial T}{\partial x_2} = 0,$$

$$\frac{\partial V}{\partial x_2} = k_2 x_2.$$

Montando as equações para x_1 e x_2,

$$m_1\ddot{x}_1 + \frac{J_0}{R^2}(\ddot{x}_1 - \ddot{x}_2) + k_1 x_1 = F(t),$$

$$m_2\ddot{x}_2 - \frac{J_0}{R^2}(\ddot{x}_1 - \ddot{x}_2) + k_2 x_2 = 0,$$

que podem ser colocadas na forma matricial,

$$\begin{bmatrix} m_1 + J_0/R^2 & -J_0/R^2 \\ -J_0/R^2 & m_2 + J_0/R^2 \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} + \begin{bmatrix} k_1 & 0 \\ 0 & k2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} F(t) \\ 0 \end{bmatrix}$$

É interessante notar que o sistema tem acoplamento dinâmico, mas não elástico, o que é bem incomum em nossos exemplos. Lembrando que para o cilindro $J_0 = m_2R^2/2$, podemos reescrever isto para,

$$\begin{bmatrix} m_1 + m_2/2 & -m_2/2 \\ -m_2/2 & 3m_2/2 \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} + \begin{bmatrix} k_1 & 0 \\ 0 & k2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} F(t) \\ 0 \end{bmatrix}$$

3.2 Impedância mecânica

Como não há amortecimento, a matrix de impedância mecânica é real, não há fase na resposta e todas as grandezas são reais.

Calculando a matriz de impedância mecânica com $\boldsymbol{Z} = \boldsymbol{k} - \omega^2 \boldsymbol{m}$, ficamos com

```
In [70]: m1 = 5
         m2 = 0.75
         k1 = 1200
         k2 = 600

         K = matrix( [[k1, 0], [0, k2]])
         M = matrix( [[m1+0.5*m2, -0.5*m2], [-0.5*m2, m2]])
         show(K)
         show(M)
```

```
[1200    0]
[   0  600]
```

```
[  5.37500000000000 -0.375000000000000]
[-0.375000000000000  0.750000000000000]
```

```
In [71]: var('omega')
         Z = K - omega^2*M
         show(Z)
```

```
[-5.37500000000000*omega^2 + 1200         0.375000000000000*omega^2]
[       0.375000000000000*omega^2 -0.750000000000000*omega^2 + 600]
```

3.3 Frequências naturais

Calculando o determinante da matriz de impedância mecânica, encontramos a equação característica do problema,

```
In [72]: d = det(Z).expand()
         show(d)
```

```
3.89062500000000*omega^4 - 4125.00000000000*omega^2 + 720000
```

E resolvendo para os quadrados das frequências naturais, temos

```
In [73]: var('eta')
         eq = d.substitute(omega^4 == eta^2, omega^2==eta)
         show(eq)
         sols = solve(eq, eta)
         show(sols)
```

```
3.89062500000000*eta^2 - 4125.00000000000*eta + 720000
```

```
[eta == -800/83*sqrt(1033) + 44000/83, eta == 800/83*sqrt(1033) + 44000/83]
```

Ops, vamos ver isto como números reais!

```
In [74]: w1sq = N(sols[0].rhs())
         w2sq = N(sols[1].rhs())
         print(w1sq)
         print(w2sq)
```

```
220.334290506252
839.906673349170
```

E as frequências naturais são

```
In [75]: w1 = sqrt(w1sq)
         w2 = sqrt(w2sq)
         print(w1)
         print(w2)
```

```
14.8436616273159
28.9811434099687
```

3.4 Resposta

Para calcular a resposta, basta calcular $\boldsymbol{X} = \boldsymbol{Z}^{-1}(\omega)\boldsymbol{F}$. A frequência da força de acionamento é 20 rad/s, e o vetor de forças aplicadas poder ser escrito como,

```
In [76]: F = vector([150, 0])
         show(F)
```

```
(150, 0)
```

Para a frequência dada, a matriz de impedância mecânica é

```
In [77]: w = 20
         Zw = Z(omega=w)
         show(Zw)
```

```
[-950.000000000000  150.000000000000]
[ 150.000000000000  300.000000000000]
```

```
In [78]: show(Zw.inverse())
```

```
[-0.000975609756097561  0.000487804878048780]
[ 0.000487804878048780   0.00308943089430894]
```

```
In [79]: X = Zw.inverse()*F
         show(X)
```

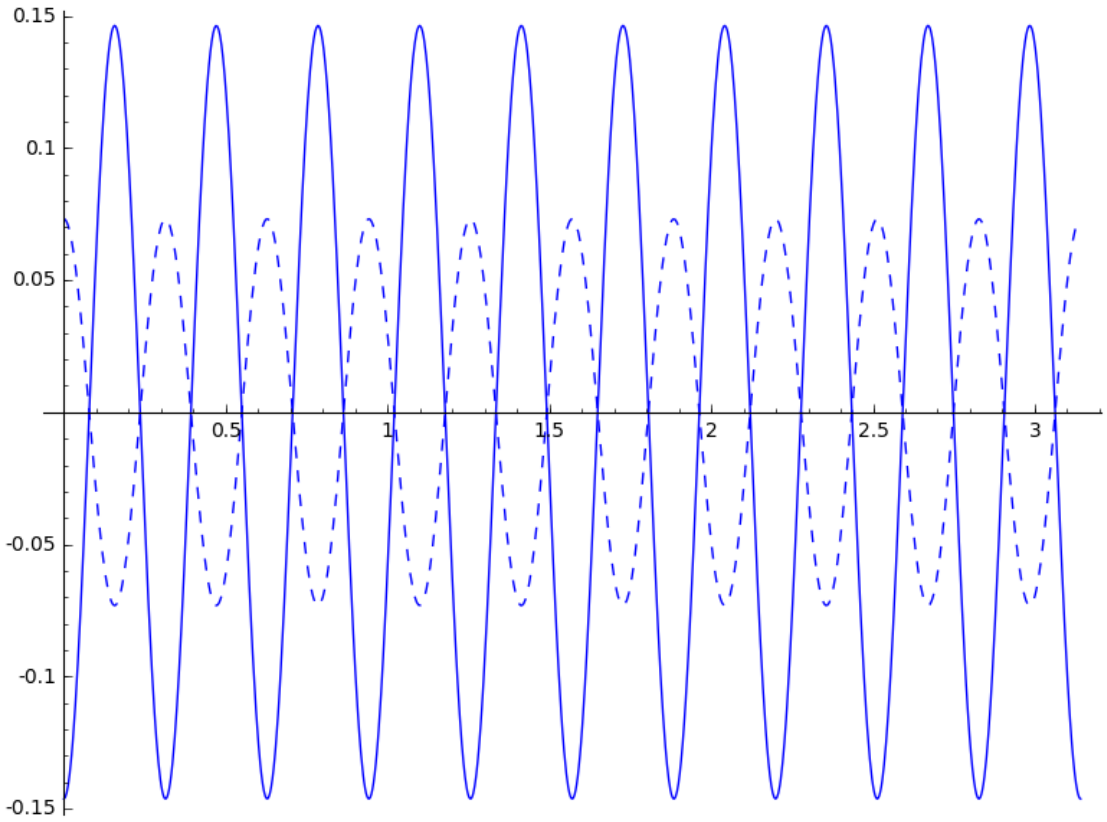
```
(-0.146341463414634, 0.0731707317073171)
```

Como as respostas estão em fase e na mesma frequência que a força de excitação, os deslocamentos de cada massa são

```
In [80]: x1(t) = X[0]*cos(w*t)
         x2(t) = X[1]*cos(w*t)
```

Plotando para ver como fica,

```
In [81]: nt = 2*pi/w*10
         p1 = plot(x1(t), (t, 0, nt), linestyle='-')
         p2 = plot(x2(t), (t, 0, nt), linestyle='--')
         show(p1+p2)
```



Os movimentos estã em oposição de fase.

4 Questão 4

Em primeiro lugar temos que reconhecer qual é a situação do fio. Para ambas as extremidades livres, os modos normais tem a forma, segundo a tabela dada no enunciado,

$$w(x) = C_n \cos \frac{n\pi x}{l},$$

para $n = 1$, esta função tem um nó, para $n = 2$ tem dois, e assim vai. É obvio então que a configuração mostrada corresponde a $n = 3$.

Para esta configuração, as frequências naturais são dadas por,

$$\omega_n = \frac{n\pi c}{l},$$

E como temos o número do modo, a frequência natural e o comprimento do fio, podemos calcular a velocidade de propagação.

```
In [82]: n=3
         l=5
         f3=150
         w3=2*pi*f3
         print(N(w3))
```

```
942.477796076938
```

```
In [83]: c = w3*l/(n*pi)
         print(c)
```

```
500
```

Para um fio em tração, a velocidade de propagação é dada por $c = \sqrt{P/\rho}$, onde ρ é a densidade linear. Temos então

```
In [84]: rho_v = 7700
         d = 0.0015
         A = N(pi)*d^2/4
         l = 5
         rho = A*rho_v
         print(rho)
```

```
0.0136070231808608
```

```
In [85]: P = rho*c^2
         print(P)
```

```
3401.75579521520
```