FinalProject

December 1, 2022

1 ME314 Final Project Code

1.1 Colab link:

https://colab.research.google.com/drive/1Gx_hA0NcI14TWKwMdBTOm3ONs18sWiMW?usp=sharing

If anything is not showed up in pdf, please reference to colab link

[1]:

```python
# Import required packages
import sympy as sym
import numpy as np
import matplotlib.pyplot as plt
```

[2]:

```python
###############################################################################################
# If you're using Google Colab, uncomment this section by selecting the whole section and press
# ctrl+'/' on your and keyboard. Run it before you start programming, this will enable the nice
# LaTeX "display()" function for you. If you're using the local Jupyter environment, leave it alone
###############################################################################################
# import sympy as sym
# import numpy as np
# import matplotlib.pyplot as plt
# def custom_latex_printer(exp,**options):
#     from google.colab.output._publish import javascript
#     url = "https://cdnjs.cloudflare.com/ajax/libs/mathjax/3.1.1/latest.js?config=TeX-AMS_HTML"
#     javascript(url=url)
#     return sym.printing.latex(exp,**options)
# sym.init_printing(use_latex="mathjax",latex_printer=custom_latex_printer)
```

[3]:

```python
def SE3_Inv(M):
  t = sym.Matrix([M[0,3],M[1,3],M[2,3]])
  R = sym.Matrix([[M[0,0],M[0,1],M[0,2]],[M[1,0],M[1,1],M[1,2]],[M[2,0],M[2,1],M[2,2]]])

  RT = R.transpose()
  tT = -RT*t
  MT = sym.
 ↪Matrix([[RT[0,0],RT[0,1],RT[0,2],tT[0]],[RT[1,0],RT[1,1],RT[1,2],tT[1]],[RT[2,0],RT[2,1],RT
  return MT

def AddHat(X):
  H = sym.Matrix([
      [0,-X[2],X[1],X[3]],
      [X[2],0,-X[0],X[4]],
      [-X[1],X[0],0,X[5]]
      [0,0,0,1]
  ])

  return H


def SE3(theta,x,y):
  g1 = sym.Matrix([[sym.cos(theta),-sym.sin(theta),0,x],[sym.sin(theta),sym.
 ↪cos(theta),0,y],[0,0,1,0],[0,0,0,1]])
  return g1

def UnHat(X):
  return sym.Matrix([X[0,3],X[1,3],X[2,3],X[2,1],X[0,2],X[1,0]])


def CreateM(m,J):
  M = sym.zeros(6,6)
  M[0,0] = m
  M[1,1] = m
  M[2,2] = m
  M[5,5] = J
  return sym.Matrix(M)
```

```
<>:14: SyntaxWarning: list indices must be integers or slices, not tuple;
perhaps you missed a comma?
<>:14: SyntaxWarning: list indices must be integers or slices, not tuple;
perhaps you missed a comma?
/tmp/ipykernel_91050/4029661927.py:14: SyntaxWarning: list indices must be
integers or slices, not tuple; perhaps you missed a comma?
  [-X[1],X[0],0,X[5]]
```

[4]:

```
# Final Project
# Big square box uses frame B, small sqaure jack uses frame A
# L for box is 4
# Diagonal for jack is 1

L2 = 6
```

```
L1 = 1
M = 0.5
m = 10
g = 9.8
J = 1
k1 = 20
#define variables x1,y1,theta1,x2,y2,theta2

t= sym.symbols('t')

theta1 = sym.Function(r'\theta_1')(t)
theta1dot = theta1.diff(t)
theta1ddot = theta1dot.diff(t)

theta2 = sym.Function(r'\theta_2')(t)
theta2dot = theta2.diff(t)
theta2ddot = theta2dot.diff(t)

x1 = sym.Function(r'x_1')(t)
x1dot = x1.diff(t)
x1ddot = x1dot.diff(t)

x2 = sym.Function(r'x_2')(t)
x2dot = x2.diff(t)
x2ddot = x2dot.diff(t)

y1 = sym.Function(r'y_1')(t)
y1dot = y1.diff(t)
y1ddot = y1dot.diff(t)

y2 = sym.Function(r'y_2')(t)
y2dot = y2.diff(t)
y2ddot = y2dot.diff(t)

#All 1s are in A frame and 2s are in B frame
#Build Frames g_wa and g_wb

g_wa = sym.Matrix([
[ sym.cos(theta1), -sym.sin(theta1), 0.,x1],
[ sym.sin(theta1), sym.cos(theta1), 0.,y1],
[0,0,1,0],
[0.,0., 0, 1.]])

print('\033[1m g_wa: ')
display(g_wa)

g_wb = sym.Matrix([
[ sym.cos(theta2), -sym.sin(theta2), 0.,x2],
[ sym.sin(theta2), sym.cos(theta2), 0.,y2],
[0,0,1,0],
[0.,0., 0, 1.]])

print('\033[1m g_wb: ')
display(g_wb)
#Build Frames for links in B

g_bb1 = sym.Matrix([
[ 1, 0, 0.,L2/2],
[ 0, 1, 0.,0],
[0,0,1,0],
[0.,0., 0, 1.]])

g_bb2 = sym.Matrix([
[ 1, 0, 0.,0],
[ 0, 1, 0.,-L2/2],
[0,0,1,0],
[0.,0., 0, 1.]])

g_bb3 = sym.Matrix([
[ 1, 0, 0.,-L2/2],
[ 0, 1, 0.,0],
[0,0,1,0],
[0.,0., 0, 1.]])

g_bb4 = sym.Matrix([
[ 1, 0, 0.,0],
[ 0, 1, 0.,L2/2],
[0,0,1,0],
[0.,0., 0, 1.]])

#Frame Transfer
g_wb1 = g_wb*g_bb1
g_wb2 = g_wb*g_bb2
g_wb3 = g_wb*g_bb3
g_wb4 = g_wb*g_bb4

# Build contact points on a frame

g_a_1 = SE3(0,L1/2,0)
g_a_2 = SE3(0,0,-L1/2)
g_a_3 = SE3(0,-L1/2,0)
g_a_4 = SE3(0,0,L1/2)

g_wa_1 = g_wa*g_a_1
```

```
g_wa_2 = g_wa*g_a_2
g_wa_3 = g_wa*g_a_3
g_wa_4 = g_wa*g_a_4

# Printing Transformation Matrix
print('\033[1m g_a_a1: ')
display(g_a_1)
print('\033[1m g_a_a2: ')
display(g_a_2)
print('\033[1m g_a_a3: ')
display(g_a_3)
print('\033[1m g_a_a4: ')
display(g_a_4)

print('\033[1m g_b_b1: ')
display(g_bb1)
print('\033[1m g_b_b2: ')
display(g_bb2)
print('\033[1m g_b_b3: ')
display(g_bb3)
print('\033[1m g_b_b4: ')
display(g_bb4)

print('\033[1m g_w_a1: ')
display(g_wa_1)
print('\033[1m g_w_a2: ')
display(g_wa_2)
print('\033[1m g_w_a3: ')
display(g_wa_3)
print('\033[1m g_w_a4: ')
display(g_wa_4)

print('\033[1m g_w_b1: ')
display(g_wb1)
print('\033[1m g_w_b22: ')
display(g_wb2)
print('\033[1m g_w_b3: ')
display(g_wb3)
print('\033[1m g_w_b4: ')
display(g_wb4)

# Transfer to box edge frame
g_b1w = SE3_Inv(g_wb1)
g_b2w = SE3_Inv(g_wb2)
g_b3w = SE3_Inv(g_wb3)
g_b4w = SE3_Inv(g_wb4)
```

```
g_b1a1 = g_b1w*g_wa_1
g_b1a2 = g_b1w*g_wa_2
g_b1a3 = g_b1w*g_wa_3
g_b1a4 = g_b1w*g_wa_4

g_b2a1 = g_b2w*g_wa_1
g_b2a2 = g_b2w*g_wa_2
g_b2a3 = g_b2w*g_wa_3
g_b2a4 = g_b2w*g_wa_4

g_b3a1 = g_b3w*g_wa_1
g_b3a2 = g_b3w*g_wa_2
g_b3a3 = g_b3w*g_wa_3
g_b3a4 = g_b3w*g_wa_4

g_b4a1 = g_b4w*g_wa_1
g_b4a2 = g_b4w*g_wa_2
g_b4a3 = g_b4w*g_wa_3
g_b4a4 = g_b4w*g_wa_4

print('\033[1m g_b1_a1: ')
display(sym.simplify(g_b1a1))
print('\033[1m g_b1_a2: ')
display(sym.simplify(g_b1a2))
print('\033[1m g_b1_a3: ')
display(sym.simplify(g_b1a3))
print('\033[1m g_b1_a4: ')
display(sym.simplify(g_b1a4))

print('\033[1m g_b2_a1: ')
display(sym.simplify(g_b2a1))
print('\033[1m g_b2_a2: ')
display(sym.simplify(g_b2a2))
print('\033[1m g_b2_a3: ')
display(sym.simplify(g_b2a3))
print('\033[1m g_b2_a4: ')
display(sym.simplify(g_b2a4))

print('\033[1m g_b3_a1: ')
display(sym.simplify(g_b3a1))
print('\033[1m g_b3_a2: ')
display(sym.simplify(g_b3a2))
print('\033[1m g_b3_a3: ')
display(sym.simplify(g_b3a3))
print('\033[1m g_b3_a4: ')
display(sym.simplify(g_b3a4))
```

```
print('\033[1m g_b4_a1: ')
display(sym.simplify(g_b4a1))
print('\033[1m g_b4_a2: ')
display(sym.simplify(g_b4a2))
print('\033[1m g_b4_a3: ')
display(sym.simplify(g_b4a3))
print('\033[1m g_b4_a4: ')
display(sym.simplify(g_b4a4))

#calculate KE and PE

V_wa = SE3_Inv(g_wa)*g_wa.diff(t)
V_wb1 = SE3_Inv(g_wb1)*g_wb1.diff(t)
V_wb2 = SE3_Inv(g_wb2)*g_wb2.diff(t)
V_wb3 = SE3_Inv(g_wb3)*g_wb3.diff(t)
V_wb4 = SE3_Inv(g_wb4)*g_wb4.diff(t)

Vb_wa = UnHat(V_wa)
Vb_wb1 = UnHat(V_wb1)
Vb_wb2 = UnHat(V_wb2)
Vb_wb3 = UnHat(V_wb3)
Vb_wb4 = UnHat(V_wb4)

Ma = CreateM(M,J)
Mb = CreateM(m,J)

KE =0.5*Vb_wa.transpose()*Ma*Vb_wa+0.5*Vb_wb1.transpose()*Mb*Vb_wb1+0.5*Vb_wb2.
 ↪transpose()*Mb*Vb_wb2+0.5*Vb_wb3.transpose()*Mb*Vb_wb3+0.5*Vb_wb4.
 ↪transpose()*Mb*Vb_wb4
PE = g_wa[1,3]*g*M+g_wb1[1,3]*g*m+g_wb2[1,3]*g*m+g_wb3[1,3]*g*m+g_wb4[1,3]*g*m

L = sym.simplify(KE[0]-PE)
print('\033[1mLagrangian: ')
display(L)
```

g_wa:

$$\begin{bmatrix} \cos(\theta_1(t)) & -\sin(\theta_1(t)) & 0 & \mathrm{x}_1(t) \\ \sin(\theta_1(t)) & \cos(\theta_1(t)) & 0 & \mathrm{y}_1(t) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1.0 \end{bmatrix}$$

g_wb:

$$\begin{bmatrix} \cos(\theta_2(t)) & -\sin(\theta_2(t)) & 0 & \mathrm{x}_2(t) \\ \sin(\theta_2(t)) & \cos(\theta_2(t)) & 0 & \mathrm{y}_2(t) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1.0 \end{bmatrix}$$

g_a_a1:

$$\begin{bmatrix}1 & 0 & 0 & 0.5\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

g_a_a2:

$$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & -0.5\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

g_a_a3:

$$\begin{bmatrix}1 & 0 & 0 & -0.5\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

g_a_a4:

$$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0.5\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

g_b_b1:

$$\begin{bmatrix}1 & 0 & 0 & 3.0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1.0\end{bmatrix}$$

g_b_b2:

$$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & -3.0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1.0\end{bmatrix}$$

g_b_b3:

$$\begin{bmatrix}1 & 0 & 0 & -3.0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1.0\end{bmatrix}$$

g_b_b4:

$$\begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 3.0\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1.0\end{bmatrix}$$

g_w_a1:

$$\begin{bmatrix}\cos\left(\theta_{1}(t)\right) & -\sin\left(\theta_{1}(t)\right) & 0 & \mathrm{x}_{1}\left(t\right) + 0.5\cos\left(\theta_{1}(t)\right)\\\sin\left(\theta_{1}(t)\right) & \cos\left(\theta_{1}(t)\right) & 0 & \mathrm{y}_{1}\left(t\right) + 0.5\sin\left(\theta_{1}(t)\right)\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1.0\end{bmatrix}$$

g_w_a2:

$$\begin{bmatrix}\cos\left(\theta_{1}(t)\right) & -\sin\left(\theta_{1}(t)\right) & 0 & \mathrm{x}_{1}\left(t\right) + 0.5\sin\left(\theta_{1}(t)\right)\\\sin\left(\theta_{1}(t)\right) & \cos\left(\theta_{1}(t)\right) & 0 & \mathrm{y}_{1}\left(t\right) - 0.5\cos\left(\theta_{1}(t)\right)\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1.0\end{bmatrix}$$

g_w_a3:

$$\begin{bmatrix}\cos\left(\theta_{1}(t)\right) & -\sin\left(\theta_{1}(t)\right) & 0 & \mathrm{x}_{1}\left(t\right) - 0.5\cos\left(\theta_{1}(t)\right)\\\sin\left(\theta_{1}(t)\right) & \cos\left(\theta_{1}(t)\right) & 0 & \mathrm{y}_{1}\left(t\right) - 0.5\sin\left(\theta_{1}(t)\right)\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1.0\end{bmatrix}$$

g_w_a4:

$$\begin{bmatrix}\cos\left(\theta_{1}(t)\right) & -\sin\left(\theta_{1}(t)\right) & 0 & \mathrm{x}_{1}\left(t\right) - 0.5\sin\left(\theta_{1}(t)\right)\\\sin\left(\theta_{1}(t)\right) & \cos\left(\theta_{1}(t)\right) & 0 & \mathrm{y}_{1}\left(t\right) + 0.5\cos\left(\theta_{1}(t)\right)\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1.0\end{bmatrix}$$

g_w_b1:

$$\begin{bmatrix}\cos\left(\theta_{2}(t)\right) & -\sin\left(\theta_{2}(t)\right) & 0 & 1.0\,\mathrm{x}_{2}\left(t\right) + 3.0\cos\left(\theta_{2}(t)\right)\\\sin\left(\theta_{2}(t)\right) & \cos\left(\theta_{2}(t)\right) & 0 & 1.0\,\mathrm{y}_{2}\left(t\right) + 3.0\sin\left(\theta_{2}(t)\right)\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1.0\end{bmatrix}$$

g_w_b22:

$$\begin{bmatrix}\cos\left(\theta_{2}(t)\right) & -\sin\left(\theta_{2}(t)\right) & 0 & 1.0\,\mathrm{x}_{2}\left(t\right) + 3.0\sin\left(\theta_{2}(t)\right)\\\sin\left(\theta_{2}(t)\right) & \cos\left(\theta_{2}(t)\right) & 0 & 1.0\,\mathrm{y}_{2}\left(t\right) - 3.0\cos\left(\theta_{2}(t)\right)\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1.0\end{bmatrix}$$

g_w_b3:

$$\begin{bmatrix}\cos\left(\theta_{2}(t)\right) & -\sin\left(\theta_{2}(t)\right) & 0 & 1.0\,\mathrm{x}_{2}\left(t\right) - 3.0\cos\left(\theta_{2}(t)\right)\\\sin\left(\theta_{2}(t)\right) & \cos\left(\theta_{2}(t)\right) & 0 & 1.0\,\mathrm{y}_{2}\left(t\right) - 3.0\sin\left(\theta_{2}(t)\right)\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1.0\end{bmatrix}$$

g_w_b4:

$$\left[\begin{matrix}\cos{\left(\theta_{2}(t) \right)} & - \sin{\left(\theta_{2}(t) \right)} & 0 & 1.0 \operatorname{x}_{2}{\left(t \right)} - 3.0 \sin{\left(\theta_{2}(t) \right)}\\\sin{\left(\theta_{2}(t) \right)} & \cos{\left(\theta_{2}(t) \right)} & 0 & 1.0 \operatorname{y}_{2}{\left(t \right)} + 3.0 \cos{\left(\theta_{2}(t) \right)}\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1.0\end{matrix}\right]$$

g_b1_a1:

$$\left[\begin{matrix}\cos{\left(\theta_{1}(t) - \theta_{2}(t) \right)} & - \sin{\left(\theta_{1}(t) - \theta_{2}(t) \right)} & 0 & 1.0 \operatorname{x}_{1}{\left(t \right)} \cos{\left(\theta_{2}(t) \right)} - 1.0 \operatorname{x}_{2}{\left(t \right)} \cos{\left(\theta_{2}(t) \right)} + 1.0 \operatorname{y}_{1}{\left(t \right)} \sin{\left(\theta_{2}(t) \right)} - 1.0\\\sin{\left(\theta_{1}(t) - \theta_{2}(t) \right)} & \cos{\left(\theta_{1}(t) - \theta_{2}(t) \right)} & 0 & - 1.0 \operatorname{x}_{1}{\left(t \right)} \sin{\left(\theta_{2}(t) \right)} + 1.0 \operatorname{x}_{2}{\left(t \right)} \sin{\left(\theta_{2}(t) \right)} + 1.0 \operatorname{y}_{1}{\left(t \right)} \cos{\left(\theta_{2}(t) \right)} -\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1.0\end{matrix}\right.$$

g_b1_a2:

$$\left[\begin{matrix}\cos{\left(\theta_{1}(t) - \theta_{2}(t) \right)} & - \sin{\left(\theta_{1}(t) - \theta_{2}(t) \right)} & 0 & 1.0 \operatorname{x}_{1}{\left(t \right)} \cos{\left(\theta_{2}(t) \right)} - 1.0 \operatorname{x}_{2}{\left(t \right)} \cos{\left(\theta_{2}(t) \right)} + 1.0 \operatorname{y}_{1}{\left(t \right)} \sin{\left(\theta_{2}(t) \right)} - 1.0\\\sin{\left(\theta_{1}(t) - \theta_{2}(t) \right)} & \cos{\left(\theta_{1}(t) - \theta_{2}(t) \right)} & 0 & - 1.0 \operatorname{x}_{1}{\left(t \right)} \sin{\left(\theta_{2}(t) \right)} + 1.0 \operatorname{x}_{2}{\left(t \right)} \sin{\left(\theta_{2}(t) \right)} + 1.0 \operatorname{y}_{1}{\left(t \right)} \cos{\left(\theta_{2}(t) \right)} -\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1.0\end{matrix}\right.$$

g_b1_a3:

$$\left[\begin{matrix}\cos{\left(\theta_{1}(t) - \theta_{2}(t) \right)} & - \sin{\left(\theta_{1}(t) - \theta_{2}(t) \right)} & 0 & 1.0 \operatorname{x}_{1}{\left(t \right)} \cos{\left(\theta_{2}(t) \right)} - 1.0 \operatorname{x}_{2}{\left(t \right)} \cos{\left(\theta_{2}(t) \right)} + 1.0 \operatorname{y}_{1}{\left(t \right)} \sin{\left(\theta_{2}(t) \right)} - 1.0\\\sin{\left(\theta_{1}(t) - \theta_{2}(t) \right)} & \cos{\left(\theta_{1}(t) - \theta_{2}(t) \right)} & 0 & - 1.0 \operatorname{x}_{1}{\left(t \right)} \sin{\left(\theta_{2}(t) \right)} + 1.0 \operatorname{x}_{2}{\left(t \right)} \sin{\left(\theta_{2}(t) \right)} + 1.0 \operatorname{y}_{1}{\left(t \right)} \cos{\left(\theta_{2}(t) \right)} -\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1.0\end{matrix}\right.$$

g_b1_a4:

$$\left[\begin{matrix}\cos{\left(\theta_{1}(t) - \theta_{2}(t) \right)} & - \sin{\left(\theta_{1}(t) - \theta_{2}(t) \right)} & 0 & 1.0 \operatorname{x}_{1}{\left(t \right)} \cos{\left(\theta_{2}(t) \right)} - 1.0 \operatorname{x}_{2}{\left(t \right)} \cos{\left(\theta_{2}(t) \right)} + 1.0 \operatorname{y}_{1}{\left(t \right)} \sin{\left(\theta_{2}(t) \right)} - 1.0\\\sin{\left(\theta_{1}(t) - \theta_{2}(t) \right)} & \cos{\left(\theta_{1}(t) - \theta_{2}(t) \right)} & 0 & - 1.0 \operatorname{x}_{1}{\left(t \right)} \sin{\left(\theta_{2}(t) \right)} + 1.0 \operatorname{x}_{2}{\left(t \right)} \sin{\left(\theta_{2}(t) \right)} + 1.0 \operatorname{y}_{1}{\left(t \right)} \cos{\left(\theta_{2}(t) \right)} -\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1.0\end{matrix}\right.$$

g_b2_a1:

$$\left[\begin{matrix}\cos{\left(\theta_{1}(t) - \theta_{2}(t) \right)} & - \sin{\left(\theta_{1}(t) - \theta_{2}(t) \right)} & 0 & 1.0 \operatorname{x}_{1}{\left(t \right)} \cos{\left(\theta_{2}(t) \right)} - 1.0 \operatorname{x}_{2}{\left(t \right)} \cos{\left(\theta_{2}(t) \right)} + 1.0 \operatorname{y}_{1}{\left(t \right)} \sin{\left(\theta_{2}(t) \right)} -\\\sin{\left(\theta_{1}(t) - \theta_{2}(t) \right)} & \cos{\left(\theta_{1}(t) - \theta_{2}(t) \right)} & 0 & - 1.0 \operatorname{x}_{1}{\left(t \right)} \sin{\left(\theta_{2}(t) \right)} + 1.0 \operatorname{x}_{2}{\left(t \right)} \sin{\left(\theta_{2}(t) \right)} + 1.0 \operatorname{y}_{1}{\left(t \right)} \cos{\left(\theta_{2}(t) \right)} - 1.\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1.0\end{matrix}\right.$$

g_b2_a2:

$$\left[\begin{matrix}\cos{\left(\theta_{1}(t) - \theta_{2}(t) \right)} & - \sin{\left(\theta_{1}(t) - \theta_{2}(t) \right)} & 0 & 1.0 \operatorname{x}_{1}{\left(t \right)} \cos{\left(\theta_{2}(t) \right)} - 1.0 \operatorname{x}_{2}{\left(t \right)} \cos{\left(\theta_{2}(t) \right)} + 1.0 \operatorname{y}_{1}{\left(t \right)} \sin{\left(\theta_{2}(t) \right)} -\\\sin{\left(\theta_{1}(t) - \theta_{2}(t) \right)} & \cos{\left(\theta_{1}(t) - \theta_{2}(t) \right)} & 0 & - 1.0 \operatorname{x}_{1}{\left(t \right)} \sin{\left(\theta_{2}(t) \right)} + 1.0 \operatorname{x}_{2}{\left(t \right)} \sin{\left(\theta_{2}(t) \right)} + 1.0 \operatorname{y}_{1}{\left(t \right)} \cos{\left(\theta_{2}(t) \right)} - 1.\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1.0\end{matrix}\right.$$

g_b2_a3:

$$\left[\begin{matrix}\cos\left(\theta_{1}(t)-\theta_{2}(t)\right) & -\sin\left(\theta_{1}(t)-\theta_{2}(t)\right) & 0 & 1.0\,\mathrm{x}_{1}\left(t\right)\cos\left(\theta_{2}(t)\right)-1.0\,\mathrm{x}_{2}\left(t\right)\cos\left(\theta_{2}(t)\right)+1.0\,\mathrm{y}_{1}\left(t\right)\sin\left(\theta_{2}(t)\right)-\\ \sin\left(\theta_{1}(t)-\theta_{2}(t)\right) & \cos\left(\theta_{1}(t)-\theta_{2}(t)\right) & 0 & -1.0\,\mathrm{x}_{1}\left(t\right)\sin\left(\theta_{2}(t)\right)+1.0\,\mathrm{x}_{2}\left(t\right)\sin\left(\theta_{2}(t)\right)+1.0\,\mathrm{y}_{1}\left(t\right)\cos\left(\theta_{2}(t)\right)-1.\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1.0\end{matrix}\right.$$

g_b2_a4:

$$\left[\begin{matrix}\cos\left(\theta_{1}(t)-\theta_{2}(t)\right) & -\sin\left(\theta_{1}(t)-\theta_{2}(t)\right) & 0 & 1.0\,\mathrm{x}_{1}\left(t\right)\cos\left(\theta_{2}(t)\right)-1.0\,\mathrm{x}_{2}\left(t\right)\cos\left(\theta_{2}(t)\right)+1.0\,\mathrm{y}_{1}\left(t\right)\sin\left(\theta_{2}(t)\right)-\\ \sin\left(\theta_{1}(t)-\theta_{2}(t)\right) & \cos\left(\theta_{1}(t)-\theta_{2}(t)\right) & 0 & -1.0\,\mathrm{x}_{1}\left(t\right)\sin\left(\theta_{2}(t)\right)+1.0\,\mathrm{x}_{2}\left(t\right)\sin\left(\theta_{2}(t)\right)+1.0\,\mathrm{y}_{1}\left(t\right)\cos\left(\theta_{2}(t)\right)-1.\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1.0\end{matrix}\right.$$

g_b3_a1:

$$\left[\begin{matrix}\cos\left(\theta_{1}(t)-\theta_{2}(t)\right) & -\sin\left(\theta_{1}(t)-\theta_{2}(t)\right) & 0 & 1.0\,\mathrm{x}_{1}\left(t\right)\cos\left(\theta_{2}(t)\right)-1.0\,\mathrm{x}_{2}\left(t\right)\cos\left(\theta_{2}(t)\right)+1.0\,\mathrm{y}_{1}\left(t\right)\sin\left(\theta_{2}(t)\right)-1.0\\ \sin\left(\theta_{1}(t)-\theta_{2}(t)\right) & \cos\left(\theta_{1}(t)-\theta_{2}(t)\right) & 0 & -1.0\,\mathrm{x}_{1}\left(t\right)\sin\left(\theta_{2}(t)\right)+1.0\,\mathrm{x}_{2}\left(t\right)\sin\left(\theta_{2}(t)\right)+1.0\,\mathrm{y}_{1}\left(t\right)\cos\left(\theta_{2}(t)\right)-\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1.0\end{matrix}\right.$$

g_b3_a2:

$$\left[\begin{matrix}\cos\left(\theta_{1}(t)-\theta_{2}(t)\right) & -\sin\left(\theta_{1}(t)-\theta_{2}(t)\right) & 0 & 1.0\,\mathrm{x}_{1}\left(t\right)\cos\left(\theta_{2}(t)\right)-1.0\,\mathrm{x}_{2}\left(t\right)\cos\left(\theta_{2}(t)\right)+1.0\,\mathrm{y}_{1}\left(t\right)\sin\left(\theta_{2}(t)\right)-1.0\\ \sin\left(\theta_{1}(t)-\theta_{2}(t)\right) & \cos\left(\theta_{1}(t)-\theta_{2}(t)\right) & 0 & -1.0\,\mathrm{x}_{1}\left(t\right)\sin\left(\theta_{2}(t)\right)+1.0\,\mathrm{x}_{2}\left(t\right)\sin\left(\theta_{2}(t)\right)+1.0\,\mathrm{y}_{1}\left(t\right)\cos\left(\theta_{2}(t)\right)-\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1.0\end{matrix}\right.$$

g_b3_a3:

$$\left[\begin{matrix}\cos\left(\theta_{1}(t)-\theta_{2}(t)\right) & -\sin\left(\theta_{1}(t)-\theta_{2}(t)\right) & 0 & 1.0\,\mathrm{x}_{1}\left(t\right)\cos\left(\theta_{2}(t)\right)-1.0\,\mathrm{x}_{2}\left(t\right)\cos\left(\theta_{2}(t)\right)+1.0\,\mathrm{y}_{1}\left(t\right)\sin\left(\theta_{2}(t)\right)-1.0\\ \sin\left(\theta_{1}(t)-\theta_{2}(t)\right) & \cos\left(\theta_{1}(t)-\theta_{2}(t)\right) & 0 & -1.0\,\mathrm{x}_{1}\left(t\right)\sin\left(\theta_{2}(t)\right)+1.0\,\mathrm{x}_{2}\left(t\right)\sin\left(\theta_{2}(t)\right)+1.0\,\mathrm{y}_{1}\left(t\right)\cos\left(\theta_{2}(t)\right)-\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1.0\end{matrix}\right.$$

g_b3_a4:

$$\left[\begin{matrix}\cos\left(\theta_{1}(t)-\theta_{2}(t)\right) & -\sin\left(\theta_{1}(t)-\theta_{2}(t)\right) & 0 & 1.0\,\mathrm{x}_{1}\left(t\right)\cos\left(\theta_{2}(t)\right)-1.0\,\mathrm{x}_{2}\left(t\right)\cos\left(\theta_{2}(t)\right)+1.0\,\mathrm{y}_{1}\left(t\right)\sin\left(\theta_{2}(t)\right)-1.0\\ \sin\left(\theta_{1}(t)-\theta_{2}(t)\right) & \cos\left(\theta_{1}(t)-\theta_{2}(t)\right) & 0 & -1.0\,\mathrm{x}_{1}\left(t\right)\sin\left(\theta_{2}(t)\right)+1.0\,\mathrm{x}_{2}\left(t\right)\sin\left(\theta_{2}(t)\right)+1.0\,\mathrm{y}_{1}\left(t\right)\cos\left(\theta_{2}(t)\right)-\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1.0\end{matrix}\right.$$

g_b4_a1:

$$\left[\begin{matrix}\cos\left(\theta_{1}(t)-\theta_{2}(t)\right) & -\sin\left(\theta_{1}(t)-\theta_{2}(t)\right) & 0 & 1.0\,\mathrm{x}_{1}\left(t\right)\cos\left(\theta_{2}(t)\right)-1.0\,\mathrm{x}_{2}\left(t\right)\cos\left(\theta_{2}(t)\right)+1.0\,\mathrm{y}_{1}\left(t\right)\sin\left(\theta_{2}(t)\right)-\\ \sin\left(\theta_{1}(t)-\theta_{2}(t)\right) & \cos\left(\theta_{1}(t)-\theta_{2}(t)\right) & 0 & -1.0\,\mathrm{x}_{1}\left(t\right)\sin\left(\theta_{2}(t)\right)+1.0\,\mathrm{x}_{2}\left(t\right)\sin\left(\theta_{2}(t)\right)+1.0\,\mathrm{y}_{1}\left(t\right)\cos\left(\theta_{2}(t)\right)-1.\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1.0\end{matrix}\right.$$

g_b4_a2:

$$\left[\begin{matrix}\cos\left(\theta_{1}(t)-\theta_{2}(t)\right) & -\sin\left(\theta_{1}(t)-\theta_{2}(t)\right) & 0 & 1.0\,\mathrm{x}_{1}\left(t\right)\cos\left(\theta_{2}(t)\right)-1.0\,\mathrm{x}_{2}\left(t\right)\cos\left(\theta_{2}(t)\right)+1.0\,\mathrm{y}_{1}\left(t\right)\sin\left(\theta_{2}(t)\right)-\\ \sin\left(\theta_{1}(t)-\theta_{2}(t)\right) & \cos\left(\theta_{1}(t)-\theta_{2}(t)\right) & 0 & -1.0\,\mathrm{x}_{1}\left(t\right)\sin\left(\theta_{2}(t)\right)+1.0\,\mathrm{x}_{2}\left(t\right)\sin\left(\theta_{2}(t)\right)+1.0\,\mathrm{y}_{1}\left(t\right)\cos\left(\theta_{2}(t)\right)-1.\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1.0\end{matrix}\right.$$

g_b4_a3:

$$\left[\begin{array}{cccc} \cos\left(\theta_1(t)-\theta_2(t)\right) & -\sin\left(\theta_1(t)-\theta_2(t)\right) & 0 & 1.0\,\mathrm{x}_1(t)\cos\left(\theta_2(t)\right)-1.0\,\mathrm{x}_2(t)\cos\left(\theta_2(t)\right)+1.0\,\mathrm{y}_1(t)\sin\left(\theta_2(t)\right)- \\ \sin\left(\theta_1(t)-\theta_2(t)\right) & \cos\left(\theta_1(t)-\theta_2(t)\right) & 0 & -1.0\,\mathrm{x}_1(t)\sin\left(\theta_2(t)\right)+1.0\,\mathrm{x}_2(t)\sin\left(\theta_2(t)\right)+1.0\,\mathrm{y}_1(t)\cos\left(\theta_2(t)\right)-1. \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1.0 \end{array}\right.$$

g_b4_a4:

$$\left[\begin{array}{cccc} \cos\left(\theta_1(t)-\theta_2(t)\right) & -\sin\left(\theta_1(t)-\theta_2(t)\right) & 0 & 1.0\,\mathrm{x}_1(t)\cos\left(\theta_2(t)\right)-1.0\,\mathrm{x}_2(t)\cos\left(\theta_2(t)\right)+1.0\,\mathrm{y}_1(t)\sin\left(\theta_2(t)\right)- \\ \sin\left(\theta_1(t)-\theta_2(t)\right) & \cos\left(\theta_1(t)-\theta_2(t)\right) & 0 & -1.0\,\mathrm{x}_1(t)\sin\left(\theta_2(t)\right)+1.0\,\mathrm{x}_2(t)\sin\left(\theta_2(t)\right)+1.0\,\mathrm{y}_1(t)\cos\left(\theta_2(t)\right)-1. \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1.0 \end{array}\right.$$

Lagrangian:

$$-4.9\,\mathrm{y}_1(t)-392.0\,\mathrm{y}_2(t)+0.5\left(\frac{d}{dt}\theta_1(t)\right)^2+182.0\left(\frac{d}{dt}\theta_2(t)\right)^2+0.25\left(\frac{d}{dt}\mathrm{x}_1(t)\right)^2+20.0\left(\frac{d}{dt}\mathrm{x}_2(t)\right)^2+0.25\left(\frac{d}{dt}\mathrm{y}_1(t)\right)^2+20.0\left(\frac{d}{dt}\mathrm{y}_2(t)\right)^2$$

[5]:

```
# External Force(touque)
F2 = -k1*(theta1-(np.pi/15+np.pi/3*sym.sin(t/2)**2))
F2y = g*m*4+g*M*0.9
#EL

F2 = sym.Matrix([0,0,0,0,F2y,500*sym.cos(t*2)]).T
#F2 = sym.Matrix([0,0,0,0,F2y,t]).T
print('\033[1mForce: ')
display(F2)
```

Force:

$$\left[\begin{matrix}0 & 0 & 0 & 0 & 396.41 & 500\cos\left(2t\right)\end{matrix}\right]$$

[6]:

```
q = sym.Matrix([x1,y1,theta1,x2,y2,theta2])
qdot = q.diff(t)
qddot = qdot.diff(t)

L_mat = sym.Matrix([L])
dldq = L_mat.jacobian(q)
dldqd = L_mat.jacobian(qdot)

EL_matrix = dldqd.diff(t)-dldq
EL_matrix = EL_matrix
display(EL_matrix)
EL_Right_side = F2

eqn_EL = sym.Eq(EL_matrix,EL_Right_side)
```

```
q_EL = sym.Matrix([x1ddot,y1ddot,theta1ddot,x2ddot,y2ddot,theta2ddot])
soln_EL = sym.solve([eqn_EL],q_EL ,dict=True)

for sol in soln_EL:
    print('\n\033[1mSolution: ')
    for v in q_EL:
        display(v)
        display(sym.Eq(v, sol[v]))
```

$$\left[0.5\frac{d^2}{dt^2}\mathrm{x}_1(t) \quad 0.5\frac{d^2}{dt^2}\mathrm{y}_1(t)+4.9 \quad 1.0\frac{d^2}{dt^2}\theta_1(t) \quad 40.0\frac{d^2}{dt^2}\mathrm{x}_2(t) \quad 40.0\frac{d^2}{dt^2}\mathrm{y}_2(t)+392.0 \quad 364.0\frac{d^2}{dt^2}\theta_2(t)\right]$$

Solution:

$$\frac{d^2}{dt^2}\mathrm{x}_1(t)$$

$$\frac{d^2}{dt^2}\mathrm{x}_1(t) = 0.0$$

$$\frac{d^2}{dt^2}\mathrm{y}_1(t)$$

$$\frac{d^2}{dt^2}\mathrm{y}_1(t) = -9.8$$

$$\frac{d^2}{dt^2}\theta_1(t)$$

$$\frac{d^2}{dt^2}\theta_1(t) = 0.0$$

$$\frac{d^2}{dt^2}\mathrm{x}_2(t)$$

$$\frac{d^2}{dt^2}\mathrm{x}_2(t) = 0.0$$

$$\frac{d^2}{dt^2}\mathrm{y}_2(t)$$

$$\frac{d^2}{dt^2}\mathrm{y}_2(t) = 0.11025$$

$$\frac{d^2}{dt^2}\theta_2(t)$$

$$\frac{d^2}{dt^2}\theta_2(t) = 1.37362637362637\cos(2.0t)$$

[7]:

```
# Setup dummy variables
x1M, y1M, theta1M, x2M, y2M, theta2M,x1dotM,y1dotM,theta1dotM,␣
 ↪x2dotM,y2dotM,theta2dotM = sym.symbols(r'x1^-,y1^-, theta_1^-, x2^-,y2^-,␣
 ↪theta_2^-,\dot{x1}^-, \dot{y1}^-, \dot{\theta_1}^-,\dot{x2}^-, \dot{y2}^-,␣
 ↪\dot{\theta_2}^-')
```

```
x1dotP,y1dotP,theta1dotP, x2dotP,y2dotP,theta2dotP = sym.symbols(r'\dot{x1}^+, \dot{y1}^+, \dot{\theta_1}^+,\dot{x2}^+, \dot{y2}^+, \dot{\theta_2}^+')
lamb = sym.symbols(r'lambda')
subs_minus1 = {q[0]:x1M,
               q[1]:y1M,
               q[2]:theta1M,
               q[3]:x2M,
               q[4]:y2M,
               q[5]:theta2M,
               qdot[0]:x1dotM,
               qdot[1]:y1dotM,
               qdot[2]:theta1dotM,
               qdot[3]:x2dotM,
               qdot[4]:y2dotM,
               qdot[5]:theta2dotM}

subs_plus1 = {q[0]:x1M,
               q[1]:y1M,
               q[2]:theta1M,
               q[3]:x2M,
               q[4]:y2M,
               q[5]:theta2M,
               qdot[0]:x1dotP,
               qdot[1]:y1dotP,
               qdot[2]:theta1dotP,
               qdot[3]:x2dotP,
               qdot[4]:y2dotP,
               qdot[5]:theta2dotP}
#display(subs_minus1)
display(g_b1a1[0,3])

#phi with width

Phi_ini = sym.Matrix([g_b1a1[0,3]+0.1,g_b1a2[0,3]+0.1,g_b1a3[0,3]+0.1,g_b1a4[0,3]+0.1,
                      g_b2a1[1,3]-0.1,g_b2a2[1,3]-0.1,g_b2a3[1,3]-0.1,g_b2a4[1,3]-0.1,
                      g_b3a1[0,3]-0.1,g_b3a2[0,3]-0.1,g_b3a3[0,3]-0.1,g_b3a4[0,3]-0.1,
                      g_b4a1[1,3]+0.1,g_b4a2[1,3]+0.1,g_b4a3[1,3]+0.1,g_b4a4[1,3]+0.1])
# Phi_ini = sym.Matrix([g_b1a1[0,3],g_b1a2[0,3],g_b1a3[0,3],g_b1a4[0,3],
#                       g_b2a1[1,3],g_b2a2[1,3],g_b2a3[1,3],g_b2a4[1,3],
#                       g_b3a1[0,3],g_b3a2[0,3],g_b3a3[0,3],g_b3a4[0,3],
#                       g_b4a1[1,3],g_b4a2[1,3],g_b4a3[1,3],g_b4a4[1,3]])
#display(Phi_ini)
dPhidq = sym.Matrix(Phi_ini).jacobian(q)
```

```
phi = Phi_ini.subs(subs_minus1)
display(phi)
```

$$(\mathrm{x}_1(t) + 0.5\cos(\theta_1(t)))\cos(\theta_2(t)) - 1.0\,(1.0\,\mathrm{x}_2(t) + 3.0\cos(\theta_2(t)))\cos(\theta_2(t)) + (\mathrm{y}_1(t) + 0.5\sin(\theta_1(t)))\sin(\theta_2(t)) - 1.0\,(1.0\,\mathrm{y}_2(t) + 3.0\sin(\theta_2(t)))\sin(\theta_2(t))$$

$$\left[\begin{matrix}
(x_1^- + 0.5\cos(\theta_1^-))\cos(\theta_2^-) - 1.0\,(1.0x_2^- + 3.0\cos(\theta_2^-))\cos(\theta_2^-) + (y_1^- + 0.5\sin(\theta_1^-))\sin(\theta_2^-) - 1.0\,(1.0y_2^- + 3.0\sin \\
(x_1^- + 0.5\sin(\theta_1^-))\cos(\theta_2^-) - 1.0\,(1.0x_2^- + 3.0\cos(\theta_2^-))\cos(\theta_2^-) + (y_1^- - 0.5\cos(\theta_1^-))\sin(\theta_2^-) - 1.0\,(1.0y_2^- + 3.0\sin \\
(x_1^- - 0.5\cos(\theta_1^-))\cos(\theta_2^-) - 1.0\,(1.0x_2^- + 3.0\cos(\theta_2^-))\cos(\theta_2^-) + (y_1^- - 0.5\sin(\theta_1^-))\sin(\theta_2^-) - 1.0\,(1.0y_2^- + 3.0\sin \\
(x_1^- - 0.5\sin(\theta_1^-))\cos(\theta_2^-) - 1.0\,(1.0x_2^- + 3.0\cos(\theta_2^-))\cos(\theta_2^-) + (y_1^- + 0.5\cos(\theta_1^-))\sin(\theta_2^-) - 1.0\,(1.0y_2^- + 3.0\sin \\
-(x_1^- + 0.5\cos(\theta_1^-))\sin(\theta_2^-) + 1.0\,(1.0x_2^- + 3.0\sin(\theta_2^-))\sin(\theta_2^-) + (y_1^- + 0.5\sin(\theta_1^-))\cos(\theta_2^-) - 1.0\,(1.0y_2^- - 3.0\cos \\
-(x_1^- + 0.5\sin(\theta_1^-))\sin(\theta_2^-) + 1.0\,(1.0x_2^- + 3.0\sin(\theta_2^-))\sin(\theta_2^-) + (y_1^- - 0.5\cos(\theta_1^-))\cos(\theta_2^-) - 1.0\,(1.0y_2^- - 3.0\cos \\
-(x_1^- - 0.5\cos(\theta_1^-))\sin(\theta_2^-) + 1.0\,(1.0x_2^- + 3.0\sin(\theta_2^-))\sin(\theta_2^-) + (y_1^- - 0.5\sin(\theta_1^-))\cos(\theta_2^-) - 1.0\,(1.0y_2^- - 3.0\cos \\
-(x_1^- - 0.5\sin(\theta_1^-))\sin(\theta_2^-) + 1.0\,(1.0x_2^- + 3.0\sin(\theta_2^-))\sin(\theta_2^-) + (y_1^- + 0.5\cos(\theta_1^-))\cos(\theta_2^-) - 1.0\,(1.0y_2^- - 3.0\cos \\
(x_1^- + 0.5\cos(\theta_1^-))\cos(\theta_2^-) - 1.0\,(1.0x_2^- - 3.0\cos(\theta_2^-))\cos(\theta_2^-) + (y_1^- + 0.5\sin(\theta_1^-))\sin(\theta_2^-) - 1.0\,(1.0y_2^- - 3.0\sin \\
(x_1^- + 0.5\sin(\theta_1^-))\cos(\theta_2^-) - 1.0\,(1.0x_2^- - 3.0\cos(\theta_2^-))\cos(\theta_2^-) + (y_1^- - 0.5\cos(\theta_1^-))\sin(\theta_2^-) - 1.0\,(1.0y_2^- - 3.0\sin \\
(x_1^- - 0.5\cos(\theta_1^-))\cos(\theta_2^-) - 1.0\,(1.0x_2^- - 3.0\cos(\theta_2^-))\cos(\theta_2^-) + (y_1^- - 0.5\sin(\theta_1^-))\sin(\theta_2^-) - 1.0\,(1.0y_2^- - 3.0\sin \\
(x_1^- - 0.5\sin(\theta_1^-))\cos(\theta_2^-) - 1.0\,(1.0x_2^- - 3.0\cos(\theta_2^-))\cos(\theta_2^-) + (y_1^- + 0.5\cos(\theta_1^-))\sin(\theta_2^-) - 1.0\,(1.0y_2^- - 3.0\sin \\
-(x_1^- + 0.5\cos(\theta_1^-))\sin(\theta_2^-) + 1.0\,(1.0x_2^- - 3.0\sin(\theta_2^-))\sin(\theta_2^-) + (y_1^- + 0.5\sin(\theta_1^-))\cos(\theta_2^-) - 1.0\,(1.0y_2^- + 3.0\cos \\
-(x_1^- + 0.5\sin(\theta_1^-))\sin(\theta_2^-) + 1.0\,(1.0x_2^- - 3.0\sin(\theta_2^-))\sin(\theta_2^-) + (y_1^- - 0.5\cos(\theta_1^-))\cos(\theta_2^-) - 1.0\,(1.0y_2^- + 3.0\cos \\
-(x_1^- - 0.5\cos(\theta_1^-))\sin(\theta_2^-) + 1.0\,(1.0x_2^- - 3.0\sin(\theta_2^-))\sin(\theta_2^-) + (y_1^- - 0.5\sin(\theta_1^-))\cos(\theta_2^-) - 1.0\,(1.0y_2^- + 3.0\cos \\
-(x_1^- - 0.5\sin(\theta_1^-))\sin(\theta_2^-) + 1.0\,(1.0x_2^- - 3.0\sin(\theta_2^-))\sin(\theta_2^-) + (y_1^- + 0.5\cos(\theta_1^-))\cos(\theta_2^-) - 1.0\,(1.0y_2^- + 3.0\cos
\end{matrix}\right.$$

[8]:

```
# Impact Hamitltonian
H = (dldqd*qdot)[0] - L
display(H)
#dPhidq = sym.Matrix(phi).jacobian(q)
display(dPhidq)
#Before
dLdqdot_Minus = dldqd.subs(subs_minus1)
H_Minus = sym.trigsimp(H.subs(subs_minus1))

dPhidq_Minus = dPhidq.subs(subs_minus1)
phi_min = lamb*dPhidq_Minus

display(dLdqdot_Minus)

#After
dLdqdot_Plus = dldqd.subs(subs_plus1)
H_Plus = sym.trigsimp(H.subs(subs_plus1))
```

$$4.9\,\mathrm{y}_1(t) + 392.0\,\mathrm{y}_2(t) + 0.5\left(\frac{d}{dt}\theta_1(t)\right)^2 + 182.0\left(\frac{d}{dt}\theta_2(t)\right)^2 + 0.25\left(\frac{d}{dt}\mathrm{x}_1(t)\right)^2 + 20.0\left(\frac{d}{dt}\mathrm{x}_2(t)\right)^2 + 0.25\left(\frac{d}{dt}\mathrm{y}_1(t)\right)^2 + 20.0\left(\frac{d}{dt}\mathrm{y}_2(t)\right)^2$$

$$\left[\begin{matrix}\cos\left(\theta_{2}(t)\right) & \sin\left(\theta_{2}(t)\right) & -0.5\sin\left(\theta_{1}(t)\right)\cos\left(\theta_{2}(t)\right) + 0.5\sin\left(\theta_{2}(t)\right)\cos\left(\theta_{1}(t)\right) & -1.0\cos\left(\theta_{2}(t)\right) & -1.0\sin\left(\theta_{2}(t)\right)\\
\cos\left(\theta_{2}(t)\right) & \sin\left(\theta_{2}(t)\right) & 0.5\sin\left(\theta_{1}(t)\right)\sin\left(\theta_{2}(t)\right) + 0.5\cos\left(\theta_{1}(t)\right)\cos\left(\theta_{2}(t)\right) & -1.0\cos\left(\theta_{2}(t)\right) & -1.0\sin\left(\theta_{2}(t)\right)\\
\cos\left(\theta_{2}(t)\right) & \sin\left(\theta_{2}(t)\right) & 0.5\sin\left(\theta_{1}(t)\right)\cos\left(\theta_{2}(t)\right) - 0.5\sin\left(\theta_{2}(t)\right)\cos\left(\theta_{1}(t)\right) & -1.0\cos\left(\theta_{2}(t)\right) & -1.0\sin\left(\theta_{2}(t)\right)\\
\cos\left(\theta_{2}(t)\right) & \sin\left(\theta_{2}(t)\right) & -0.5\sin\left(\theta_{1}(t)\right)\sin\left(\theta_{2}(t)\right) - 0.5\cos\left(\theta_{1}(t)\right)\cos\left(\theta_{2}(t)\right) & -1.0\cos\left(\theta_{2}(t)\right) & -1.0\sin\left(\theta_{2}(t)\right)\\
-\sin\left(\theta_{2}(t)\right) & \cos\left(\theta_{2}(t)\right) & 0.5\sin\left(\theta_{1}(t)\right)\sin\left(\theta_{2}(t)\right) + 0.5\cos\left(\theta_{1}(t)\right)\cos\left(\theta_{2}(t)\right) & 1.0\sin\left(\theta_{2}(t)\right) & -1.0\cos\left(\theta_{2}(t)\right)\\
-\sin\left(\theta_{2}(t)\right) & \cos\left(\theta_{2}(t)\right) & 0.5\sin\left(\theta_{1}(t)\right)\cos\left(\theta_{2}(t)\right) - 0.5\sin\left(\theta_{2}(t)\right)\cos\left(\theta_{1}(t)\right) & 1.0\sin\left(\theta_{2}(t)\right) & -1.0\cos\left(\theta_{2}(t)\right)\\
-\sin\left(\theta_{2}(t)\right) & \cos\left(\theta_{2}(t)\right) & -0.5\sin\left(\theta_{1}(t)\right)\sin\left(\theta_{2}(t)\right) - 0.5\cos\left(\theta_{1}(t)\right)\cos\left(\theta_{2}(t)\right) & 1.0\sin\left(\theta_{2}(t)\right) & -1.0\cos\left(\theta_{2}(t)\right)\\
-\sin\left(\theta_{2}(t)\right) & \cos\left(\theta_{2}(t)\right) & -0.5\sin\left(\theta_{1}(t)\right)\cos\left(\theta_{2}(t)\right) + 0.5\sin\left(\theta_{2}(t)\right)\cos\left(\theta_{1}(t)\right) & 1.0\sin\left(\theta_{2}(t)\right) & -1.0\cos\left(\theta_{2}(t)\right)\\
\cos\left(\theta_{2}(t)\right) & \sin\left(\theta_{2}(t)\right) & -0.5\sin\left(\theta_{1}(t)\right)\cos\left(\theta_{2}(t)\right) + 0.5\sin\left(\theta_{2}(t)\right)\cos\left(\theta_{1}(t)\right) & -1.0\cos\left(\theta_{2}(t)\right) & -1.0\sin\left(\theta_{2}(t)\right)\\
\cos\left(\theta_{2}(t)\right) & \sin\left(\theta_{2}(t)\right) & 0.5\sin\left(\theta_{1}(t)\right)\sin\left(\theta_{2}(t)\right) + 0.5\cos\left(\theta_{1}(t)\right)\cos\left(\theta_{2}(t)\right) & -1.0\cos\left(\theta_{2}(t)\right) & -1.0\sin\left(\theta_{2}(t)\right)\\
\cos\left(\theta_{2}(t)\right) & \sin\left(\theta_{2}(t)\right) & 0.5\sin\left(\theta_{1}(t)\right)\cos\left(\theta_{2}(t)\right) - 0.5\sin\left(\theta_{2}(t)\right)\cos\left(\theta_{1}(t)\right) & -1.0\cos\left(\theta_{2}(t)\right) & -1.0\sin\left(\theta_{2}(t)\right)\\
\cos\left(\theta_{2}(t)\right) & \sin\left(\theta_{2}(t)\right) & -0.5\sin\left(\theta_{1}(t)\right)\sin\left(\theta_{2}(t)\right) - 0.5\cos\left(\theta_{1}(t)\right)\cos\left(\theta_{2}(t)\right) & -1.0\cos\left(\theta_{2}(t)\right) & -1.0\sin\left(\theta_{2}(t)\right)\\
-\sin\left(\theta_{2}(t)\right) & \cos\left(\theta_{2}(t)\right) & 0.5\sin\left(\theta_{1}(t)\right)\sin\left(\theta_{2}(t)\right) + 0.5\cos\left(\theta_{1}(t)\right)\cos\left(\theta_{2}(t)\right) & 1.0\sin\left(\theta_{2}(t)\right) & -1.0\cos\left(\theta_{2}(t)\right)\\
-\sin\left(\theta_{2}(t)\right) & \cos\left(\theta_{2}(t)\right) & 0.5\sin\left(\theta_{1}(t)\right)\cos\left(\theta_{2}(t)\right) - 0.5\sin\left(\theta_{2}(t)\right)\cos\left(\theta_{1}(t)\right) & 1.0\sin\left(\theta_{2}(t)\right) & -1.0\cos\left(\theta_{2}(t)\right)\\
-\sin\left(\theta_{2}(t)\right) & \cos\left(\theta_{2}(t)\right) & -0.5\sin\left(\theta_{1}(t)\right)\sin\left(\theta_{2}(t)\right) - 0.5\cos\left(\theta_{1}(t)\right)\cos\left(\theta_{2}(t)\right) & 1.0\sin\left(\theta_{2}(t)\right) & -1.0\cos\left(\theta_{2}(t)\right)\\
-\sin\left(\theta_{2}(t)\right) & \cos\left(\theta_{2}(t)\right) & -0.5\sin\left(\theta_{1}(t)\right)\cos\left(\theta_{2}(t)\right) + 0.5\sin\left(\theta_{2}(t)\right)\cos\left(\theta_{1}(t)\right) & 1.0\sin\left(\theta_{2}(t)\right) & -1.0\cos\left(\theta_{2}(t)\right)
\end{matrix}\right.$$

$$\left[\begin{matrix}0.5\dot{x}1^{-} & 0.5\dot{y}1^{-} & 1.0\dot{\theta}_{1}^{-} & 40.0\dot{x}2^{-} & 40.0\dot{y}2^{-} & 364.0\dot{\theta}_{2}^{-}\end{matrix}\right]$$

[9]:

```python
# Impact function
global impact_eqns
impact_eqns = []

impact_eqns_lhs = sym.Matrix([dLdqdot_Plus[0]-dLdqdot_Minus[0],
                              dLdqdot_Plus[1]-dLdqdot_Minus[1],
                              dLdqdot_Plus[2]-dLdqdot_Minus[2],
                              dLdqdot_Plus[3]-dLdqdot_Minus[3],
                              dLdqdot_Plus[4]-dLdqdot_Minus[4],
                              dLdqdot_Plus[5]-dLdqdot_Minus[5],
                              H_Plus-H_Minus])

for i in range(16):
  impact_eqns_rhs = sym.Matrix([lamb*dPhidq_Minus[i,0],
                                lamb*dPhidq_Minus[i,1],
                                lamb*dPhidq_Minus[i,2],
                                lamb*dPhidq_Minus[i,3],
                                lamb*dPhidq_Minus[i,4],
                                lamb*dPhidq_Minus[i,5],
                                0])
  impact_eqns.append(sym.Eq(impact_eqns_lhs, impact_eqns_rhs))
display(impact_eqns)
```

```
[Eq(Matrix([
 [                                                                    ␣
 ↪                                                                     ␣
 ↪                                                                     ␣
 ↪      0.5*\dot{x1}^+ - 0.5*\dot{x1}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪     0.5*\dot{y1}^+ - 0.5*\dot{y1}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                         1.
 ↪0*\dot{\theta_1}^+ - 1.0*\dot{\theta_1}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪   40.0*\dot{x2}^+ - 40.0*\dot{x2}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪   40.0*\dot{y2}^+ - 40.0*\dot{y2}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                     364.
 ↪0*\dot{\theta_2}^+ - 364.0*\dot{\theta_2}^-],
 [0.5*\dot{\theta_1}^+**2 - 0.5*\dot{\theta_1}^-**2 + 182.0*\dot{\theta_2}^+**2␣
 ↪- 182.0*\dot{\theta_2}^-**2 + 0.25*\dot{x1}^+**2 - 0.25*\dot{x1}^-**2 + 20.
 ↪0*\dot{x2}^+**2 - 20.0*\dot{x2}^-**2 + 0.25*\dot{y1}^+**2 - 0.25*\dot{y1}^-**2␣
 ↪+ 20.0*\dot{y2}^+**2 - 20.0*\dot{y2}^-**2]]), Matrix([
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                  lambda*cos(theta_2^-)],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                  lambda*sin(theta_2^-)],
 [                                                                              ␣
 ↪                                      lambda*(-0.
 ↪5*sin(theta_1^-)*cos(theta_2^-) + 0.5*sin(theta_2^-)*cos(theta_1^-))],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪             -1.0*lambda*cos(theta_2^-)],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪             -1.0*lambda*sin(theta_2^-)],
 [lambda*(-(x1^- + 0.5*cos(theta_1^-))*sin(theta_2^-) - (-1.0*x2^- - 3.
 ↪0*cos(theta_2^-))*sin(theta_2^-) + (y1^- + 0.5*sin(theta_1^-))*cos(theta_2^-)␣
 ↪+ (-1.0*y2^- - 3.0*sin(theta_2^-))*cos(theta_2^-))],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                      0]])),
 Eq(Matrix([
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪     0.5*\dot{x1}^+ - 0.5*\dot{x1}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪     0.5*\dot{y1}^+ - 0.5*\dot{y1}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                        1.
 ↪0*\dot{\theta_1}^+ - 1.0*\dot{\theta_1}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪   40.0*\dot{x2}^+ - 40.0*\dot{x2}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪   40.0*\dot{y2}^+ - 40.0*\dot{y2}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                    364.
 ↪0*\dot{\theta_2}^+ - 364.0*\dot{\theta_2}^-],
 [0.5*\dot{\theta_1}^+**2 - 0.5*\dot{\theta_1}^-**2 + 182.0*\dot{\theta_2}^+**2␣
 ↪- 182.0*\dot{\theta_2}^-**2 + 0.25*\dot{x1}^+**2 - 0.25*\dot{x1}^-**2 + 20.
 ↪0*\dot{x2}^+**2 - 20.0*\dot{x2}^-**2 + 0.25*\dot{y1}^+**2 - 0.25*\dot{y1}^-**2␣
 ↪+ 20.0*\dot{y2}^+**2 - 20.0*\dot{y2}^-**2]]), Matrix([
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                  lambda*cos(theta_2^-)],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                  lambda*sin(theta_2^-)],
 [                                                                              ␣
 ↪                                        lambda*(0.
 ↪5*sin(theta_1^-)*sin(theta_2^-) + 0.5*cos(theta_1^-)*cos(theta_2^-))],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪             -1.0*lambda*cos(theta_2^-)],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪             -1.0*lambda*sin(theta_2^-)],
 [lambda*(-(x1^- + 0.5*sin(theta_1^-))*sin(theta_2^-) - (-1.0*x2^- - 3.
 ↪0*cos(theta_2^-))*sin(theta_2^-) + (y1^- - 0.5*cos(theta_1^-))*cos(theta_2^-)␣
 ↪+ (-1.0*y2^- - 3.0*sin(theta_2^-))*cos(theta_2^-))],
 [                                                                                           ␣
 ↪                                                                                            ␣
 ↪                                       0]])),
 Eq(Matrix([
 [                                                                                           ␣
 ↪                                                                                            ␣
 ↪                                                                                            ␣
 ↪      0.5*\dot{x1}^+ - 0.5*\dot{x1}^-],
 [                                                                                           ␣
 ↪                                                                                            ␣
 ↪                                                                                            ␣
 ↪      0.5*\dot{y1}^+ - 0.5*\dot{y1}^-],
 [                                                                                           ␣
 ↪                                                                                            ␣
 ↪                                                                                      1.
 ↪0*\dot{\theta_1}^+ - 1.0*\dot{\theta_1}^-],
 [                                                                                           ␣
 ↪                                                                                            ␣
 ↪                                                                                            ␣
 ↪    40.0*\dot{x2}^+ - 40.0*\dot{x2}^-],
 [                                                                                           ␣
 ↪                                                                                            ␣
 ↪                                                                                            ␣
 ↪    40.0*\dot{y2}^+ - 40.0*\dot{y2}^-],
 [                                                                                           ␣
 ↪                                                                                            ␣
 ↪                                                                                  364.
 ↪0*\dot{\theta_2}^+ - 364.0*\dot{\theta_2}^-],
 [0.5*\dot{\theta_1}^+**2 - 0.5*\dot{\theta_1}^-**2 + 182.0*\dot{\theta_2}^+**2␣
 ↪- 182.0*\dot{\theta_2}^-**2 + 0.25*\dot{x1}^+**2 - 0.25*\dot{x1}^-**2 + 20.
 ↪0*\dot{x2}^+**2 - 20.0*\dot{x2}^-**2 + 0.25*\dot{y1}^+**2 - 0.25*\dot{y1}^-**2␣
 ↪+ 20.0*\dot{y2}^+**2 - 20.0*\dot{y2}^-**2]]), Matrix([
 [                                                                                           ␣
 ↪                                                                                            ␣
 ↪                   lambda*cos(theta_2^-)],
 [                                                                                           ␣
 ↪                                                                                            ␣
 ↪                   lambda*sin(theta_2^-)],
 [                                                                                           ␣
 ↪                                         lambda*(0.
 ↪5*sin(theta_1^-)*cos(theta_2^-) - 0.5*sin(theta_2^-)*cos(theta_1^-))],
 [                                                                                           ␣
 ↪                                                                                            ␣
 ↪              -1.0*lambda*cos(theta_2^-)],
 [                                                                                           ␣
 ↪                                                                                            ␣
 ↪              -1.0*lambda*sin(theta_2^-)],
 [lambda*(-(x1^- - 0.5*cos(theta_1^-))*sin(theta_2^-) - (-1.0*x2^- - 3.
 ↪0*cos(theta_2^-))*sin(theta_2^-) + (y1^- - 0.5*sin(theta_1^-))*cos(theta_2^-)␣
 ↪+ (-1.0*y2^- - 3.0*sin(theta_2^-))*cos(theta_2^-))],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                     0]])),
 Eq(Matrix([
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪     0.5*\dot{x1}^+ - 0.5*\dot{x1}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪     0.5*\dot{y1}^+ - 0.5*\dot{y1}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                     1.
 ↪0*\dot{\theta_1}^+ - 1.0*\dot{\theta_1}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪   40.0*\dot{x2}^+ - 40.0*\dot{x2}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪   40.0*\dot{y2}^+ - 40.0*\dot{y2}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                 364.
 ↪0*\dot{\theta_2}^+ - 364.0*\dot{\theta_2}^-],
 [0.5*\dot{\theta_1}^+**2 - 0.5*\dot{\theta_1}^-**2 + 182.0*\dot{\theta_2}^+**2␣
 ↪- 182.0*\dot{\theta_2}^-**2 + 0.25*\dot{x1}^+**2 - 0.25*\dot{x1}^-**2 + 20.
 ↪0*\dot{x2}^+**2 - 20.0*\dot{x2}^-**2 + 0.25*\dot{y1}^+**2 - 0.25*\dot{y1}^-**2␣
 ↪+ 20.0*\dot{y2}^+**2 - 20.0*\dot{y2}^-**2]]), Matrix([
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                  lambda*cos(theta_2^-)],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                  lambda*sin(theta_2^-)],
 [                                                                              ␣
 ↪                                        lambda*(-0.
 ↪5*sin(theta_1^-)*sin(theta_2^-) - 0.5*cos(theta_1^-)*cos(theta_2^-))],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪             -1.0*lambda*cos(theta_2^-)],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪             -1.0*lambda*sin(theta_2^-)],
 [lambda*(-(x1^- - 0.5*sin(theta_1^-))*sin(theta_2^-) - (-1.0*x2^- - 3.
 ↪0*cos(theta_2^-))*sin(theta_2^-) + (y1^- + 0.5*cos(theta_1^-))*cos(theta_2^-)␣
 ↪+ (-1.0*y2^- - 3.0*sin(theta_2^-))*cos(theta_2^-))],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                    0]])),
 Eq(Matrix([
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪     0.5*\dot{x1}^+ - 0.5*\dot{x1}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪     0.5*\dot{y1}^+ - 0.5*\dot{y1}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                      1.
 ↪0*\dot{\theta_1}^+ - 1.0*\dot{\theta_1}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪   40.0*\dot{x2}^+ - 40.0*\dot{x2}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪   40.0*\dot{y2}^+ - 40.0*\dot{y2}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                  364.
 ↪0*\dot{\theta_2}^+ - 364.0*\dot{\theta_2}^-],
 [0.5*\dot{\theta_1}^+**2 - 0.5*\dot{\theta_1}^-**2 + 182.0*\dot{\theta_2}^+**2␣
 ↪- 182.0*\dot{\theta_2}^-**2 + 0.25*\dot{x1}^+**2 - 0.25*\dot{x1}^-**2 + 20.
 ↪0*\dot{x2}^+**2 - 20.0*\dot{x2}^-**2 + 0.25*\dot{y1}^+**2 - 0.25*\dot{y1}^-**2␣
 ↪+ 20.0*\dot{y2}^+**2 - 20.0*\dot{y2}^-**2]]), Matrix([
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                -lambda*sin(theta_2^-)],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                 lambda*cos(theta_2^-)],
 [                                                                              ␣
 ↪                                     lambda*(0.
 ↪5*sin(theta_1^-)*sin(theta_2^-) + 0.5*cos(theta_1^-)*cos(theta_2^-))],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪             1.0*lambda*sin(theta_2^-)],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪            -1.0*lambda*cos(theta_2^-)],
 [lambda*((-x1^- - 0.5*cos(theta_1^-))*cos(theta_2^-) + (1.0*x2^- + 3.
 ↪0*sin(theta_2^-))*cos(theta_2^-) - (y1^- + 0.5*sin(theta_1^-))*sin(theta_2^-)␣
 ↪- (-1.0*y2^- + 3.0*cos(theta_2^-))*sin(theta_2^-))],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                     0]])),
 Eq(Matrix([
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪     0.5*\dot{x1}^+ - 0.5*\dot{x1}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪     0.5*\dot{y1}^+ - 0.5*\dot{y1}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                        1.
 ↪0*\dot{\theta_1}^+ - 1.0*\dot{\theta_1}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪   40.0*\dot{x2}^+ - 40.0*\dot{x2}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪   40.0*\dot{y2}^+ - 40.0*\dot{y2}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                    364.
 ↪0*\dot{\theta_2}^+ - 364.0*\dot{\theta_2}^-],
 [0.5*\dot{\theta_1}^+**2 - 0.5*\dot{\theta_1}^-**2 + 182.0*\dot{\theta_2}^+**2␣
 ↪- 182.0*\dot{\theta_2}^-**2 + 0.25*\dot{x1}^+**2 - 0.25*\dot{x1}^-**2 + 20.
 ↪0*\dot{x2}^+**2 - 20.0*\dot{x2}^-**2 + 0.25*\dot{y1}^+**2 - 0.25*\dot{y1}^-**2␣
 ↪+ 20.0*\dot{y2}^+**2 - 20.0*\dot{y2}^-**2]]), Matrix([
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                -lambda*sin(theta_2^-)],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                 lambda*cos(theta_2^-)],
 [                                                                             ␣
 ↪                                       lambda*(0.
 ↪5*sin(theta_1^-)*cos(theta_2^-) - 0.5*sin(theta_2^-)*cos(theta_1^-))],
 [                                                                             ␣
 ↪                                                                             ␣
 ↪             1.0*lambda*sin(theta_2^-)],
 [                                                                             ␣
 ↪                                                                             ␣
 ↪            -1.0*lambda*cos(theta_2^-)],
 [lambda*((-x1^- - 0.5*sin(theta_1^-))*cos(theta_2^-) + (1.0*x2^- + 3.
 ↪0*sin(theta_2^-))*cos(theta_2^-) - (y1^- - 0.5*cos(theta_1^-))*sin(theta_2^-)␣
 ↪- (-1.0*y2^- + 3.0*cos(theta_2^-))*sin(theta_2^-))],
 [                                                                             ␣
 ↪                                                                             ␣
 ↪                                     0]])),
 Eq(Matrix([
 [                                                                             ␣
 ↪                                                                             ␣
 ↪                                                                             ␣
 ↪     0.5*\dot{x1}^+ - 0.5*\dot{x1}^-],
 [                                                                             ␣
 ↪                                                                             ␣
 ↪                                                                             ␣
 ↪     0.5*\dot{y1}^+ - 0.5*\dot{y1}^-],
 [                                                                             ␣
 ↪                                                                             ␣
 ↪                                                                        1.
 ↪0*\dot{\theta_1}^+ - 1.0*\dot{\theta_1}^-],
 [                                                                             ␣
 ↪                                                                             ␣
 ↪                                                                             ␣
 ↪   40.0*\dot{x2}^+ - 40.0*\dot{x2}^-],
 [                                                                             ␣
 ↪                                                                             ␣
 ↪                                                                             ␣
 ↪   40.0*\dot{y2}^+ - 40.0*\dot{y2}^-],
 [                                                                             ␣
 ↪                                                                             ␣
 ↪                                                                    364.
 ↪0*\dot{\theta_2}^+ - 364.0*\dot{\theta_2}^-],
 [0.5*\dot{\theta_1}^+**2 - 0.5*\dot{\theta_1}^-**2 + 182.0*\dot{\theta_2}^+**2␣
 ↪- 182.0*\dot{\theta_2}^-**2 + 0.25*\dot{x1}^+**2 - 0.25*\dot{x1}^-**2 + 20.
 ↪0*\dot{x2}^+**2 - 20.0*\dot{x2}^-**2 + 0.25*\dot{y1}^+**2 - 0.25*\dot{y1}^-**2␣
 ↪+ 20.0*\dot{y2}^+**2 - 20.0*\dot{y2}^-**2]]), Matrix([
 [                                                                             ␣
 ↪                                                                             ␣
 ↪                lambda*sin(theta_2^-)],
 [                                                                              ␣
 ↪                                                                               ␣
 ↪                 lambda*cos(theta_2^-)],
 [                                                                              ␣
 ↪                                      lambda*(-0.
 ↪5*sin(theta_1^-)*sin(theta_2^-) - 0.5*cos(theta_1^-)*cos(theta_2^-))],
 [                                                                              ␣
 ↪                                                                               ␣
 ↪             1.0*lambda*sin(theta_2^-)],
 [                                                                              ␣
 ↪                                                                               ␣
 ↪            -1.0*lambda*cos(theta_2^-)],
 [lambda*((-x1^- + 0.5*cos(theta_1^-))*cos(theta_2^-) + (1.0*x2^- + 3.
 ↪0*sin(theta_2^-))*cos(theta_2^-) - (y1^- - 0.5*sin(theta_1^-))*sin(theta_2^-)␣
 ↪- (-1.0*y2^- + 3.0*cos(theta_2^-))*sin(theta_2^-))],
 [                                                                              ␣
 ↪                                                                               ␣
 ↪                                     0]])),
 Eq(Matrix([
 [                                                                              ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪     0.5*\dot{x1}^+ - 0.5*\dot{x1}^-],
 [                                                                              ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪     0.5*\dot{y1}^+ - 0.5*\dot{y1}^-],
 [                                                                              ␣
 ↪                                                                               ␣
 ↪                                                                        1.
 ↪0*\dot{\theta_1}^+ - 1.0*\dot{\theta_1}^-],
 [                                                                              ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪   40.0*\dot{x2}^+ - 40.0*\dot{x2}^-],
 [                                                                              ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪   40.0*\dot{y2}^+ - 40.0*\dot{y2}^-],
 [                                                                              ␣
 ↪                                                                               ␣
 ↪                                                                    364.
 ↪0*\dot{\theta_2}^+ - 364.0*\dot{\theta_2}^-],
 [0.5*\dot{\theta_1}^+**2 - 0.5*\dot{\theta_1}^-**2 + 182.0*\dot{\theta_2}^+**2␣
 ↪- 182.0*\dot{\theta_2}^-**2 + 0.25*\dot{x1}^+**2 - 0.25*\dot{x1}^-**2 + 20.
 ↪0*\dot{x2}^+**2 - 20.0*\dot{x2}^-**2 + 0.25*\dot{y1}^+**2 - 0.25*\dot{y1}^-**2␣
 ↪+ 20.0*\dot{y2}^+**2 - 20.0*\dot{y2}^-**2]]), Matrix([
 [                                                                              ␣
 ↪                                                                               ␣
 ↪                -lambda*sin(theta_2^-)],
 [                                                                              ␣
 ↪                                                                               ␣
 ↪                 lambda*cos(theta_2^-)],
 [                                                                              ␣
 ↪                                     lambda*(-0.
 ↪5*sin(theta_1^-)*cos(theta_2^-) + 0.5*sin(theta_2^-)*cos(theta_1^-))],
 [                                                                              ␣
 ↪                                                                               ␣
 ↪             1.0*lambda*sin(theta_2^-)],
 [                                                                              ␣
 ↪                                                                               ␣
 ↪            -1.0*lambda*cos(theta_2^-)],
 [lambda*((-x1^- + 0.5*sin(theta_1^-))*cos(theta_2^-) + (1.0*x2^- + 3.
 ↪0*sin(theta_2^-))*cos(theta_2^-) - (y1^- + 0.5*cos(theta_1^-))*sin(theta_2^-)␣
 ↪- (-1.0*y2^- + 3.0*cos(theta_2^-))*sin(theta_2^-))],
 [                                                                              ␣
 ↪                                                                               ␣
 ↪                                     0]])),
 Eq(Matrix([
 [                                                                              ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪     0.5*\dot{x1}^+ - 0.5*\dot{x1}^-],
 [                                                                              ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪     0.5*\dot{y1}^+ - 0.5*\dot{y1}^-],
 [                                                                              ␣
 ↪                                                                               ␣
 ↪                                                                       1.
 ↪0*\dot{\theta_1}^+ - 1.0*\dot{\theta_1}^-],
 [                                                                              ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪   40.0*\dot{x2}^+ - 40.0*\dot{x2}^-],
 [                                                                              ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪   40.0*\dot{y2}^+ - 40.0*\dot{y2}^-],
 [                                                                              ␣
 ↪                                                                               ␣
 ↪                                                                   364.
 ↪0*\dot{\theta_2}^+ - 364.0*\dot{\theta_2}^-],
 [0.5*\dot{\theta_1}^+**2 - 0.5*\dot{\theta_1}^-**2 + 182.0*\dot{\theta_2}^+**2␣
 ↪- 182.0*\dot{\theta_2}^-**2 + 0.25*\dot{x1}^+**2 - 0.25*\dot{x1}^-**2 + 20.
 ↪0*\dot{x2}^+**2 - 20.0*\dot{x2}^-**2 + 0.25*\dot{y1}^+**2 - 0.25*\dot{y1}^-**2␣
 ↪+ 20.0*\dot{y2}^+**2 - 20.0*\dot{y2}^-**2]]), Matrix([
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                     lambda*cos(theta_2^-)],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                     lambda*sin(theta_2^-)],
 [                                                                              ␣
 ↪                                         lambda*(-0.
 ↪5*sin(theta_1^-)*cos(theta_2^-) + 0.5*sin(theta_2^-)*cos(theta_1^-))],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                -1.0*lambda*cos(theta_2^-)],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                -1.0*lambda*sin(theta_2^-)],
 [lambda*(-(x1^- + 0.5*cos(theta_1^-))*sin(theta_2^-) - (-1.0*x2^- + 3.
 ↪0*cos(theta_2^-))*sin(theta_2^-) + (y1^- + 0.5*sin(theta_1^-))*cos(theta_2^-)␣
 ↪+ (-1.0*y2^- + 3.0*sin(theta_2^-))*cos(theta_2^-))],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                        0]])),
 Eq(Matrix([
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪      0.5*\dot{x1}^+ - 0.5*\dot{x1}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪      0.5*\dot{y1}^+ - 0.5*\dot{y1}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                       1.
 ↪0*\dot{\theta_1}^+ - 1.0*\dot{\theta_1}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪    40.0*\dot{x2}^+ - 40.0*\dot{x2}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪    40.0*\dot{y2}^+ - 40.0*\dot{y2}^-],
 [                                                                                          ␣
 ↪                                                                                          ␣
 ↪                                                                                  364.
 ↪0*\dot{\theta_2}^+ - 364.0*\dot{\theta_2}^-],
 [0.5*\dot{\theta_1}^+**2 - 0.5*\dot{\theta_1}^-**2 + 182.0*\dot{\theta_2}^+**2␣
 ↪- 182.0*\dot{\theta_2}^-**2 + 0.25*\dot{x1}^+**2 - 0.25*\dot{x1}^-**2 + 20.
 ↪0*\dot{x2}^+**2 - 20.0*\dot{x2}^-**2 + 0.25*\dot{y1}^+**2 - 0.25*\dot{y1}^-**2␣
 ↪+ 20.0*\dot{y2}^+**2 - 20.0*\dot{y2}^-**2]]), Matrix([
 [                                                                                          ␣
 ↪                                                                                          ␣
 ↪                  lambda*cos(theta_2^-)],
 [                                                                                          ␣
 ↪                                                                                          ␣
 ↪                  lambda*sin(theta_2^-)],
 [                                                                                          ␣
 ↪                                          lambda*(0.
 ↪5*sin(theta_1^-)*sin(theta_2^-) + 0.5*cos(theta_1^-)*cos(theta_2^-))],
 [                                                                                          ␣
 ↪                                                                                          ␣
 ↪             -1.0*lambda*cos(theta_2^-)],
 [                                                                                          ␣
 ↪                                                                                          ␣
 ↪             -1.0*lambda*sin(theta_2^-)],
 [lambda*(-(x1^- + 0.5*sin(theta_1^-))*sin(theta_2^-) - (-1.0*x2^- + 3.
 ↪0*cos(theta_2^-))*sin(theta_2^-) + (y1^- - 0.5*cos(theta_1^-))*cos(theta_2^-)␣
 ↪+ (-1.0*y2^- + 3.0*sin(theta_2^-))*cos(theta_2^-))],
 [                                                                                          ␣
 ↪                                                                                          ␣
 ↪                                        0]])),
 Eq(Matrix([
 [                                                                                          ␣
 ↪                                                                                          ␣
 ↪                                                                                          ␣
 ↪     0.5*\dot{x1}^+ - 0.5*\dot{x1}^-],
 [                                                                                          ␣
 ↪                                                                                          ␣
 ↪                                                                                          ␣
 ↪     0.5*\dot{y1}^+ - 0.5*\dot{y1}^-],
 [                                                                                          ␣
 ↪                                                                                          ␣
 ↪                                                                                      1.
 ↪0*\dot{\theta_1}^+ - 1.0*\dot{\theta_1}^-],
 [                                                                                          ␣
 ↪                                                                                          ␣
 ↪                                                                                          ␣
 ↪   40.0*\dot{x2}^+ - 40.0*\dot{x2}^-],
 [                                                                          ␣
 ↪                                                                          ␣
 ↪                                                                          ␣
 ↪   40.0*\dot{y2}^+ - 40.0*\dot{y2}^-],
 [                                                                          ␣
 ↪                                                                          ␣
 ↪                                                                     364.
 ↪0*\dot{\theta_2}^+ - 364.0*\dot{\theta_2}^-],
 [0.5*\dot{\theta_1}^+**2 - 0.5*\dot{\theta_1}^-**2 + 182.0*\dot{\theta_2}^+**2␣
 ↪- 182.0*\dot{\theta_2}^-**2 + 0.25*\dot{x1}^+**2 - 0.25*\dot{x1}^-**2 + 20.
 ↪0*\dot{x2}^+**2 - 20.0*\dot{x2}^-**2 + 0.25*\dot{y1}^+**2 - 0.25*\dot{y1}^-**2␣
 ↪+ 20.0*\dot{y2}^+**2 - 20.0*\dot{y2}^-**2]]), Matrix([
 [                                                                          ␣
 ↪                                                                          ␣
 ↪                  lambda*cos(theta_2^-)],
 [                                                                          ␣
 ↪                                                                          ␣
 ↪                  lambda*sin(theta_2^-)],
 [                                                                          ␣
 ↪                                       lambda*(0.
 ↪5*sin(theta_1^-)*cos(theta_2^-) - 0.5*sin(theta_2^-)*cos(theta_1^-))],
 [                                                                          ␣
 ↪                                                                          ␣
 ↪             -1.0*lambda*cos(theta_2^-)],
 [                                                                          ␣
 ↪                                                                          ␣
 ↪             -1.0*lambda*sin(theta_2^-)],
 [lambda*(-(x1^- - 0.5*cos(theta_1^-))*sin(theta_2^-) - (-1.0*x2^- + 3.
 ↪0*cos(theta_2^-))*sin(theta_2^-) + (y1^- - 0.5*sin(theta_1^-))*cos(theta_2^-)␣
 ↪+ (-1.0*y2^- + 3.0*sin(theta_2^-))*cos(theta_2^-))],
 [                                                                          ␣
 ↪                                                                          ␣
 ↪                                     0]])),
 Eq(Matrix([
 [                                                                          ␣
 ↪                                                                          ␣
 ↪                                                                          ␣
 ↪     0.5*\dot{x1}^+ - 0.5*\dot{x1}^-],
 [                                                                          ␣
 ↪                                                                          ␣
 ↪                                                                          ␣
 ↪     0.5*\dot{y1}^+ - 0.5*\dot{y1}^-],
 [                                                                          ␣
 ↪                                                                          ␣
 ↪                                                                        1.
 ↪0*\dot{\theta_1}^+ - 1.0*\dot{\theta_1}^-],
 [                                                                             ␣
 ↪                                                                             ␣
 ↪                                                                             ␣
 ↪   40.0*\dot{x2}^+ - 40.0*\dot{x2}^-],
 [                                                                             ␣
 ↪                                                                             ␣
 ↪                                                                             ␣
 ↪   40.0*\dot{y2}^+ - 40.0*\dot{y2}^-],
 [                                                                             ␣
 ↪                                                                             ␣
 ↪                                                                   364.
 ↪0*\dot{\theta_2}^+ - 364.0*\dot{\theta_2}^-],
 [0.5*\dot{\theta_1}^+**2 - 0.5*\dot{\theta_1}^-**2 + 182.0*\dot{\theta_2}^+**2␣
 ↪- 182.0*\dot{\theta_2}^-**2 + 0.25*\dot{x1}^+**2 - 0.25*\dot{x1}^-**2 + 20.
 ↪0*\dot{x2}^+**2 - 20.0*\dot{x2}^-**2 + 0.25*\dot{y1}^+**2 - 0.25*\dot{y1}^-**2␣
 ↪+ 20.0*\dot{y2}^+**2 - 20.0*\dot{y2}^-**2]]), Matrix([
 [                                                                             ␣
 ↪                                                                             ␣
 ↪                  lambda*cos(theta_2^-)],
 [                                                                             ␣
 ↪                                                                             ␣
 ↪                  lambda*sin(theta_2^-)],
 [                                                                             ␣
 ↪                                       lambda*(-0.
 ↪5*sin(theta_1^-)*sin(theta_2^-) - 0.5*cos(theta_1^-)*cos(theta_2^-))],
 [                                                                             ␣
 ↪                                                                             ␣
 ↪             -1.0*lambda*cos(theta_2^-)],
 [                                                                             ␣
 ↪                                                                             ␣
 ↪             -1.0*lambda*sin(theta_2^-)],
 [lambda*(-(x1^- - 0.5*sin(theta_1^-))*sin(theta_2^-) - (-1.0*x2^- + 3.
 ↪0*cos(theta_2^-))*sin(theta_2^-) + (y1^- + 0.5*cos(theta_1^-))*cos(theta_2^-)␣
 ↪+ (-1.0*y2^- + 3.0*sin(theta_2^-))*cos(theta_2^-))],
 [                                                                             ␣
 ↪                                                                             ␣
 ↪                                      0]])),
 Eq(Matrix([
 [                                                                             ␣
 ↪                                                                             ␣
 ↪                                                                             ␣
 ↪     0.5*\dot{x1}^+ - 0.5*\dot{x1}^-],
 [                                                                             ␣
 ↪                                                                             ␣
 ↪                                                                             ␣
 ↪     0.5*\dot{y1}^+ - 0.5*\dot{y1}^-],
 [                                                                        ␣
 ↪                                                                        ␣
 ↪                                                                  1.
 ↪0*\dot{\theta_1}^+ - 1.0*\dot{\theta_1}^-],
 [                                                                        ␣
 ↪                                                                        ␣
 ↪                                                                        ␣
 ↪   40.0*\dot{x2}^+ - 40.0*\dot{x2}^-],
 [                                                                        ␣
 ↪                                                                        ␣
 ↪                                                                        ␣
 ↪   40.0*\dot{y2}^+ - 40.0*\dot{y2}^-],
 [                                                                        ␣
 ↪                                                                        ␣
 ↪                                                               364.
 ↪0*\dot{\theta_2}^+ - 364.0*\dot{\theta_2}^-],
 [0.5*\dot{\theta_1}^+**2 - 0.5*\dot{\theta_1}^-**2 + 182.0*\dot{\theta_2}^+**2␣
 ↪- 182.0*\dot{\theta_2}^-**2 + 0.25*\dot{x1}^+**2 - 0.25*\dot{x1}^-**2 + 20.
 ↪0*\dot{x2}^+**2 - 20.0*\dot{x2}^-**2 + 0.25*\dot{y1}^+**2 - 0.25*\dot{y1}^-**2␣
 ↪+ 20.0*\dot{y2}^+**2 - 20.0*\dot{y2}^-**2]]), Matrix([
 [                                                                        ␣
 ↪                                                                        ␣
 ↪                -lambda*sin(theta_2^-)],
 [                                                                        ␣
 ↪                                                                        ␣
 ↪                 lambda*cos(theta_2^-)],
 [                                                                        ␣
 ↪                                      lambda*(0.
 ↪5*sin(theta_1^-)*sin(theta_2^-) + 0.5*cos(theta_1^-)*cos(theta_2^-))],
 [                                                                        ␣
 ↪                                                                        ␣
 ↪             1.0*lambda*sin(theta_2^-)],
 [                                                                        ␣
 ↪                                                                        ␣
 ↪            -1.0*lambda*cos(theta_2^-)],
 [lambda*((-x1^- - 0.5*cos(theta_1^-))*cos(theta_2^-) + (1.0*x2^- - 3.
 ↪0*sin(theta_2^-))*cos(theta_2^-) - (y1^- + 0.5*sin(theta_1^-))*sin(theta_2^-)␣
 ↪- (-1.0*y2^- - 3.0*cos(theta_2^-))*sin(theta_2^-))],
 [                                                                        ␣
 ↪                                                                        ␣
 ↪                                    0]])),
 Eq(Matrix([
 [                                                                        ␣
 ↪                                                                        ␣
 ↪                                                                        ␣
 ↪     0.5*\dot{x1}^+ - 0.5*\dot{x1}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪     0.5*\dot{y1}^+ - 0.5*\dot{y1}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                          1.
 ↪0*\dot{\theta_1}^+ - 1.0*\dot{\theta_1}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪   40.0*\dot{x2}^+ - 40.0*\dot{x2}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪   40.0*\dot{y2}^+ - 40.0*\dot{y2}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                      364.
 ↪0*\dot{\theta_2}^+ - 364.0*\dot{\theta_2}^-],
 [0.5*\dot{\theta_1}^+**2 - 0.5*\dot{\theta_1}^-**2 + 182.0*\dot{\theta_2}^+**2␣
 ↪- 182.0*\dot{\theta_2}^-**2 + 0.25*\dot{x1}^+**2 - 0.25*\dot{x1}^-**2 + 20.
 ↪0*\dot{x2}^+**2 - 20.0*\dot{x2}^-**2 + 0.25*\dot{y1}^+**2 - 0.25*\dot{y1}^-**2␣
 ↪+ 20.0*\dot{y2}^+**2 - 20.0*\dot{y2}^-**2]]), Matrix([
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                -lambda*sin(theta_2^-)],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                 lambda*cos(theta_2^-)],
 [                                                                              ␣
 ↪                                         lambda*(0.
 ↪5*sin(theta_1^-)*cos(theta_2^-) - 0.5*sin(theta_2^-)*cos(theta_1^-))],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪             1.0*lambda*sin(theta_2^-)],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪            -1.0*lambda*cos(theta_2^-)],
 [lambda*((-x1^- - 0.5*sin(theta_1^-))*cos(theta_2^-) + (1.0*x2^- - 3.
 ↪0*sin(theta_2^-))*cos(theta_2^-) - (y1^- - 0.5*cos(theta_1^-))*sin(theta_2^-)␣
 ↪- (-1.0*y2^- - 3.0*cos(theta_2^-))*sin(theta_2^-))],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                      0]])),
 Eq(Matrix([
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪     0.5*\dot{x1}^+ - 0.5*\dot{x1}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪     0.5*\dot{y1}^+ - 0.5*\dot{y1}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                        1.
 ↪0*\dot{\theta_1}^+ - 1.0*\dot{\theta_1}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪   40.0*\dot{x2}^+ - 40.0*\dot{x2}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪   40.0*\dot{y2}^+ - 40.0*\dot{y2}^-],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                    364.
 ↪0*\dot{\theta_2}^+ - 364.0*\dot{\theta_2}^-],
 [0.5*\dot{\theta_1}^+**2 - 0.5*\dot{\theta_1}^-**2 + 182.0*\dot{\theta_2}^+**2␣
 ↪- 182.0*\dot{\theta_2}^-**2 + 0.25*\dot{x1}^+**2 - 0.25*\dot{x1}^-**2 + 20.
 ↪0*\dot{x2}^+**2 - 20.0*\dot{x2}^-**2 + 0.25*\dot{y1}^+**2 - 0.25*\dot{y1}^-**2␣
 ↪+ 20.0*\dot{y2}^+**2 - 20.0*\dot{y2}^-**2]]), Matrix([
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                -lambda*sin(theta_2^-)],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪                 lambda*cos(theta_2^-)],
 [                                                                              ␣
 ↪                                      lambda*(-0.
 ↪5*sin(theta_1^-)*sin(theta_2^-) - 0.5*cos(theta_1^-)*cos(theta_2^-))],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪             1.0*lambda*sin(theta_2^-)],
 [                                                                              ␣
 ↪                                                                              ␣
 ↪            -1.0*lambda*cos(theta_2^-)],
 [lambda*((-x1^- + 0.5*cos(theta_1^-))*cos(theta_2^-) + (1.0*x2^- - 3.
 ↪0*sin(theta_2^-))*cos(theta_2^-) - (y1^- - 0.5*sin(theta_1^-))*sin(theta_2^-)␣
 ↪- (-1.0*y2^- - 3.0*cos(theta_2^-))*sin(theta_2^-))],
 [                                                                               ␣
 ↪                                                                               ␣
 ↪                                      0]])),
 Eq(Matrix([
 [                                                                               ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪     0.5*\dot{x1}^+ - 0.5*\dot{x1}^-],
 [                                                                               ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪     0.5*\dot{y1}^+ - 0.5*\dot{y1}^-],
 [                                                                               ␣
 ↪                                                                               ␣
 ↪                                                                          1.
 ↪0*\dot{\theta_1}^+ - 1.0*\dot{\theta_1}^-],
 [                                                                               ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪   40.0*\dot{x2}^+ - 40.0*\dot{x2}^-],
 [                                                                               ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪   40.0*\dot{y2}^+ - 40.0*\dot{y2}^-],
 [                                                                               ␣
 ↪                                                                               ␣
 ↪                                                                      364.
 ↪0*\dot{\theta_2}^+ - 364.0*\dot{\theta_2}^-],
 [0.5*\dot{\theta_1}^+**2 - 0.5*\dot{\theta_1}^-**2 + 182.0*\dot{\theta_2}^+**2␣
 ↪- 182.0*\dot{\theta_2}^-**2 + 0.25*\dot{x1}^+**2 - 0.25*\dot{x1}^-**2 + 20.
 ↪0*\dot{x2}^+**2 - 20.0*\dot{x2}^-**2 + 0.25*\dot{y1}^+**2 - 0.25*\dot{y1}^-**2␣
 ↪+ 20.0*\dot{y2}^+**2 - 20.0*\dot{y2}^-**2]]), Matrix([
 [                                                                               ␣
 ↪                                                                               ␣
 ↪                -lambda*sin(theta_2^-)],
 [                                                                               ␣
 ↪                                                                               ␣
 ↪                 lambda*cos(theta_2^-)],
 [                                                                               ␣
 ↪                                       lambda*(-0.
 ↪5*sin(theta_1^-)*cos(theta_2^-) + 0.5*sin(theta_2^-)*cos(theta_1^-))],
 [                                                                               ␣
 ↪                                                                               ␣
 ↪             1.0*lambda*sin(theta_2^-)],
 [                                                                               ␣
 ↪                                                                               ␣
 ↪            -1.0*lambda*cos(theta_2^-)],
 [lambda*((-x1^- + 0.5*sin(theta_1^-))*cos(theta_2^-) + (1.0*x2^- - 3.
 ↪0*sin(theta_2^-))*cos(theta_2^-) - (y1^- + 0.5*cos(theta_1^-))*sin(theta_2^-)␣
 ↪- (-1.0*y2^- - 3.0*cos(theta_2^-))*sin(theta_2^-))],
 [                                                                               ␣
 ↪                                                                               ␣
 ↪                                           0]]))]
```

[10]:

```
# impact help function
global Phi_func
Phi_func = sym.
 ↪lambdify([x1,y1,theta1,x2,y2,theta2,x1dot,y1dot,theta1dot,x2dot,y2dot,theta2dot],␣
 ↪Phi_ini)
# detect impact
def impact_condition(s, threshold=1e-1):
  global Phi_func
  phi_val = Phi_func(*s)
  #print("shape")
  #print(phi_val.shape)
  for i in range(phi_val.shape[0]):
    #print(i)
    #print("check")
    #print(phi_val[0])
    if (phi_val[i] > -threshold) and (phi_val[i] < threshold):
      print("contact")
      return (True,i)
  return (False, None)
```

[11]:

```
# impact update
def impact_update(s,i):
  #x1dotP,y1dotP,theta1dotP, x2dotP,y2dotP,theta2dotP
  subs_minus = {x1M:s[0],
                y1M:s[1],
                theta1M:s[2],
                x2M:s[3],
                y2M:s[4],
                theta2M:s[5],
                x1dotM:s[6],
                y1dotM:s[7],
                theta1dotM:s[8],
                x2dotM:s[9],
                y2dotM:s[10],
                theta2dotM:s[11]}
  #subs_plus = {q[0]:s[0], q[1]:s[1], q[2]:s[2],q[3]:s[3], q[4]:s[4], q[5]:
 ↪s[5], qdot[0]:x1dotP, qdot[1]:y1dotP, qdot[2]:theta1dotP,qdot[3]:x2dotP,␣
 ↪qdot[4]:y2dotP, qdot[5]:theta2dotP}
  global impact_eqns
  impact_e = impact_eqns[i]
  impact_val = impact_e.subs(subs_minus)
  #print(impact_val)
  impact_solns = sym.solve(impact_val,␣
 ↪[x1dotP,y1dotP,theta1dotP,x2dotP,y2dotP,theta2dotP, lamb], dict=True)

  #print(impact_solns)

  for i in range(len(impact_solns)):
    if abs(impact_solns[i][lamb]) >= 1e-6:
      #print(impact_solns[i][lamb])
      correcti = i
      #print(i)
      break
  lambda1 = impact_solns[correcti][lamb]
  #print(lambda1)
  impact_soln1 = impact_solns[correcti]

  x1dot_upsol = impact_soln1[x1dotP]
  y1dot_upsol = impact_soln1[y1dotP]
  theta1dot_upsol = impact_soln1[theta1dotP]
  x2dot_upsol = impact_soln1[x2dotP]
  y2dot_upsol = impact_soln1[y2dotP]
  theta2dot_upsol = impact_soln1[theta2dotP]
  #print(x1dot_upsol)
  ans = np.array([s[0],s[1],s[2],s[3],s[4],s[5],
                   impact_soln1[x1dotP],
                   impact_soln1[y1dotP],
                   impact_soln1[theta1dotP],
                   impact_soln1[x2dotP],
                   impact_soln1[y2dotP],
                   impact_soln1[theta2dotP]])
  #print(ans)
  return ans
```

[12]:

```
v1 = qddot[0]
v2 = qddot[1]
v3 = qddot[2]
v4 = qddot[3]
v5 = qddot[4]
v6 = qddot[5]

global x1acc
x1acc = sym.lambdify([[x1,y1,theta1,x2,y2,theta2,x1dot, y1dot,theta1dot,x2dot,␣
 ↪y2dot,theta2dot,t]], sol[v1])
global y1acc
y1acc = sym.lambdify([[x1,y1,theta1,x2,y2,theta2,x1dot, y1dot,theta1dot,x2dot,␣
 ↪y2dot,theta2dot,t]], sol[v2])
```

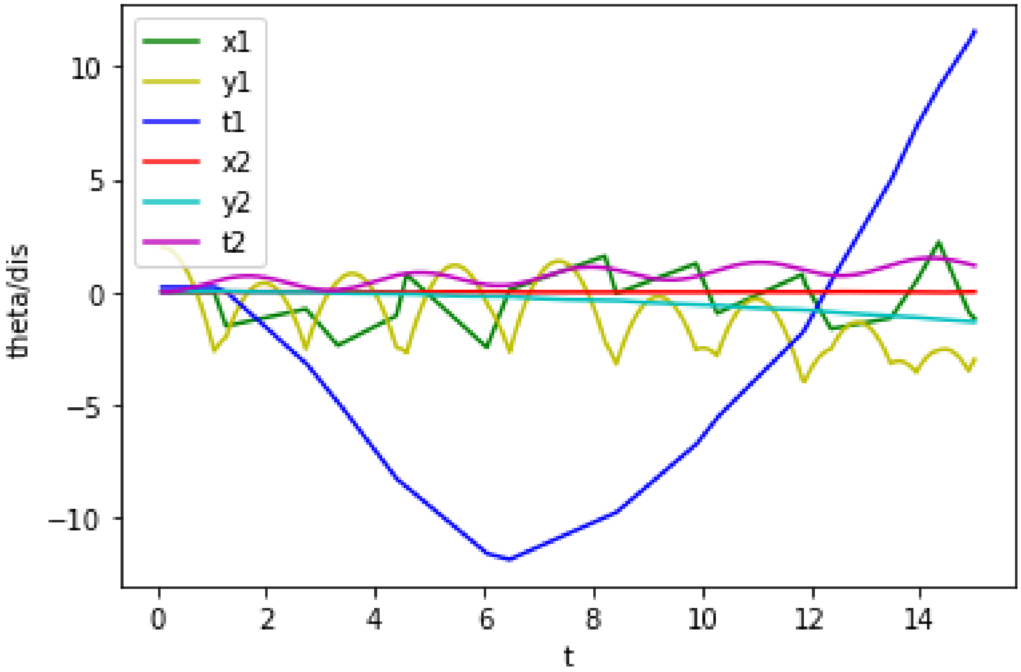
```
global theta1acc
theta1acc = sym.lambdify([[x1,y1,theta1,x2,y2,theta2,x1dot,␣
 ↪y1dot,theta1dot,x2dot, y2dot,theta2dot,t]], sol[v3])
global x2acc
x2acc = sym.lambdify([[x1,y1,theta1,x2,y2,theta2,x1dot, y1dot,theta1dot,x2dot,␣
 ↪y2dot,theta2dot,t]], sol[v4])
global y2acc
y2acc = sym.lambdify([[x1,y1,theta1,x2,y2,theta2,x1dot, y1dot,theta1dot,x2dot,␣
 ↪y2dot,theta2dot,t]], sol[v5])
global theta2acc
theta2acc = sym.lambdify([[x1,y1,theta1,x2,y2,theta2,x1dot,␣
 ↪y1dot,theta1dot,x2dot, y2dot,theta2dot,t]], sol[v6])

test = y1acc([0, 0, np.pi/15, 0,0,-np.pi/15,0,0,0,0,0,0,0])
print(test)

# def impact_update(s,i):
#   #x1dotP,y1dotP,theta1dotP, x2dotP,y2dotP,theta2dotP
#   subs_minus = {x1M:s[0],
#                 y1M:s[1],
#                 theta1M:s[2],
#                 x2M:s[3],
#                 y2M:s[4],
#                 theta2M:s[5],
#                 x1dotM:s[6],
#                 y1dotM:s[7],
#                 theta1dotM:s[8],
#                 x2dotM:s[9],
#                 y2dotM:s[10],
#                 theta2dotM:s[11]}
#   #subs_plus = {q[0]:s[0], q[1]:s[1], q[2]:s[2],q[3]:s[3], q[4]:s[4], q[5]:
 ↪s[5], qdot[0]:x1dotP, qdot[1]:y1dotP, qdot[2]:theta1dotP,qdot[3]:x2dotP,␣
 ↪qdot[4]:y2dotP, qdot[5]:theta2dotP}
#   global impact_eqns
#   impact_e = impact_eqns[i]
#   impact_val = impact_e.subs(subs_minus)
#   #print(impact_val)
#   impact_solns = sym.solve(impact_val,␣
 ↪[x1dotP,y1dotP,theta1dotP,x2dotP,y2dotP,theta2dotP, lamb], dict=True)

#   #print(impact_solns)

#   for i in range(len(impact_solns)):
#     if abs(impact_solns[i][lamb]) >= 1e-6:
#       #print(impact_solns[i][lamb])
#       correcti = i
#       #print(i)
#       break
#   lambda1 = impact_solns[correcti][lamb]
#   #print(lambda1)
#   impact_soln1 = impact_solns[correcti]

#   x1dot_upsol = impact_soln1[x1dotP]
#   y1dot_upsol = impact_soln1[y1dotP]
#   theta1dot_upsol = impact_soln1[theta1dotP]
#   x2dot_upsol = impact_soln1[x2dotP]
#   y2dot_upsol = impact_soln1[y2dotP]
#   theta2dot_upsol = impact_soln1[theta2dotP]
#   #print(x1dot_upsol)
#   ans = np.array([s[0],s[1],s[2],s[3],s[4],s[5],
#                    impact_soln1[x1dotP],
#                    impact_soln1[y1dotP],
#                    impact_soln1[theta1dotP],
#                    impact_soln1[x2dotP],
#                    impact_soln1[y2dotP],
#                    impact_soln1[theta2dotP]])
#   #print(ans)
#   return ans

def simulate(f, x0, tspan, dt, integrate):#, acc, anacc):
    """
    This function takes in an initial condition x0, a timestep dt,
    a time span tspan consisting of a list [min_time, max_time],
    as well as a dynamical system f(x) that outputs a vector of the
    same dimension as x0. It outputs a full trajectory simulated
    over the time span of dimensions (xvec_size, time_vec_size).

    Parameters
    ============
    f: Python function
        derivate of the system at a given step x(t),
        it can considered as \dot{x}(t) = func(x(t))
    x0: NumPy array
        initial conditions
    tspan: Python list
        tspan = [min_time, max_time], it defines the start and end
        time of simulation
    dt:
        time step for numerical integration
    integrate: Python function
        numerical integration method used in this simulation

    Return
    ============
    x_traj:
        simulated trajectory of x(t) from t=0 to tf
    """
    N = int((max(tspan)-min(tspan))/dt)
    x = np.copy(x0)
    tvec = np.linspace(min(tspan),max(tspan),N)
    xtraj = np.zeros((len(x0),N))
    time=0.0
    for i in range(N):
        #print(x)
        #print(dt)
        #print(time)
        impact,funcnum = impact_condition(x, threshold=1e-1)
        if impact is True:
          print(funcnum)
          x = impact_update(x,funcnum)
          xtraj[:,i]=integrate(f,x,dt,time)
        else:
          xtraj[:,i]=integrate(f,x,dt,time)#, acc, anacc)
        x = np.copy(xtraj[:,i])
        time += 0.01
    return xtraj


def xddot1(x1,y1,theta1,x2,y2,theta2,x1dot, y1dot,theta1dot,x2dot,␣
 ↪y2dot,theta2dot,t):#, acc, anacc):
    """
    Acceleration of the particle in terms of
    position and velocity. Here it's a constant.
    """
    global y1acc
    global x1acc
    global y2acc
    global x2acc
    global theta1acc
    global theta2acc
    x1dd = x1acc([x1,y1,theta1,x2,y2,theta2,x1dot, y1dot,theta1dot,x2dot,␣
 ↪y2dot,theta2dot,t])
    #print(ans1,acc([2.0, 1.0, 1.0, 9.8, 0.0, 0.1, 0.0,0.0]))
    y1dd = y1acc([x1,y1,theta1,x2,y2,theta2,x1dot, y1dot,theta1dot,x2dot,␣
 ↪y2dot,theta2dot,t])
    t1dd = theta1acc([x1,y1,theta1,x2,y2,theta2,x1dot, y1dot,theta1dot,x2dot,␣
 ↪y2dot,theta2dot,t])
    x2dd = x2acc([x1,y1,theta1,x2,y2,theta2,x1dot, y1dot,theta1dot,x2dot,␣
 ↪y2dot,theta2dot,t])
```

```
    y2dd = y2acc([x1,y1,theta1,x2,y2,theta2,x1dot, y1dot,theta1dot,x2dot,␣
 ↪y2dot,theta2dot,t])
    t2dd = theta2acc([x1,y1,theta1,x2,y2,theta2,x1dot, y1dot,theta1dot,x2dot,␣
 ↪y2dot,theta2dot,t])
    #ans = np.array([ans1, ans2])
    return x1dd,y1dd,t1dd,x2dd,y2dd,t2dd

def dyn(s,t):#,acc,anacc):
    """
    System dynamics function (extended)

    Parameters
    ============
    s: NumPy array
        s = [x, xdot] is the extended system
        state vector, includng the position and
        the velocity of the particle

    Return
    ============
    sdot: NumPy array
        time derivative of input state vector,
        sdot = [xdot, xddot]
    """
    x1ac, y1ac,theta1ac,x2ac, y2ac,theta2ac =␣
 ↪xddot1(s[0],s[1],s[2],s[3],s[4],s[5],s[6],s[7],s[8],s[9],s[10],s[11],t)#,acc,anacc)
    x1v = s[6]
    y1v = s[7]
    x2v = s[9]
    y2v = s[10]
    theta1v = s[8]
    theta2v = s[11]
    return np.array([x1v,y1v,theta1v,x2v,y2v,theta2v,x1ac, y1ac,theta1ac,x2ac,␣
 ↪y2ac,theta2ac])

def integrate(f, xt, dt,t):#, acc, anacc):
    """
    This function takes in an initial condition x(t) and a timestep dt,
    as well as a dynamical system f(x) that outputs a vector of the
    same dimension as x(t). It outputs a vector x(t+dt) at the future
    time step.

    Parameters
    ============
    dyn: Python function
        derivate of the system at a given step x(t),
        it can considered as \dot{x}(t) = func(x(t))
    xt: NumPy array
        current step x(t)
    dt:
        step size for integration

    Return
    ============
    new_xt:
        value of x(t+dt) integrated from x(t)
    """
    k1 = dt * f(xt,t)#, acc, anacc)
    k2 = dt * f(xt+k1/2.,t)#, acc, anacc)
    k3 = dt * f(xt+k2/2.,t)#, acc, anacc)
    k4 = dt * f(xt+k3,t)#, acc, anacc)
    new_xt = xt + (1/6.) * (k1+2.0*k2+2.0*k3+k4)
    return new_xt


s0 = np.array([0,2, np.pi/14, 0,0,0,0,0,0,0,0,0])
print(s0)


traj = simulate(dyn, s0, [0, 15], 0.01, integrate)#, acc, anacc)
print('\033[1mShape of traj: \033[0m', traj.shape)

t_list = np.linspace(0.1, 15, 1500)

plt.plot(t_list, traj[0,:],'g',t_list, traj[1,:],'y',t_list, traj[2,:
 ↪],'b',t_list, traj[3,:],'r',t_list, traj[4,:],'c',t_list, traj[5,:],'m')
plt.xlabel('t')
plt.ylabel('theta/dis')
plt.legend(['x1', 'y1','t1', 'x2','y2', 't2'])
```

```
-9.8
[0.         2.         0.22439948 0.         0.         0.
 0.         0.         0.         0.         0.         0.        ]
contact
5
contact
10
contact
7
contact
11
contact
8
contact
```

```
7
contact
10
contact
5
contact
6
contact
11
contact
4
contact
10
contact
7
contact
8
contact
10
contact
8
contact
10
contact
6
contact
8
Shape of traj:  (12, 1500)
```

```
[12]: <matplotlib.legend.Legend at 0x7f56d3dd5cc0>
```



[13]:
```
def animate_jack(p_array,T=5):
    """
    Function to generate web-based animation of double-pendulum system

    Parameters:
    ================================================
    theta_array:
        trajectory of x, y, theta1 and theta2, should be a NumPy array with
        shape of (4,N)or (8,N)
    L:
        length of the leg
    T:
        length/seconds of animation duration

    Returns: None
    """

    ################################
    # Imports required for animation.
    from plotly.offline import init_notebook_mode, iplot
    from IPython.display import display, HTML
    import plotly.graph_objects as go

    #######################
    # Browser configuration.
```

```
def configure_plotly_browser_state():
    import IPython
    display(IPython.core.display.HTML('''
        <script src="/static/components/requirejs/require.js"></script>
        <script>
          requirejs.config({
            paths: {
              base: '/static/base',
              plotly: 'https://cdn.plot.ly/plotly-1.5.1.min.js?noext',
            },
          });
        </script>
        '''))
configure_plotly_browser_state()
init_notebook_mode(connected=False)


###############################################
# Getting data from pendulum angle trajectories.
#xx1=L1*np.sin(theta_array[0])
#yy1=-L1*np.cos(theta_array[0])
#xx2=xx1+L2*np.sin(theta_array[0]+theta_array[1])
#yy2=yy1-L2*np.cos(theta_array[0]+theta_array[1])
N = len(p_array[0]) # Need this for specifying length of simulation

###############################################
# Define arrays containing data for frame axes
# In each frame, the x and y axis are always fixed
p1 = np.array([0, 0])
p2 = np.array([0, 0])
p3 = np.array([0, 0])
p4 = np.array([0, 0])
# Use homogeneous tranformation to transfer these two axes/points
# back to the fixed frame
frame_a_1_point = np.zeros((2,N))
frame_a_2_point = np.zeros((2,N))
frame_a_3_point = np.zeros((2,N))
frame_a_4_point = np.zeros((2,N))
frame_b1_1_axis = np.zeros((2,N))
frame_b1_2_axis = np.zeros((2,N))
frame_b1_3_axis = np.zeros((2,N))
frame_b1_4_axis = np.zeros((2,N))
frame_b2_1_axis = np.zeros((2,N))
frame_b2_2_axis = np.zeros((2,N))
frame_b2_3_axis = np.zeros((2,N))
frame_b2_4_axis = np.zeros((2,N))
for i in range(N): # iteration through each time step
```

```
        # evaluate homogeneous transformation
        t_wa = np.array([[sym.cos(p_array[2][i]), -sym.
↪sin(p_array[2][i]),p_array[0][i]],
                        [ sym.sin(p_array[2][i]), sym.
↪cos(p_array[2][i]),p_array[1][i]],
                         [                       0,                           ␣
↪0, 1]])
        # transfer the x and y axes in body frame back to fixed frame at
        # the current time step
        t_wb = np.array([[sym.cos(p_array[5][i]), -sym.
↪sin(p_array[5][i]),p_array[3][i]],
                        [ sym.sin(p_array[5][i]), sym.
↪cos(p_array[5][i]),p_array[4][i]],
                         [                       0,                           ␣
↪0, 1]])
        # a jack
        t_a1 = np.array([[1, 0, 0],
                        [0,  1, 0.5],
                        [0,0, 1]])
        t_a2 = np.array([[1, 0, 0.5],
                        [0,  1, 0],
                        [0,0, 1]])
        t_a3 = np.array([[1, 0, 0],
                        [0,  1, -0.5],
                        [0,0, 1]])
        t_a4 = np.array([[1, 0, -0.5],
                        [0,  1, 0],
                        [0,0, 1]])
        # b box
        t_b11 = np.array([[1, 0, 2.9],
                        [0,  1, 2.9],
                        [0,0, 1]])
        t_b12 = np.array([[1, 0, 2.9],
                        [0,  1, -2.9],
                        [0,0, 1]])
        t_b13 = np.array([[1, 0, -2.9],
                        [0,  1, -2.9],
                        [0,0, 1]])
        t_b14 = np.array([[1, 0, -2.9],
                        [0,  1, 2.9],
                        [0,0, 1]])

        t_b21 = np.array([[1, 0, 3.1],
                        [0,  1, 3.1],
                        [0,0, 1]])
        t_b22 = np.array([[1, 0, 3.1],
                        [0,  1, -3.1],
                        [0,0, 1]])
        t_b23 = np.array([[1, 0, -3.1],
                        [0,  1, -3.1],
                        [0,0, 1]])
        t_b24 = np.array([[1, 0, -3.1],
                        [0,  1, 3.1],
                        [0,0, 1]])

        t_wa1 = np.dot(t_wa,t_a1)
        t_wa2 = np.dot(t_wa,t_a2)
        t_wa3 = np.dot(t_wa,t_a3)
        t_wa4 = np.dot(t_wa,t_a4)

        t_wb11 = np.dot(t_wb,t_b11)
        t_wb12 = np.dot(t_wb,t_b12)
        t_wb13 = np.dot(t_wb,t_b13)
        t_wb14 = np.dot(t_wb,t_b14)

        t_wb21 = np.dot(t_wb,t_b21)
        t_wb22 = np.dot(t_wb,t_b22)
        t_wb23 = np.dot(t_wb,t_b23)
        t_wb24 = np.dot(t_wb,t_b24)

        #t_wc = np.dot(np.dot(t_wa,t_ac1),t_c1c)

        #print(np.dot(t_wb,np.array([p1[0], p1[1], 1])))


        frame_a_1_point[:,i] = t_wa1.dot([p1[0], p1[1], 1])[0:2]
        #print(frame_b_1_axis)
        frame_a_2_point[:,i] = t_wa2.dot([p2[0], p2[1], 1])[0:2]
        frame_a_3_point[:,i] = t_wa3.dot([p3[0], p3[1], 1])[0:2]
        frame_a_4_point[:,i] = t_wa4.dot([p4[0], p4[1], 1])[0:2]

        frame_b1_1_axis[:,i] = t_wb11.dot([p1[0], p1[1], 1])[0:2]
        frame_b1_2_axis[:,i] = t_wb12.dot([p2[0], p2[1], 1])[0:2]
        frame_b1_3_axis[:,i] = t_wb13.dot([p3[0], p3[1], 1])[0:2]
        frame_b1_4_axis[:,i] = t_wb14.dot([p4[0], p4[1], 1])[0:2]

        frame_b2_1_axis[:,i] = t_wb21.dot([p1[0], p1[1], 1])[0:2]
        frame_b2_2_axis[:,i] = t_wb22.dot([p2[0], p2[1], 1])[0:2]
        frame_b2_3_axis[:,i] = t_wb23.dot([p3[0], p3[1], 1])[0:2]
        frame_b2_4_axis[:,i] = t_wb24.dot([p4[0], p4[1], 1])[0:2]


    ####################################
```

```
    # Using these to specify axis limits.
    xm = -10 #np.min(xx1)-0.5
    xM = 10 #np.max(xx1)+0.5
    ym = -10 #np.min(yy1)-2.5
    yM = 10 #np.max(yy1)+1.5

    #print(frame_b_1_axis)

    ###########################
    # Defining data dictionary.
    # Trajectories are here.
    data=[
        # note that except for the trajectory (which you don't need this time),
        # you don't need to define entries other than "name". The items defined
        # in this list will be related to the items defined in the "frames" list
        # later in the same order. Therefore, these entries can be considered␣
↪as
        # labels for the components in each animation frame
        dict(name='jack1'),
        dict(name='jack2'),
        dict(name='box1_inside'),
        dict(name='box1_outside'),
        # dict(name='B Frame Y Axis'),
        # dict(name='C Frame X Axis'),
        # dict(name='C Frame Y Axis'),
        # dict(name='D Frame X Axis'),
        # dict(name='D Frame Y Axis'),

        # You don't need to show trajectory this time,
        # but if you want to show the whole trajectory in the animation (like␣
↪what
        # you did in previous homeworks), you will need to define entries other␣
↪than
        # "name", such as "x", "y". and "mode".

        # dict(x=xx1, y=yy1,
        #      mode='markers', name='Pendulum 1 Traj',
        #      marker=dict(color="fuchsia", size=2)
        #     ),
        # dict(x=xx2, y=yy2,
        #      mode='markers', name='Pendulum 2 Traj',
        #      marker=dict(color="purple", size=2)
        #     ),
        ]

    ##################################
    # Preparing simulation layout.
    # Title and axis ranges are here.
    layout=dict(autosize=False, width=1000, height=1000,
                xaxis=dict(range=[xm, xM], autorange=False,␣
↪zeroline=False,dtick=1),
                yaxis=dict(range=[ym, yM], autorange=False,␣
↪zeroline=False,scaleanchor = "x",dtick=1),
                title='Jack in a Box Simulation',
                hovermode='closest',
                updatemenus= [{'type': 'buttons',
                               'buttons': [{'label': 'Play','method': 'animate',
                                            'args': [None, {'frame':␣
↪{'duration': T, 'redraw': False}}]},
                                           {'args': [[None], {'frame':␣
↪{'duration': T, 'redraw': False}, 'mode': 'immediate',
                                            'transition': {'duration':␣
↪0}}],'label': 'Pause','method': 'animate'}
                                          ]
                              }]
               )

    ########################################
    # Defining the frames of the simulation.
    # This is what draws the lines from
    # joint to joint of the pendulum.
    frames=[dict(data=[# first three objects correspond to the arms and two␣
↪masses,
                          # same order as in the "data" variable defined above␣
↪(thus
                          # they will be labeled in the same order)
                         #  dict(x=[0,xx1[k],xx2[k]],
                         #       y=[0,yy1[k],yy2[k]],
                         #       mode='lines',
                         #       line=dict(color='orange', width=3),
                         #       ),
                         #  go.Scatter(
                         #       x=[xx1[k]],
                         #       y=[yy1[k]],
                         #       mode="markers",
                         #       marker=dict(color="blue", size=12)),
                         #  go.Scatter(
                         #       x=[xx2[k]],
                         #       y=[yy2[k]],
                         #       mode="markers",
                         #       marker=dict(color="blue", size=12)),
                          # display x and y axes of the fixed frame in each␣
↪animation frame
```

```
                       dict(x=[frame_a_1_point[0][k],frame_a_3_point[0][k]],
                            y=[frame_a_1_point[1][k],frame_a_3_point[1][k]],
                            mode='lines',
                            line=dict(color='red', width=3),
                            ),
                       dict(x=[frame_a_2_point[0][k],frame_a_4_point[0][k]],
                            y=[frame_a_2_point[1][k],frame_a_4_point[1][k]],
                            mode='lines',
                            line=dict(color='green', width=3),
                            ),
                      ␣
 ↪dict(x=[frame_b1_1_axis[0][k],frame_b1_2_axis[0][k],frame_b1_3_axis[0][k],frame_b1_4_axis[0
                           ␣
 ↪y=[frame_b1_1_axis[1][k],frame_b1_2_axis[1][k],frame_b1_3_axis[1][k],frame_b1_4_axis[1][k],
                            mode='lines',
                            line=dict(color='purple', width=3),
                            ),
                       # display x and y axes of the {A} frame in each␣
 ↪animation frame
                      ␣
 ↪dict(x=[frame_b2_1_axis[0][k],frame_b2_2_axis[0][k],frame_b2_3_axis[0][k],frame_b2_4_axis[0
                           ␣
 ↪y=[frame_b2_1_axis[1][k],frame_b2_2_axis[1][k],frame_b2_3_axis[1][k],frame_b2_4_axis[1][k],
                            mode='lines',
                            line=dict(color='blue', width=3),
                            ),
                      #  dict(x=[xx2[k], frame_d_y_axis[0][k]],
                      #       y=[yy2[k], frame_d_y_axis[1][k]],
                      #       mode='lines',
                      #       line=dict(color='red', width=3),
                      #       ),
                      ]) for k in range(N)]

    #######################################
    # Putting it all together and plotting.
    figure1=dict(data=data, layout=layout, frames=frames)
    iplot(figure1)

animate_jack(traj,T=15)
```

```
<IPython.core.display.HTML object>
```