PART B PENDULUM

December 11, 2023

```
[2]: from sympy import symbols, Matrix, pi, zeros, cos, sin
    # Define symbolic variables
    q1, q2, q3, q4 = symbols('q1 q2 q3 q4', real=True)
    d1, d2, d3, d4 = symbols('d1 d2 d3 d4', real=True)
    a1, a2 = symbols('a1 a2')
    g = symbols('g')
    # Initialize DH parameters
    DH = [
              0,
         [a1,
                      0, q1, 'R'],
         [a2,
             0,
                      0,
                               q2,
                                         'R']
    ]
    # Create length
    # This accounts for the prismatic or revolute
     # column not being included in the calculations
    LENGTH = len(DH)
    # Initialize transformation matrix
    T = Matrix.eye(4)
    # Initialize list for homogeneous transformations
    Ti = [None] * LENGTH
    # Define function to compute DH matrix
    def compute_dh_matrix(a, alpha, d, theta, joint_type):
         if joint_type == "R":
            return Matrix([
                 [cos(theta), -sin(theta)*cos(alpha), sin(theta)*sin(alpha),
      \Rightarrowa*cos(theta)],
                 [sin(theta), cos(theta)*cos(alpha), -cos(theta)*sin(alpha),__
      ⇔a*sin(theta)],
                 [0,
                             sin(alpha),
                                                      cos(alpha),
                                                                             d],
                 [0,
                             0,
                                                      0,
                                                                             1]
            ])
        else:
```

```
return Matrix([
                                                               [cos(alpha), -sin(alpha), 0, a*cos(alpha)],
                                                               [sin(alpha), cos(alpha), 0, a*sin(alpha)],
                                                                                                                                                            1, theta],
                                                               [0,
                                                                                                           0,
                                                               [0,
                                                                                                            0,
                                                                                                                                                          0, 1]
                                                ])
                    # Compute homogeneous transformations
                    for i in range(LENGTH):
                                  temp = compute_dh_matrix(*DH[i])
                                  T = T * temp
                                  Ti[i] = T
                    # Display the resulting homogeneous transformations
                    for i, transform in enumerate(Ti):
                                  print(f'T{i + 1} =')
                                  display(transform)
                  T1 =
                    \lceil \cos(q_1) - \sin(q_1) \quad 0 \quad a_1 \cos(q_1) \rceil
                      \sin(q_1) \quad \cos(q_1) \quad 0 \quad a_1 \sin(q_1)
                                                             0
                                                                                 1
                              0
                               0
                                                             0
                  T2 =
                    \lceil -\sin{(q_1)}\sin{(q_2)} + \cos{(q_1)}\cos{(q_2)} - \sin{(q_1)}\cos{(q_2)} - \sin{(q_2)}\cos{(q_1)} - 0 \quad a_1\cos{(q_1)} - a_2\sin{(q_1)}\sin{(q_2)} + a_2\cos{(q_2)} - a_2\sin{(q_2)}\cos{(q_2)} - a_2\cos{(q_2)}\cos{(q_2)} - a_2\cos{(q_2)}\cos{(q_2)}\cos{(q_2)}\cos{(q_2)} - a_2\cos{(q_2)}\cos{(q_2)}\cos{(q_2)} - a_2\cos{(q_2)}\cos{(q_2)}\cos{(q_2)} - a_2\cos{(q_2
                         \sin(q_1)\cos(q_2) + \sin(q_2)\cos(q_1) -\sin(q_1)\sin(q_2) + \cos(q_1)\cos(q_2) 0 a_1\sin(q_1) + a_2\sin(q_1)\cos(q_2) + a_2\sin(q_1)\cos(q_2)
                                                                                                                                                                               0
                                                                      0
                                                                                                                                                                               0
                                                                                                                                                                                                                                         0
[20]: # Define symbolic variables
                    Jw = [symbols(f'Jw{i+1}', real=True) for i in range(LENGTH)] # Angular_
                         ⇔velocity jacobian
                     # Angular velocity jacobian solution
                    Jw[0] = Matrix([[0], [0], [1]]).row_join(zeros(3, LENGTH-1))
                     # Calculate revolute joints
                    for i in range(1, LENGTH):
                                  jw = Matrix([[0], [0], [1]])
                                  for j in range(i):
                                                jw = jw.row_join(Ti[j][0:3, 2])
                                  jw = jw.row_join(zeros(3, LENGTH-1-i))
                                  Jw[i] = jw
```

1

```
# get indices of prismatic joints
      pris_indices = []
      for i in range(LENGTH):
          if DH[i][4] == 'P':
              pris_indices.append(i)
              print(f"Prismatic joint at index: {i}")
      if len(pris_indices) > 0:
          # Calculate prismatic and update each matrix
          m = \Gamma
          prismatic_matrix = Matrix([[0], [0], [0]])
          for i, jw_matrix in enumerate(Jw):
              for j in range(len(pris_indices)):
                  new_matrix = jw_matrix[:, :pris_indices[j]].
       -row_join(prismatic_matrix).row_join(jw_matrix[:, pris_indices[j]+1:])
                  m.append(new_matrix)
          # Update the matrix with the prismatic values
          Jw = m
      # Display the resulting angular velocity jacobian
      for i, jw_matrix in enumerate(Jw):
          print(f'Jw{i + 1} = ')
          display(jw_matrix)
          print("----")
     Jw1 =
      [0 \quad 0]
      0 0
     Jw2 =
      [0 0]
[21]: from sympy import symbols, Matrix, diff
      # Define symbolic variables
      q = [symbols(f'q{x}', real=True) for x in range(1, LENGTH+1)]
      c = [[symbols(f'c{x}x c{x}y c{x}z', real=True)] for x in range(1, LENGTH+1)]
      # Initialize linear velocity jacobian
```

```
Jv = [None] * LENGTH
          # Linear velocity jacobian solution
          # Dependent on joint type
          if DH[0][4] == 'R':
                 P = Matrix.eye(4)
          else:
                 P = Matrix.zeros(4)
          for i in range(LENGTH):
                 c_list = list(*c[i]) # Convert from tuple to list to unpack
                 P = Ti[i] * Matrix([[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [*c_list, ___
            →1]]).T
                 x = P[0, 3]
                 y = P[1, 3]
                 z = P[2, 3]
                 for j in range(LENGTH):
                         Jv[i] = Jv[i].row_join(Matrix([[x.diff(q[j])], [y.diff(q[j])], [z.]))

diff(q[j])]])) if Jv[i] else \

                               Matrix([[x.diff(q[j])], [y.diff(q[j])], [z.diff(q[j])]])
          # Display the resulting linear velocity jacobian
          for i, jv_matrix in enumerate(Jv):
                 print(f'Jv{i + 1} = ')
                 display(jv_matrix)
                 print("----")
         Jv1 =
           [-a_1 \sin{(q_1)} - c1x \sin{(q_1)} - c1y \cos{(q_1)} \quad 0]
            a_{1}\cos\left(q_{1}\right)+c1x\cos\left(q_{1}\right)-c1y\sin\left(q_{1}\right)
         Jv2 =
          a_{1}\cos\left(q_{1}\right)-a_{2}\sin\left(q_{1}\right)\sin\left(q_{2}\right)+a_{2}\cos\left(q_{1}\right)\cos\left(q_{2}\right)+c2x\left(-\sin\left(q_{1}\right)\sin\left(q_{2}\right)+\cos\left(q_{1}\right)\cos\left(q_{2}\right)\right)+c2y\left(-\sin\left(q_{1}\right)\cos\left(q_{2}\right)+\cos\left(q_{1}\right)\cos\left(q_{2}\right)\right)+c2y\left(-\sin\left(q_{1}\right)\cos\left(q_{2}\right)+\cos\left(q_{1}\right)\cos\left(q_{2}\right)\right)+c2y\left(-\sin\left(q_{1}\right)\cos\left(q_{2}\right)+\cos\left(q_{2}\right)\cos\left(q_{2}\right)\right)+c2y\left(-\sin\left(q_{1}\right)\cos\left(q_{2}\right)+\cos\left(q_{2}\right)\cos\left(q_{2}\right)\right)+c2y\left(-\sin\left(q_{1}\right)\cos\left(q_{2}\right)+\cos\left(q_{2}\right)\cos\left(q_{2}\right)\right)+c2y\left(-\sin\left(q_{2}\right)\cos\left(q_{2}\right)\cos\left(q_{2}\right)\cos\left(q_{2}\right)\cos\left(q_{2}\right)\cos\left(q_{2}\right)\right)
[22]: # Define symbolic variables
          m = [symbols(f'm{x}', real=True) for x in range(1, LENGTH+1)]
          # Potential energy solution
          P = Matrix.eye(4)
```

```
[23]: from sympy import symbols, Matrix, eye, Symbol, Function, symarray
      def inertia_tensor(num):
          n = str(num)
          symbols_list = [f'Ixx{n}', f'Ixy{n}', f'Ixz{n}',
                          f'Iyx{n}', f'Iyy{n}', f'Iyz{n}',
                          f'Izx{n}', f'Izy{n}', f'Izz{n}']
          tensor = symarray('', len(symbols_list)).reshape(3, 3)
          for i in range(3):
              for j in range(3):
                  tensor[i, j] = symbols_list[i * 3 + j]
          display(tensor)
          return tensor
      # Define symbolic variables
      qd = [symbols(f'qd\{x\}', real=True) for x in range(1, LENGTH+1)] # joint_{L}
      ⇔velocities
      g = Symbol('g', real=True) # gravitational acceleration
      # Inertia tensor for each link relative to the inertial frame stored in an nx1_{\sqcup}
      I = [inertia_tensor(i) for i in range(1, LENGTH + 1)]
     array([['Ixx1', 'Ixy1', 'Ixz1'],
            ['Iyx1', 'Iyy1', 'Iyz1'],
            ['Izx1', 'Izy1', 'Izz1']], dtype=object)
     array([['Ixx2', 'Ixy2', 'Ixz2'],
            ['Iyx2', 'Iyy2', 'Iyz2'],
            ['Izx2', 'Izy2', 'Izz2']], dtype=object)
[24]: # D = Inertia matrix solution & P = Potential Energy
      D = None
      PE = 0
      # Calculate D and PE
      for i in range(LENGTH):
          # Term one
          term_1 = (m[i] * Jv[i].T * Jv[i])
          # Term 2
          term_2 = Jw[i].T * I[i] * Jw[i]
```

PE = 0

```
if i < 1:
    D = term_1 + term_2
else:
    D = D + term_1 + term_2

c_list = list(*c[i]) # Convert from tuple to list to unpack
P = Ti[i] * Matrix([[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [*c_list,"])).T
PE += m[i] * g * P[2, 3]

# KE = Kinetic energy solution
q_dot_matrix = Matrix(qd)
KE = 0.5 * q_dot_matrix.T * D * q_dot_matrix

# Display the resulting kinetic energy expression
print("Kinetic Energy:")
display(KE)</pre>
```

Kinetic Energy:

```
\left[qd_{1}\cdot\left(0.5qd_{1}\left(Izz_{1}+Izz_{2}+m_{1}\left(-a_{1}\sin\left(q_{1}\right)-c1x\sin\left(q_{1}\right)-c1y\cos\left(q_{1}\right)\right)^{2}+m_{1}\left(a_{1}\cos\left(q_{1}\right)+c1x\cos\left(q_{1}\right)-c1y\sin\left(q_{1}\right)\right)\right]\right]+c_{1}^{2}\cos\left(q_{1}^{2}\right)+c_{2}^{2}\sin\left(q_{1}^{2}\right)+c_{3}^{2}\sin\left(q_{1}^{2}\right)+c_{4}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{2}\right)+c_{5}^{2}\sin\left(q_{1}^{
```

```
[26]: from sympy import symbols, diff, zeros
      import numpy as np
      # Define symbolic variables
      q = symbols('q:{}'.format(LENGTH), real=True)
      qdd = symbols('qdd:{}'.format(LENGTH), real=True)
      christoffel = []
      for i in range(LENGTH):
           temp = Matrix.zeros(4,4)
           christoffel.append(temp)
      # Calculate Christoffel symbols
      for k in range(LENGTH):
           for i in range(LENGTH):
               for j in range(LENGTH):
                    curr_matrix = christoffel[i]
                    \operatorname{curr\_matrix}[j,k] = 0.5 * (\operatorname{diff}(D[k, j], q[i]) + \operatorname{diff}(D[k, i], q[j])_{\sqcup}
       → diff(D[i, j], q[k]))
      # Display the resulting Christoffel symbols
      print("Christoffel Symbols:")
      display(christoffel)
```

Christoffel Symbols:

```
[Matrix([
    Ш
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             ш
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             Ш
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             ш
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             ш
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             Ш
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             Ш
                                            0, -0.5*m1*(-2*a1*sin(q1) - 2*c1x*sin(q1) - 2*c1y*cos(q1))*(a1*cos(q1) + 0.5*m1*(-2*a1*sin(q1) - 2*c1y*cos(q1))*(a1*cos(q1) - 2*c1y*cos(q1) + 0.5*m1*(-2*a1*sin(q1) - 2*c1y*cos(q1))*(a1*cos(q1) - 2*c1y*cos(q1) + 0.5*m1*(-2*a1*sin(q1) - 2*c1y*cos(q1))*(a1*cos(q1) - 2*c1y*cos(q1) + 0.5*m1*(-2*a1*cos(q1) - 2*c1y
       \neg c1x*cos(q1) - c1y*sin(q1)) - 0.5*m1*(-a1*sin(q1) - c1x*sin(q1) - c1x*sin(q1)
       \neg c1y*cos(q1))*(-2*a1*cos(q1) - 2*c1x*cos(q1) + 2*c1y*sin(q1)) - 0.
       _{5*m2*(-2*a1*sin(q1) - 2*a2*sin(q1)*cos(q2) - 2*a2*sin(q2)*cos(q1) +_{\sqcup}
       42*c2x*(-\sin(q1)*\cos(q2) - \sin(q2)*\cos(q1)) + 2*c2y*(\sin(q1)*\sin(q2) - u)

\varphi\cos(q1)*\cos(q2)))*(a1*\cos(q1) - a2*\sin(q1)*\sin(q2) + a2*\cos(q1)*\cos(q2) + u

       4c2x*(-\sin(q1)*\sin(q2) + \cos(q1)*\cos(q2)) + c2y*(-\sin(q1)*\cos(q2) - 1)
       \Rightarrowsin(q2)*cos(q1))) - 0.5*m2*(-a1*sin(q1) - a2*sin(q1)*cos(q2) -
       \Rightarrow a2*sin(q2)*cos(q1) + c2x*(-sin(q1)*cos(q2) - sin(q2)*cos(q1)) + c2x*(-sin(q1)*cos(q2) - sin(q2)*cos(q2)) + c2x*(-sin(q1)*cos(q2) - sin(q2)*cos(q2)) + c2x*(-sin(q1)*cos(q2) - sin(q2)*cos(q2)) + c2x*(-sin(q1)*cos(q2)*cos(q2) - sin(q2)*cos(q2) - sin(q2)*cos(q2) + c2x*(-sin(q1)*cos(q2)*cos(q2) - sin(q2)*cos(q2) + c2x*(-sin(q1)*cos(q2)*cos(q2) + c2x*(-sin(q1)*cos(q2)*cos(q2) + c2x*(-sin(q1)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q
       \neg c2y*(\sin(q1)*\sin(q2) - \cos(q1)*\cos(q2)))*(-2*a1*\cos(q1) + 2*a2*\sin(q1)*\sin(q2)
       _{-} 2*a2*cos(q1)*cos(q2) + 2*c2x*(sin(q1)*sin(q2) - cos(q1)*cos(q2)) +_{\cup}
       42*c2y*(sin(q1)*cos(q2) + sin(q2)*cos(q1))), 0, 0],
      [0.5*m1*(-2*a1*sin(q1) - 2*c1x*sin(q1) - 2*c1y*cos(q1))*(a1*cos(q1) + (2*a1*sin(q1) - 2*c1y*cos(q1))*(a1*cos(q1) - 2*c1y*cos(q1) - 2*c1y
       \neg c1x*cos(q1) - c1y*sin(q1)) + 0.5*m1*(-a1*sin(q1) - c1x*sin(q1) - c1x
       c1y*cos(q1))*(-2*a1*cos(q1) - 2*c1x*cos(q1) + 2*c1y*sin(q1)) + 0.
       _{5*m2*(-2*a1*sin(q1) - 2*a2*sin(q1)*cos(q2) - 2*a2*sin(q2)*cos(q1) +_{\sqcup}
       42*c2x*(-\sin(q1)*\cos(q2) - \sin(q2)*\cos(q1)) + 2*c2y*(\sin(q1)*\sin(q2) - u)
       (q1)*\cos(q1)*\cos(q2)) (a1*cos(q1) - a2*sin(q1)*sin(q2) + a2*cos(q1)*cos(q2) +
       c2x*(-\sin(q1)*\sin(q2) + \cos(q1)*\cos(q2)) + c2y*(-\sin(q1)*\cos(q2) - c2x*(-\sin(q1)*\cos(q2)) + c2y*(-\sin(q1)*\cos(q2)) + c2y*(-\sin(q1)*\cos(q1)*\cos(q2)) + c2y*(-\sin(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c2y*(-c
       \Rightarrow \sin(q2)*\cos(q1))) + 0.5*m2*(-a1*\sin(q1) - a2*\sin(q1)*\cos(q2) - \Box
       \Rightarrowa2*sin(q2)*cos(q1) + c2x*(-sin(q1)*cos(q2) - sin(q2)*cos(q1)) +
       \neg c2y*(\sin(q1)*\sin(q2) - \cos(q1)*\cos(q2)))*(-2*a1*\cos(q1) + 2*a2*\sin(q1)*\sin(q2)
       -2*a2*cos(q1)*cos(q2) + 2*c2x*(sin(q1)*sin(q2) - cos(q1)*cos(q2)) + (1)
       42*c2y*(sin(q1)*cos(q2) + sin(q2)*cos(q1))),
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             ш
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             ш
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             Ш
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             Ш
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             Ш
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             Ш
                                                                                                                                                                                                                                                                                                                                                                            0, 0, 0],
```

[ш
\hookrightarrow			ш
\hookrightarrow			ш
\hookrightarrow			
\hookrightarrow			ш
\hookrightarrow	Ο,		ш
\hookrightarrow			ш
\hookrightarrow			Ц
\hookrightarrow			ш
\hookrightarrow			Ц
\hookrightarrow			ш
\hookrightarrow		0, 0, 0],	
[u
\hookrightarrow			Ц
\hookrightarrow			Ц
\hookrightarrow			Ц
\hookrightarrow			ш
\hookrightarrow			ш
\hookrightarrow			Ц
\hookrightarrow			П
\hookrightarrow			П
\hookrightarrow			П
\hookrightarrow	Ο,		ш
\hookrightarrow			ш
\hookrightarrow			ш
\hookrightarrow			Ц
\hookrightarrow		0 0 011)	Ц
\hookrightarrow		0, 0, 0]]),	
Matrix([

```
Ш
                                                                                                                                                                                                                                                                                                                                                                                                                 Ш
                                                                                                                                                                                                                                                                                                                                                                                                                  Ш
                                                                                                                                                                                                                                                                                                                                                                                                   0.
 5*m1*(-2*a1*sin(q1) - 2*c1x*sin(q1) - 2*c1y*cos(q1))*(a1*cos(q1) + c1x*cos(q1)_{0}
 \rightarrow c1y*sin(q1)) + 0.5*m1*(-a1*sin(q1) - c1x*sin(q1) -
 _{\circ}c1y*cos(q1))*(-2*a1*cos(q1) - 2*c1x*cos(q1) + 2*c1y*sin(q1)) + 0.
 _{\downarrow}5*m2*(-2*a1*sin(q1) - 2*a2*sin(q1)*cos(q2) - 2*a2*sin(q2)*cos(q1) +_{\cup}
 _{-2}*c2x*(-sin(q1)*cos(q2) - sin(q2)*cos(q1)) + 2*c2y*(sin(q1)*sin(q2) -_{-1}
 _{\circ}\cos(q1)*\cos(q2)))*(a1*\cos(q1) - a2*\sin(q1)*\sin(q2) + a2*\cos(q1)*\cos(q2) +_{\cup}
 _{\neg}c2x*(-\sin(q1)*\sin(q2) + \cos(q1)*\cos(q2)) + c2y*(-\sin(q1)*\cos(q2) -_{\Box}
 -\sin(q2)*\cos(q1))) + 0.5*m2*(-a1*\sin(q1) - a2*\sin(q1)*\cos(q2) - ___
 \Rightarrowa2*sin(q2)*cos(q1) + c2x*(-sin(q1)*cos(q2) - sin(q2)*cos(q1)) +
 _{\Rightarrow}c2y*(sin(q1)*sin(q2) - cos(q1)*cos(q2)))*(-2*a1*cos(q1) + 2*a2*sin(q1)*sin(q2)_\(\)
 _{-} 2*a2*cos(q1)*cos(q2) + 2*c2x*(sin(q1)*sin(q2) - cos(q1)*cos(q2)) +_{-}
 42*c2y*(\sin(q1)*\cos(q2) + \sin(q2)*\cos(q1))),
                                                                                                                                                                                                                                                                                                                                                                                                                  Ш
                                                                                                                                                                                                                                                                                                                                                                                                                  Ш
                                                                                                                                                                                                                                                                                                                                                                                                                  ш
                                                                                                                                                                                                                                                                                                                                                                                                                  Ш
                                                                                                                                                                                                                                                                                                                                                                                                                  Ш
                                                                                                                                                                                                                                                                                                                                                                                                                  Ш
                                                                                                                                                                                                                                                                                                                                                                                                                  Ш
                                                                                                                                                                        0, 0, 0],
[1.0*m2*(-a2*sin(q1)*sin(q2) + a2*cos(q1)*cos(q2) + c2x*(-sin(q1)*sin(q2) +_{\cup}
 -\cos(q1)*\cos(q2)) + c2y*(-\sin(q1)*\cos(q2) - \sin(q2)*\cos(q1)))*(-a1*\sin(q1) - \cos(q1))
 _{\neg}a2*sin(q1)*cos(q2) - a2*sin(q2)*cos(q1) + c2x*(-sin(q1)*cos(q2) -_{\square}
 \Rightarrowsin(q2)*cos(q1)) + c2y*(sin(q1)*sin(q2) - cos(q1)*cos(q2))) + 1.
 _{\circ}0*m2*(a2*sin(q1)*sin(q2) - a2*cos(q1)*cos(q2) + c2x*(sin(q1)*sin(q2) - _{\sqcup}
 _{-cos}(q1)*cos(q2)) + c2y*(sin(q1)*cos(q2) + sin(q2)*cos(q1)))*(-a1*sin(q1) -_{-1})
 \Rightarrowa2*sin(q1)*cos(q2) - a2*sin(q2)*cos(q1) + c2x*(-sin(q1)*cos(q2) -
 \Rightarrow \sin(q^2) * \cos(q^1) + c^2y * (\sin(q^1) * \sin(q^2) - \cos(q^1) * \cos(q^2))) + 1.
 _{\hookrightarrow}0*m2*(-a2*sin(q1)*cos(q2) - a2*sin(q2)*cos(q1) + c2x*(-sin(q1)*cos(q2) -_{\sqcup})*(-a2*sin(q1)*cos(q2) -_{\sqcup})*(-a2*sin(q1)*cos(q2) -_{\sqcup})*(-a2*sin(q1)*cos(q2) -_{\sqcup})*(-a2*sin(q1)*cos(q2) -_{\sqcup})*(-a2*sin(q1)*cos(q2) -_{\sqcup})*(-a2*sin(q1)*cos(q2) -_{\sqcup})*(-a2*sin(q1)*cos(q2) -_{\sqcup})*(-a2*sin(q1)*cos(q2) -_{\sqcup})*(-a2*sin(q1)*cos(q2) -_{\sqcup})*(-a2*sin(q2)*cos(q2) -_{\sqcup})*(-
 _{\circ}\sin(q2)*\cos(q1)) + c2y*(\sin(q1)*\sin(q2) - \cos(q1)*\cos(q2)))*(-a1*\cos(q1) +_{\cup}
 _{-a}2*\sin(q1)*\sin(q2) - a2*\cos(q1)*\cos(q2) + c2x*(\sin(q1)*\sin(q2) -_{-1}

\varphi\cos(q_1)*\cos(q_2) + c_2y*(\sin(q_1)*\cos(q_2) + \sin(q_2)*\cos(q_1)) + 1.

 40*m2*(-a2*sin(q1)*cos(q2) - a2*sin(q2)*cos(q1) + c2x*(-sin(q1)*cos(q2) - a2*sin(q2)*cos(q2) - a2*sin(q2)*cos(q2) - a2*sin(q2)*cos(q2) + c2x*(-sin(q2)*cos(q2) + c2x*(-sin(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)
 _{\circ}\sin(q2)*\cos(q1)) + c2y*(\sin(q1)*\sin(q2) - \cos(q1)*\cos(q2)))*(a1*\cos(q1) -_{\cup}
 \Rightarrowa2*sin(q1)*sin(q2) + a2*cos(q1)*cos(q2) + c2x*(-sin(q1)*sin(q2) +

\varphi\cos(q1)*\cos(q2)) + c2y*(-\sin(q1)*\cos(q2) - \sin(q2)*\cos(q1))), 0.

 _{4}5*m2*(-a2*sin(q1)*sin(q2) + a2*cos(q1)*cos(q2) + c2x*(-sin(q1)*sin(q2) + (-a2*sin(q1)*sin(q2) + (-a2*sin(q2)*sin(q2) + (-a2*sin(q2)

\downarrow \cos(q1) * \cos(q2)) + c2y*(-\sin(q1) * \cos(q2) - 

 \neg \sin(q2)*\cos(q1)))*(-2*a2*\sin(q1)*\cos(q2) - 2*a2*\sin(q2)*\cos(q1) +_{\square}
 _{-2}*c2x*(-sin(q1)*cos(q2) - sin(q2)*cos(q1)) + 2*c2y*(sin(q1)*sin(q2) -_{-1}
 _{\circ}cos(q1)*cos(q2))) + 0.5*m2*(2*a2*sin(q1)*sin(q2) - 2*a2*cos(q1)*cos(q2) +_{\cup}
 42 \times 2 \times (\sin(q1) \times \sin(q2) - \cos(q1) \times \cos(q2)) + 2 \times 2 \times (\sin(q1) \times \cos(q2) + 2 \times 2 \times (\sin(q1) \times \cos(q2)) + 2 \times (\sin(q1) \times \cos(q1) \times \cos(q2)) + 2 \times (\sin(q1) \times \cos(q1) \times (\cos(q1) \times \cos(q1)) + 2 \times (\cos(q1) \times \cos(q1) \times (\cos(q1) \times \cos(q1)) + 2 \times (\cos(q1) \times \cos(q1) \times (\cos(q1) \times \cos(q1)) + 2 \times (\cos(q1) \times \cos(q1) \times (\cos(q1) \times (\cos(q1) \times \cos(q1)) + 2 \times (\cos(q1) \times (\cos(q1) \times \cos(q1) \times (\cos(q1) \times 
 \Rightarrowsin(q2)*cos(q1)))*(-a2*sin(q1)*cos(q2) - a2*sin(q2)*cos(q1) +
 _{\neg}c2x*(-\sin(q1)*\cos(q2) - \sin(q2)*\cos(q1)) + c2y*(\sin(q1)*\sin(q2) -_{\Box}

\varphi\cos(q1)*\cos(q2)), 0, 0],
```

```
[
                                                                                                           Ш
                                                                                                           Ш
                                                                                                           П
                                                                                                           Ш
        \hookrightarrow
        ⇔0,
                                                                                                           Ш
                                                                                                           Ш
                                                                                                 0, 0, 0],
                                                                                                           Ш
                                                                                                           П
                                                                                                           Ш
                                                                                                           Ш
        \hookrightarrow
        ⇔0,
                                                                                                           Ш
                                                                                                 0, 0,
        →0]])]
[27]: from sympy import zeros, symbols
       # Define symbolic variables
```

```
qd = symbols('qd:{}'.format(LENGTH), real=True)

# Initialize a square matrix for the Coriolis matrix
C = zeros(LENGTH, LENGTH)

# Calculate the Coriolis matrix
for k in range(LENGTH):
    for j in range(LENGTH):
        temp = 0
        for i in range(LENGTH):
            temp_christoffel = christoffel[i]
            temp += temp_christoffel[j, k] * qd[i]
        C[j, k] = temp

# Display the resulting Coriolis matrix
print("Coriolis Matrix:")
print(C)
display(C.shape)
```

```
Coriolis Matrix:
```

```
Matrix([[qd1*(0.5*m1*(-2*a1*sin(q1) - 2*c1x*sin(q1) - 2*c1y*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))*(a1*cos(q1))
+ c1x*cos(q1) - c1y*sin(q1)) + 0.5*m1*(-a1*sin(q1) - c1x*sin(q1) -
c1y*cos(q1))*(-2*a1*cos(q1) - 2*c1x*cos(q1) + 2*c1y*sin(q1)) +
0.5*m2*(-2*a1*sin(q1) - 2*a2*sin(q1)*cos(q2) - 2*a2*sin(q2)*cos(q1) +
2*c2x*(-\sin(q1)*\cos(q2) - \sin(q2)*\cos(q1)) + 2*c2y*(\sin(q1)*\sin(q2) - \cos(q2))
\cos(q1)*\cos(q2)))*(a1*\cos(q1) - a2*\sin(q1)*\sin(q2) + a2*\cos(q1)*\cos(q2) +
c2x*(-\sin(q1)*\sin(q2) + \cos(q1)*\cos(q2)) + c2y*(-\sin(q1)*\cos(q2) -
\sin(q2)*\cos(q1))) + 0.5*m2*(-a1*\sin(q1) - a2*\sin(q1)*\cos(q2) -
a2*sin(q2)*cos(q1) + c2x*(-sin(q1)*cos(q2) - sin(q2)*cos(q1)) +
c2y*(sin(q1)*sin(q2) - cos(q1)*cos(q2)))*(-2*a1*cos(q1) + 2*a2*sin(q1)*sin(q2) -
2*a2*cos(q1)*cos(q2) + 2*c2x*(sin(q1)*sin(q2) - cos(q1)*cos(q2)) +
2*c2y*(sin(q1)*cos(q2) + sin(q2)*cos(q1)))), qd0*(-0.5*m1*(-2*a1*sin(q1) - cos(q2))))
2*c1x*sin(q1) - 2*c1y*cos(q1))*(a1*cos(q1) + c1x*cos(q1) - c1y*sin(q1)) -
0.5*m1*(-a1*sin(q1) - c1x*sin(q1) - c1y*cos(q1))*(-2*a1*cos(q1) - 2*c1x*cos(q1)
+ 2*c1y*sin(q1)) - 0.5*m2*(-2*a1*sin(q1) - 2*a2*sin(q1)*cos(q2) -
2*a2*sin(q2)*cos(q1) + 2*c2x*(-sin(q1)*cos(q2) - sin(q2)*cos(q1)) +
2*c2y*(sin(q1)*sin(q2) - cos(q1)*cos(q2)))*(a1*cos(q1) - a2*sin(q1)*sin(q2) +
a2*cos(q1)*cos(q2) + c2x*(-sin(q1)*sin(q2) + cos(q1)*cos(q2)) +
c2y*(-sin(q1)*cos(q2) - sin(q2)*cos(q1))) - 0.5*m2*(-a1*sin(q1) - c2y*(-sin(q1)*cos(q2) - sin(q2)*cos(q1))) - 0.5*m2*(-a1*sin(q1) - c2y*(-a1*sin(q1) - c2y*(-a1*sin
a2*sin(q1)*cos(q2) - a2*sin(q2)*cos(q1) + c2x*(-sin(q1)*cos(q2) -
\sin(q_2)*\cos(q_1)) + c_2y*(\sin(q_1)*\sin(q_2) - \cos(q_1)*\cos(q_2)))*(-2*a_1*\cos(q_1) +
2*a2*sin(q1)*sin(q2) - 2*a2*cos(q1)*cos(q2) + 2*c2x*(sin(q1)*sin(q2) -
cos(q1)*cos(q2)) + 2*c2y*(sin(q1)*cos(q2) + sin(q2)*cos(q1))))],
[qd0*(0.5*m1*(-2*a1*sin(q1) - 2*c1x*sin(q1) - 2*c1y*cos(q1))*(a1*cos(q1) + (a1*cos(q1) + (a1*cos(q
c1x*cos(q1) - c1y*sin(q1)) + 0.5*m1*(-a1*sin(q1) - c1x*sin(q1) -
c1y*cos(q1))*(-2*a1*cos(q1) - 2*c1x*cos(q1) + 2*c1y*sin(q1)) +
0.5*m2*(-2*a1*sin(q1) - 2*a2*sin(q1)*cos(q2) - 2*a2*sin(q2)*cos(q1) +
2*c2x*(-\sin(q1)*\cos(q2) - \sin(q2)*\cos(q1)) + 2*c2y*(\sin(q1)*\sin(q2) - \cos(q2))
```

```
c2x*(-sin(q1)*sin(q2) + cos(q1)*cos(q2)) + c2y*(-sin(q1)*cos(q2) - c2x*(-sin(q1)*sin(q2)) + c2
                            \sin(q2)*\cos(q1))) + 0.5*m2*(-a1*\sin(q1) - a2*\sin(q1)*\cos(q2) -
                            a2*sin(q2)*cos(q1) + c2x*(-sin(q1)*cos(q2) - sin(q2)*cos(q1)) +
                             c2y*(sin(q1)*sin(q2) - cos(q1)*cos(q2)))*(-2*a1*cos(q1) + 2*a2*sin(q1)*sin(q2) - cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos
                            2*a2*cos(q1)*cos(q2) + 2*c2x*(sin(q1)*sin(q2) - cos(q1)*cos(q2)) +
                            a2*cos(q1)*cos(q2) + c2x*(-sin(q1)*sin(q2) + cos(q1)*cos(q2)) +
                            c2y*(-\sin(q1)*\cos(q2) - \sin(q2)*\cos(q1)))*(-a1*\sin(q1) - a2*\sin(q1)*\cos(q2) - \sin(q2)*\cos(q2)
                            a2*sin(q2)*cos(q1) + c2x*(-sin(q1)*cos(q2) - sin(q2)*cos(q1)) +
                            c2y*(sin(q1)*sin(q2) - cos(q1)*cos(q2))) + 1.0*m2*(a2*sin(q1)*sin(q2) -
                            a2*cos(q1)*cos(q2) + c2x*(sin(q1)*sin(q2) - cos(q1)*cos(q2)) +
                            c2y*(sin(q1)*cos(q2) + sin(q2)*cos(q1)))*(-a1*sin(q1) - a2*sin(q1)*cos(q2) - a2*sin(q1)*cos(q2)*cos(q2) - a2*sin(q1)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*
                            a2*sin(q2)*cos(q1) + c2x*(-sin(q1)*cos(q2) - sin(q2)*cos(q1)) +
                             c2y*(\sin(q1)*\sin(q2) - \cos(q1)*\cos(q2))) + 1.0*m2*(-a2*\sin(q1)*\cos(q2) - (a2*\sin(q1)*\cos(q2)) + (a2*\cos(q1)*\cos(q2)) + (a2*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)*\cos(q1)
                            a2*sin(q2)*cos(q1) + c2x*(-sin(q1)*cos(q2) - sin(q2)*cos(q1)) +
                            c2y*(sin(q1)*sin(q2) - cos(q1)*cos(q2)))*(-a1*cos(q1) + a2*sin(q1)*sin(q2) - cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*c
                            a2*cos(q1)*cos(q2) + c2x*(sin(q1)*sin(q2) - cos(q1)*cos(q2)) +
                            c2y*(sin(q1)*cos(q2) + sin(q2)*cos(q1))) + 1.0*m2*(-a2*sin(q1)*cos(q2) -
                            a2*sin(q2)*cos(q1) + c2x*(-sin(q1)*cos(q2) - sin(q2)*cos(q1)) +
                            c2y*(sin(q1)*sin(q2) - cos(q1)*cos(q2)))*(a1*cos(q1) - a2*sin(q1)*sin(q2) +
                            a2*cos(q1)*cos(q2) + c2x*(-sin(q1)*sin(q2) + cos(q1)*cos(q2)) +
                            c2y*(-sin(q1)*cos(q2) - sin(q2)*cos(q1)))), qd1*(0.5*m2*(-a2*sin(q1)*sin(q2) + c2*sin(q1)*sin(q2) + c2*sin(q1)*sin(q2)*cos(q1))))
                            a2*cos(q1)*cos(q2) + c2x*(-sin(q1)*sin(q2) + cos(q1)*cos(q2)) +
                            c2y*(-sin(q1)*cos(q2) - sin(q2)*cos(q1)))*(-2*a2*sin(q1)*cos(q2) - sin(q2)*cos(q2))
                            2*a2*sin(q2)*cos(q1) + 2*c2x*(-sin(q1)*cos(q2) - sin(q2)*cos(q1)) +
                            2*c2y*(sin(q1)*sin(q2) - cos(q1)*cos(q2))) + 0.5*m2*(2*a2*sin(q1)*sin(q2) - cos(q1)*cos(q2)))
                            2*a2*cos(q1)*cos(q2) + 2*c2x*(sin(q1)*sin(q2) - cos(q1)*cos(q2)) +
                            2*c2y*(sin(q1)*cos(q2) + sin(q2)*cos(q1)))*(-a2*sin(q1)*cos(q2) -
                             a2*sin(q2)*cos(q1) + c2x*(-sin(q1)*cos(q2) - sin(q2)*cos(q1)) +
                             c2y*(sin(q1)*sin(q2) - cos(q1)*cos(q2))))]])
                             (2, 2)
[28]: from sympy import diff, symbols, zeros, simplify
                                # Calculate the gravitational terms
                                G = zeros(LENGTH, 1)
                                for k in range(LENGTH):
                                                     G[k] = diff(PE, q[k])
                                qdd_matrix = Matrix([qdd]).T
                                qd_matrix = Matrix([qd]).T
                                # Calculate the left-hand side of the equations of motion
                                eom_lhs = D * qdd_matrix + C * qd_matrix + G
                                simplified_matrix = eom_lhs.applyfunc(simplify)
```

 $\cos(q1)*\cos(q2))*(a1*\cos(q1) - a2*\sin(q1)*\sin(q2) + a2*\cos(q1)*\cos(q2) +$

```
# Display the resulting gravitational terms and equations of motion
      print("\nEquations of Motion (eom_lhs):")
      display(simplified_matrix)
Gravitational Terms (G):
Matrix([[0], [0]])
Equations of Motion (eom_lhs):
Matrix([[qd0*qd1*(-0.5*m1*(-2*a1*sin(q1) - 2*c1x*sin(q1) - 2
2*c1y*cos(q1))*(a1*cos(q1) + c1x*cos(q1) - c1y*sin(q1)) - 0.5*m1*(-a1*sin(q1) - c1y*sin(q1))
c1x*sin(q1) - c1y*cos(q1))*(-2*a1*cos(q1) - 2*c1x*cos(q1) + 2*c1y*sin(q1)) -
0.5*m2*(-2*a1*sin(q1) - 2*a2*sin(q1)*cos(q2) - 2*a2*sin(q2)*cos(q1) +
2*c2x*(-\sin(q1)*\cos(q2) - \sin(q2)*\cos(q1)) + 2*c2y*(\sin(q1)*\sin(q2) - \cos(q2))
\cos(q1)*\cos(q2))*(a1*\cos(q1) - a2*\sin(q1)*\sin(q2) + a2*\cos(q1)*\cos(q2) +
c2x*(-sin(q1)*sin(q2) + cos(q1)*cos(q2)) + c2y*(-sin(q1)*cos(q2) -
\sin(q2)*\cos(q1))) - 0.5*m2*(-a1*\sin(q1) - a2*\sin(q1)*\cos(q2) -
a2*sin(q2)*cos(q1) + c2x*(-sin(q1)*cos(q2) - sin(q2)*cos(q1)) +
c2y*(sin(q1)*sin(q2) - cos(q1)*cos(q2)))*(-2*a1*cos(q1) + 2*a2*sin(q1)*sin(q2) - cos(q1)*cos(q1)*cos(q2)))*(-2*a1*cos(q1) + 2*a2*sin(q1)*sin(q2) - cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos
2*a2*cos(q1)*cos(q2) + 2*c2x*(sin(q1)*sin(q2) - cos(q1)*cos(q2)) +
2*c2y*(\sin(q1)*\cos(q2) + \sin(q2)*\cos(q1)))) + qd0*qd1*(0.5*m1*(-2*a1*\sin(q1) - a2*a2*a2*q1)))) + qd0*qd1*(0.5*m1*(-2*a1*a2*q1) + a2*a2*q2*q1)))) + qd0*qd1*(0.5*m1*(-2*a1*a2*q1) + a2*q2*q1*(0.5*m1*(-2*a1*a2*q1)))) + qd0*qd1*(0.5*m1*(-2*a1*a2*q1)))) + qd0*qd1*(0.5*m1*(-2*a1*a2*q1))) + qd0*qd1*(0.5*m1*(-2*a1*a2*q1)))) + qd0*qd1*(0.5*m1*(-2*a1*a2*q1))) + qd0*qd1*(0.5*m1*(-2*a1*a2*q1)))) + qd0*qd1*(0.5*m1*(-2*a1*a2*q1)))) + qd0*qd1*(0.5*m1*(-2*a1*a2*q1)))) + qd0*qd1*(0.5*m1*(-2*a1*q1)))) + qd0*qd1*(0.5*m1*(-2*a1*q1)))) + qd0*qd1*(0.5*m1*(-2*a1*q1)))) + qd0*qd1*(0.5*m1*(-2*a1*q1)))) + qd0*qd1*(0.5*m1*(-2*a1*q1))) + qd0*qd1*(0.5*m1*(-2*a1*q1)))) + qd0*qd1*(0.5*m1*(-2*a1*q1)))) + qd0*qd1*(0.5*m1*(-2*a1*q1)))) + qd0*qd1*(0.5*m1*(-2*a1*q1)))) + qd0*qd1*(0.5*m1*(-2*a1*q1)))) + qd0*qd1*(0.5*m1*(-2*a1*q1))) + qd0*qd1*(0.5*m1*(-2*a1*q1))) + qd0*qd1*(0.5*m1*(-2*a1*q1))) + qd0*qd1*(0.5*m1*(-2*a1*q1))) + qd0*qd1*(0.5*m1*(-2*a1*q1)) + qd0*q1*(0.5*m1*(-2*a1*q1)) + qd0*q1*(0.5*m1*(-2*a1*q1)) + qd0*q1*(0.5*m1*(-2*a1*q1)) + qd0*q1*(0.5*m1*(-2*a1*q1)) + qd0*q1*(0.5*m1*(-2*a1*q1)) + qd0*q1*(0.5*m1*(0.5*m1*(-2*q1)) + qd0*q1*(0.5*m1*(-2*a1*q1)) + qd0*q1*(0.5*m1*
2*c1x*sin(q1) - 2*c1y*cos(q1))*(a1*cos(q1) + c1x*cos(q1) - c1y*sin(q1)) +
0.5*m1*(-a1*sin(q1) - c1x*sin(q1) - c1y*cos(q1))*(-2*a1*cos(q1) - 2*c1x*cos(q1))
+ 2*c1y*sin(q1)) + 0.5*m2*(-2*a1*sin(q1) - 2*a2*sin(q1)*cos(q2) -
2*a2*sin(q2)*cos(q1) + 2*c2x*(-sin(q1)*cos(q2) - sin(q2)*cos(q1)) +
2*c2y*(sin(q1)*sin(q2) - cos(q1)*cos(q2)))*(a1*cos(q1) - a2*sin(q1)*sin(q2) + a2*sin(q2)*sin(q2) + a2*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*sin(q2)*si
a2*cos(q1)*cos(q2) + c2x*(-sin(q1)*sin(q2) + cos(q1)*cos(q2)) +
c2y*(-sin(q1)*cos(q2) - sin(q2)*cos(q1))) + 0.5*m2*(-a1*sin(q1) - c2y*(-sin(q1)*cos(q2) - sin(q2)*cos(q1))) + 0.5*m2*(-a1*sin(q1) - c2y*(-a1*sin(q1) - c2y*(-a1*sin
a2*sin(q1)*cos(q2) - a2*sin(q2)*cos(q1) + c2x*(-sin(q1)*cos(q2) -
\sin(q_2)*\cos(q_1)) + c_2y*(\sin(q_1)*\sin(q_2) - \cos(q_1)*\cos(q_2)))*(-2*a_1*\cos(q_1) + c_2y*(a_1))*(-2*a_1*a_2))*(-2*a_1*a_2)*(a_1)*(a_1)*(a_2)*(a_2)*(a_2)*(a_1)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(a_2)*(
2*a2*sin(q1)*sin(q2) - 2*a2*cos(q1)*cos(q2) + 2*c2x*(sin(q1)*sin(q2) - 2*a2*sin(q1)*sin(q2) - 2*a2*sin(q1)*sin(q1)*sin(q2) - 2*a2*sin(q1)*sin(q1)*sin(q2) - 2*a2*sin(q1)*sin(q2) - 2*a2*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q2)*sin(q1)*sin(q1)*sin(q2)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q2)*sin(q1)*sin(q1)*sin(q2)*sin(q1)*sin(q1)*sin(q2)*sin(q1)*sin(q2)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1)*sin(q1
\cos(q1)*\cos(q2)) + 2*c2y*(\sin(q1)*\cos(q2) + \sin(q2)*\cos(q1)))) + qdd0*(Izz1 + cos(q2)) 
c1x*cos(q1) - c1y*sin(q1))**2 + m2*(-a1*sin(q1) - a2*sin(q1)*cos(q2) -
a2*sin(q2)*cos(q1) + c2x*(-sin(q1)*cos(q2) - sin(q2)*cos(q1)) +
c2y*(sin(q1)*sin(q2) - cos(q1)*cos(q2)))**2 + m2*(a1*cos(q1) - cos(q1)*cos(q1)*cos(q2)))**2 + m2*(a1*cos(q1) - cos(q1)*cos(q2)))**2 + m2*(a1*cos(q1) - cos(q1)*cos(q2)))**2 + m2*(a1*cos(q1) - cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q
a2*sin(q1)*sin(q2) + a2*cos(q1)*cos(q2) + c2x*(-sin(q1)*sin(q2) +
\cos(q1)*\cos(q2)) + c2y*(-\sin(q1)*\cos(q2) - \sin(q2)*\cos(q1)))**2) + qdd1*(Izz2 + qdd1)*(Izz2 + qd1)*(Izz2 
m2*(-a2*sin(q1)*sin(q2) + a2*cos(q1)*cos(q2) + c2x*(-sin(q1)*sin(q2) +
\cos(q1)*\cos(q2)) \; + \; c2y*(-\sin(q1)*\cos(q2) \; - \; \sin(q2)*\cos(q1)))*(a1*\cos(q1) \; - \; \cos(q1)*\cos(q1)) + \; c2y*(-\sin(q1)*\cos(q2) \; - \; \sin(q2)*\cos(q1))) + \; c2y*(-\sin(q1)*\cos(q2) \; - \; \sin(q2)*\cos(q1))) + \; c2y*(-\sin(q1)*\cos(q2) \; - \; \sin(q2)*\cos(q1))) + \; c2y*(-\sin(q1)*\cos(q2) \; - \; \cos(q2) \; - \; \cos(q2)) + \; c3y*(-\sin(q1)*\cos(q2) \; - \; \cos(q2)) + \; c3y*(-\sin(q1)*\cos(q2) \; - \; \cos(q2)) + \; c3y*(-\sin(q1)*\cos(q2)) + \; c3y*(-\sin(q1)*\cos(q1)) + \; c3y*(-\sin(q1)*\cos(q1)*(-\cos(q1)*\cos(q1)*(-\cos(q1)*\cos(q1)) + \; c3y*(-\sin(q1)*\cos(q1)*(-\cos(q1)*\cos(q1)*(-\cos(q1)*(-\cos(q1)*\cos(
a2*sin(q1)*sin(q2) + a2*cos(q1)*cos(q2) + c2x*(-sin(q1)*sin(q2) +
cos(q1)*cos(q2)) + c2y*(-sin(q1)*cos(q2) - sin(q2)*cos(q1))) +
m2*(-a2*sin(q1)*cos(q2) - a2*sin(q2)*cos(q1) + c2x*(-sin(q1)*cos(q2) - a2*sin(q2)*cos(q2) + c2x*(-sin(q2)*cos(q2) + c2x*(-sin(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)
\sin(q2)*\cos(q1)) + c2y*(\sin(q1)*\sin(q2) - \cos(q1)*\cos(q2)))*(-a1*\sin(q1) - \cos(q1)*\cos(q2))
a2*sin(q1)*cos(q2) - a2*sin(q2)*cos(q1) + c2x*(-sin(q1)*cos(q2) -
\sin(q2)*\cos(q1)) + c2y*(\sin(q1)*\sin(q2) - \cos(q1)*\cos(q2))))],
  [qd0*(qd0*(0.5*m1*(-2*a1*sin(q1) - 2*c1x*sin(q1) - 2*c1y*cos(q1))*(a1*cos(q1) + (-2*a1*sin(q1) - 2*c1y*cos(q1))*(a1*cos(q1) + (-2*a1*sin(q1) - 2*c1y*cos(q1))*(a1*cos(q1) + (-2*a1*sin(q1) - 2*c1y*cos(q1))*(a1*cos(q1) - (-2*a1*sin(q1) - (-2*a1*s
 c1x*cos(q1) - c1y*sin(q1)) + 0.5*m1*(-a1*sin(q1) - c1x*sin(q1) -
```

```
c1y*cos(q1))*(-2*a1*cos(q1) - 2*c1x*cos(q1) + 2*c1y*sin(q1)) +
0.5*m2*(-2*a1*sin(q1) - 2*a2*sin(q1)*cos(q2) - 2*a2*sin(q2)*cos(q1) +
2*c2x*(-\sin(q1)*\cos(q2) - \sin(q2)*\cos(q1)) + 2*c2y*(\sin(q1)*\sin(q2) - \cos(q2))
 \cos(q1)*\cos(q2)))*(a1*\cos(q1) - a2*\sin(q1)*\sin(q2) + a2*\cos(q1)*\cos(q2) +
 c2x*(-sin(q1)*sin(q2) + cos(q1)*cos(q2)) + c2y*(-sin(q1)*cos(q2) -
 \sin(q_2)*\cos(q_1))) + 0.5*m_2*(-a_1*\sin(q_1) - a_2*\sin(q_1)*\cos(q_2) - a_2*\sin(q_1)*\cos(q_2)
a2*sin(q2)*cos(q1) + c2x*(-sin(q1)*cos(q2) - sin(q2)*cos(q1)) +
c2y*(sin(q1)*sin(q2) - cos(q1)*cos(q2)))*(-2*a1*cos(q1) + 2*a2*sin(q1)*sin(q2) - cos(q1)*cos(q1)*cos(q2)))*(-2*a1*cos(q1) + 2*a2*sin(q1)*sin(q2) - cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos
2*a2*cos(q1)*cos(q2) + 2*c2x*(sin(q1)*sin(q2) - cos(q1)*cos(q2)) +
2*c2y*(\sin(q1)*\cos(q2) + \sin(q2)*\cos(q1))) + qd1*(1.0*m2*(-a2*\sin(q1)*\sin(q2) + \sin(q2)*\cos(q1)))) + qd1*(1.0*m2*(-a2*\sin(q1)*\sin(q2) + \sin(q2)*\cos(q1)))) + qd1*(1.0*m2*(-a2*\sin(q1)*\sin(q2) + \sin(q2)*\cos(q1)))) + qd1*(1.0*m2*(-a2*\sin(q1)*\sin(q2) + \cos(q2) + \sin(q2)*\cos(q3))))
a2*cos(q1)*cos(q2) + c2x*(-sin(q1)*sin(q2) + cos(q1)*cos(q2)) +
 c2y*(-sin(q1)*cos(q2) - sin(q2)*cos(q1)))*(-a1*sin(q1) - a2*sin(q1)*cos(q2) - a2*sin(q1)*cos(q2)*cos(q2) - a2*sin(q1)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)
a2*sin(q2)*cos(q1) + c2x*(-sin(q1)*cos(q2) - sin(q2)*cos(q1)) +
c2y*(sin(q1)*sin(q2) - cos(q1)*cos(q2))) + 1.0*m2*(a2*sin(q1)*sin(q2) - cos(q1)*cos(q2)))
a2*cos(q1)*cos(q2) + c2x*(sin(q1)*sin(q2) - cos(q1)*cos(q2)) +
c2y*(sin(q1)*cos(q2) + sin(q2)*cos(q1)))*(-a1*sin(q1) - a2*sin(q1)*cos(q2) - a2*sin(q2)*cos(q2) - a2*sin(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2)*cos(q2
a2*sin(q2)*cos(q1) + c2x*(-sin(q1)*cos(q2) - sin(q2)*cos(q1)) +
c2y*(sin(q1)*sin(q2) - cos(q1)*cos(q2))) + 1.0*m2*(-a2*sin(q1)*cos(q2) - cos(q1)*cos(q2)) + 1.0*m2*(-a2*sin(q1)*cos(q2)) + 1.0*m2*(-a2*sin(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*
a2*sin(q2)*cos(q1) + c2x*(-sin(q1)*cos(q2) - sin(q2)*cos(q1)) +
c2y*(sin(q1)*sin(q2) - cos(q1)*cos(q2)))*(-a1*cos(q1) + a2*sin(q1)*sin(q2) - cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*cos(q1)*c
a2*cos(q1)*cos(q2) + c2x*(sin(q1)*sin(q2) - cos(q1)*cos(q2)) +
c2y*(sin(q1)*cos(q2) + sin(q2)*cos(q1))) + 1.0*m2*(-a2*sin(q1)*cos(q2) - a2*sin(q1)*cos(q2))
a2*sin(q2)*cos(q1) + c2x*(-sin(q1)*cos(q2) - sin(q2)*cos(q1)) +
c2y*(sin(q1)*sin(q2) - cos(q1)*cos(q2)))*(a1*cos(q1) - a2*sin(q1)*sin(q2) + a2*sin(q2) + a
a2*cos(q1)*cos(q2) + c2x*(-sin(q1)*sin(q2) + cos(q1)*cos(q2)) +
c2y*(-sin(q1)*cos(q2) - sin(q2)*cos(q1))))) +
qd1**2*(0.5*m2*(-a2*sin(q1)*sin(q2) + a2*cos(q1)*cos(q2) + c2x*(-sin(q1)*sin(q2))
+ \cos(q1)*\cos(q2)) + c2y*(-\sin(q1)*\cos(q2) -
 \sin(q_2)*\cos(q_1))*(-2*a_2*\sin(q_1)*\cos(q_2) - 2*a_2*\sin(q_2)*\cos(q_1) +
2*c2x*(-\sin(q1)*\cos(q2) - \sin(q2)*\cos(q1)) + 2*c2y*(\sin(q1)*\sin(q2) - \cos(q2))
\cos(q1)*\cos(q2))) + 0.5*m2*(2*a2*\sin(q1)*\sin(q2) - 2*a2*\cos(q1)*\cos(q2) +
2*c2x*(\sin(q1)*\sin(q2) - \cos(q1)*\cos(q2)) + 2*c2y*(\sin(q1)*\cos(q2) + (\cos(q1)*\cos(q2)))
\sin(q_2)*\cos(q_1))*(-a_2*\sin(q_1)*\cos(q_2) - a_2*\sin(q_2)*\cos(q_1) +
c2x*(-sin(q1)*cos(q2) - sin(q2)*cos(q1)) + c2y*(sin(q1)*sin(q2) -
\cos(q1)*\cos(q2)))) \ + \ qdd0*(Izz2 \ + \ m2*(-a2*\sin(q1)*\sin(q2) \ + \ a2*\cos(q1)*\cos(q2) \ + \ a2*\cos(q2)*\cos(q2) \ + \ a2*\cos(q2) \ + \ 
c2x*(-sin(q1)*sin(q2) + cos(q1)*cos(q2)) + c2y*(-sin(q1)*cos(q2) - c2x*(-sin(q1)*sin(q2)) + c2
\sin(q_2)*\cos(q_1))*(a_1*\cos(q_1) - a_2*\sin(q_1)*\sin(q_2) + a_2*\cos(q_1)*\cos(q_2) +
c2x*(-sin(q1)*sin(q2) + cos(q1)*cos(q2)) + c2y*(-sin(q1)*cos(q2) -
\sin(q2)*\cos(q1))) + m2*(-a2*\sin(q1)*\cos(q2) - a2*\sin(q2)*\cos(q1) +
c2x*(-\sin(q1)*\cos(q2) - \sin(q2)*\cos(q1)) + c2y*(\sin(q1)*\sin(q2) - \cos(q2))
\cos(q1)*\cos(q2)))*(-a1*\sin(q1) - a2*\sin(q1)*\cos(q2) - a2*\sin(q2)*\cos(q1) + a2*\sin(q2)*\cos(q2) + a2*\cos(q2) + a2*\sin(q2)*\cos(q2) + a2*\cos(q2) +
c2x*(-\sin(q1)*\cos(q2) - \sin(q2)*\cos(q1)) + c2y*(\sin(q1)*\sin(q2) - \cos(q2))
\cos(q1)*\cos(q2)))) + qdd1*(Izz2 + m2*(-a2*\sin(q1)*\sin(q2) + a2*\cos(q1)*\cos(q2) + a2*\cos(q1)*\cos(q2))))
 c2x*(-sin(q1)*sin(q2) + cos(q1)*cos(q2)) + c2y*(-sin(q1)*cos(q2) - c2x*(-sin(q1)*sin(q2)) + c2
\sin(q2)*\cos(q1)))**2 + m2*(-a2*\sin(q1)*\cos(q2) - a2*\sin(q2)*\cos(q1) +
c2x*(-\sin(q1)*\cos(q2) - \sin(q2)*\cos(q1)) + c2y*(\sin(q1)*\sin(q2) - \cos(q2))
 cos(q1)*cos(q2)))**2)]])
```