

```
In [1]: import numpy as np
        from scipy.linalg import expm, logm
        from ikin_api import IKinSpace, IKinBody
```

Trajectory Generation

```
In [2]: t = 0.9

s = 0.25*t + 0.125*t**2
s_dot = 0.25 + 0.25*t
s_ddot = 0.25

x = np.cos(2*np.pi*s)
x_dot = -2*np.pi*s_dot*np.sin(2*np.pi*s)
x_ddot = -2*np.pi*s_ddot*np.sin(2*np.pi*s) - (2*np.pi)**2*s_dot**2*np.cos(2*np.pi*s)

y = np.sin(2*np.pi*s)
y_dot = 2*np.pi*s_dot*np.cos(2*np.pi*s)
y_ddot = 2*np.pi*s_ddot*np.cos(2*np.pi*s) - (2*np.pi)**2*s_dot**2*np.sin(2*np.pi*s)

z = 2*s
z_dot = 2*s_dot
z_ddot = 2*s_ddot

print('x: ', x)
print('y: ', y)
print('z: ', z)

print('x dot: ', x_dot)
print('y dot: ', y_dot)
print('z dot: ', z_dot)

print('x ddot: ', x_ddot)
print('y ddot: ', y_ddot)
print('z ddot: ', z_ddot)

x:  -0.46097437453546264
y:  0.887413447059283
z:  0.6525000000000001
x dot:  -2.6484969876793265
y dot:  -1.3757840231070713
z dot:  0.95
x ddot:  2.7120995479342964
y ddot:  -8.628570599837545
z ddot:  0.5
```

Motion Planning

```
In [3]: def is_in_collision(p_1, r_1, p_2, r_2):
        d = np.sqrt((p_1[0,0]-p_2[0,0])**2 + (p_1[1,0]-p_2[1,0])**2 + (p_1[2,0]-p_2[2,0])**2)
        return d <= r_1 + r_2
```

```
In [4]: import numpy as np

p_1 = np.array([-1.9706], [-1.2766], [-4.7440]])
p_2 = np.array([-1.3391], [-2.0459], [-1.9077]])
r_1 = 1.9808
r_2 = 1.7493

print('is in collision: ', is_in_collision(p_1, r_1, p_2, r_2))
```

is in collision: True

```
In [5]: import numpy as np

N_robot = 20.00000000
N_obstacles = 40.00000000
robot_spheres = np.array([[0.00000000, 0.00000000, 1.09776243], [0.46914896, -0.17291401, 1.09776243], [0.93829791, -0.34582803, 1.09776243], [1.40744687, -0.51874204, 1.09776243], [1.87659582, -0.68975795, 1.09776243], [2.34574478, -0.86077386, 1.09776243], [2.81489374, -0.17291401, 1.09776243], [3.28404269, -0.34582803, 1.09776243], [3.75319165, -0.51874204, 1.09776243], [4.22234061, -0.68975795, 1.09776243], [4.69148957, -0.86077386, 1.09776243], [5.16063853, -0.17291401, 1.09776243], [5.62978749, -0.34582803, 1.09776243], [6.09893645, -0.51874204, 1.09776243], [6.56808541, -0.68975795, 1.09776243], [7.03723437, -0.86077386, 1.09776243], [7.50638333, -0.17291401, 1.09776243], [7.97553229, -0.34582803, 1.09776243], [8.44468125, -0.51874204, 1.09776243], [8.91383021, -0.68975795, 1.09776243], [9.38297917, -0.86077386, 1.09776243], [9.85212813, -0.17291401, 1.09776243], [10.32127709, -0.34582803, 1.09776243], [10.79042605, -0.51874204, 1.09776243], [11.25957501, -0.68975795, 1.09776243], [11.72872397, -0.86077386, 1.09776243], [12.19787293, -0.17291401, 1.09776243], [12.66702189, -0.34582803, 1.09776243], [13.13617085, -0.51874204, 1.09776243], [13.60531981, -0.68975795, 1.09776243], [14.07446877, -0.86077386, 1.09776243], [14.54361773, -0.17291401, 1.09776243], [15.01276669, -0.34582803, 1.09776243], [15.48191565, -0.51874204, 1.09776243], [15.95106461, -0.68975795, 1.09776243], [16.42021357, -0.86077386, 1.09776243], [16.88936253, -0.17291401, 1.09776243], [17.35851149, -0.34582803, 1.09776243], [17.82766045, -0.51874204, 1.09776243], [18.29680941, -0.68975795, 1.09776243], [18.76595837, -0.86077386, 1.09776243], [19.23510733, -0.17291401, 1.09776243], [19.70425629, -0.34582803, 1.09776243], [20.17340525, -0.51874204, 1.09776243], [20.64255421, -0.68975795, 1.09776243], [21.11170317, -0.86077386, 1.09776243], [21.58085213, -0.17291401, 1.09776243], [22.05000109, -0.34582803, 1.09776243], [22.51915005, -0.51874204, 1.09776243], [22.98829901, -0.68975795, 1.09776243], [23.45744797, -0.86077386, 1.09776243], [23.92659693, -0.17291401, 1.09776243], [24.39574589, -0.34582803, 1.09776243], [24.86489485, -0.51874204, 1.09776243], [25.33404381, -0.68975795, 1.09776243], [25.80319277, -0.86077386, 1.09776243], [26.27234173, -0.17291401, 1.09776243], [26.74149069, -0.34582803, 1.09776243], [27.21063965, -0.51874204, 1.09776243], [27.67978861, -0.68975795, 1.09776243], [28.14893757, -0.86077386, 1.09776243], [28.61808653, -0.17291401, 1.09776243], [29.08723549, -0.34582803, 1.09776243], [29.55638445, -0.51874204, 1.09776243], [30.02553341, -0.68975795, 1.09776243], [30.49468237, -0.86077386, 1.09776243], [30.96383133, -0.17291401, 1.09776243], [31.43298029, -0.34582803, 1.09776243], [31.90212925, -0.51874204, 1.09776243], [32.37127821, -0.68975795, 1.09776243], [32.84042717, -0.86077386, 1.09776243], [33.30957613, -0.17291401, 1.09776243], [33.77872509, -0.34582803, 1.09776243], [34.24787405, -0.51874204, 1.09776243], [34.71702301, -0.68975795, 1.09776243], [35.18617197, -0.86077386, 1.09776243], [35.65532093, -0.17291401, 1.09776243], [36.12446989, -0.34582803, 1.09776243], [36.59361885, -0.51874204, 1.09776243], [37.06276781, -0.68975795, 1.09776243], [37.53191677, -0.86077386, 1.09776243], [37.99106573, -0.17291401, 1.09776243], [38.46021469, -0.34582803, 1.09776243], [38.92936365, -0.51874204, 1.09776243], [39.39851261, -0.68975795, 1.09776243], [39.86766157, -0.86077386, 1.09776243], [40.33681053, -0.17291401, 1.09776243], [40.80595949, -0.34582803, 1.09776243], [41.27510845, -0.51874204, 1.09776243], [41.74425741, -0.68975795, 1.09776243], [42.21340637, -0.86077386, 1.09776243], [42.68255533, -0.17291401, 1.09776243], [43.15170429, -0.34582803, 1.09776243], [43.62085325, -0.51874204, 1.09776243], [44.09000221, -0.68975795, 1.09776243], [44.55915117, -0.86077386, 1.09776243], [45.02830013, -0.17291401, 1.09776243], [45.49744909, -0.34582803, 1.09776243], [45.96659805, -0.51874204, 1.09776243], [46.43574701, -0.68975795, 1.09776243], [46.90489597, -0.86077386, 1.09776243], [47.37404493, -0.17291401, 1.09776243], [47.84319389, -0.34582803, 1.09776243], [48.31234285, -0.51874204, 1.09776243], [48.78149181, -0.68975795, 1.09776243], [49.25064077, -0.86077386, 1.09776243], [49.71978973, -0.17291401, 1.09776243], [50.18893869, -0.34582803, 1.09776243], [50.65808765, -0.51874204, 1.09776243], [51.12723661, -0.68975795, 1.09776243], [51.59638557, -0.86077386, 1.09776243], [52.06553453, -0.17291401, 1.09776243], [52.53468349, -0.34582803, 1.09776243], [53.00383245, -0.51874204, 1.09776243], [53.47298141, -0.68975795, 1.09776243], [53.94213037, -0.86077386, 1.09776243], [54.41127933, -0.17291401, 1.09776243], [54.88042829, -0.34582803, 1.09776243], [55.34957725, -0.51874204, 1.09776243], [55.81872621, -0.68975795, 1.09776243], [56.28787517, -0.86077386, 1.09776243], [56.75702413, -0.17291401, 1.09776243], [57.22617309, -0.34582803, 1.09776243], [57.69532205, -0.51874204, 1.09776243], [58.16447101, -0.68975795, 1.09776243], [58.63362001, -0.86077386, 1.09776243], [59.10276897, -0.17291401, 1.09776243], [59.57191793, -0.34582803, 1.09776243], [60.04106689, -0.51874204, 1.09776243], [60.51021585, -0.68975795, 1.09776243], [60.97936481, -0.86077386, 1.09776243], [61.44851377, -0.17291401, 1.09776243], [61.91766273, -0.34582803, 1.09776243], [62.38681169, -0.51874204, 1.09776243], [62.85596065, -0.68975795, 1.09776243], [63.32510961, -0.86077386, 1.09776243], [63.79425857, -0.17291401, 1.09776243], [64.26340753, -0.34582803, 1.09776243], [64.73255649, -0.51874204, 1.09776243], [65.20170545, -0.68975795, 1.09776243], [65.67085441, -0.86077386, 1.09776243], [66.14000337, -0.17291401, 1.09776243], [66.60915233, -0.34582803, 1.09776243], [67.07830129, -0.51874204, 1.09776243], [67.54745025, -0.68975795, 1.09776243], [68.01659921, -0.86077386, 1.09776243], [68.48574817, -0.17291401, 1.09776243], [68.95489713, -0.34582803, 1.09776243], [69.42404609, -0.51874204, 1.09776243], [69.89319505, -0.68975795, 1.09776243], [70.36234401, -0.86077386, 1.09776243], [70.83149297, -0.17291401, 1.09776243], [71.30064193, -0.34582803, 1.09776243], [71.76979089, -0.51874204, 1.09776243], [72.23893985, -0.68975795, 1.09776243], [72.70808881, -0.86077386, 1.09776243], [73.17723777, -0.17291401, 1.09776243], [73.64638673, -0.34582803, 1.09776243], [74.11553569, -0.51874204, 1.09776243], [74.58468465, -0.68975795, 1.09776243], [75.05383361, -0.86077386, 1.09776243], [75.52298257, -0.17291401, 1.09776243], [75.99213153, -0.34582803, 1.09776243], [76.46128049, -0.51874204, 1.09776243], [76.93042945, -0.68975795, 1.09776243], [77.39957841, -0.86077386, 1.09776243], [77.86872737, -0.17291401, 1.09776243], [78.33787633, -0.34582803, 1.09776243], [78.80702529, -0.51874204, 1.09776243], [79.27617425, -0.68975795, 1.09776243], [79.74532321, -0.86077386, 1.09776243], [80.21447217, -0.17291401, 1.09776243], [80.68362113, -0.34582803, 1.09776243], [81.15277009, -0.51874204, 1.09776243], [81.62191905, -0.68975795, 1.09776243], [82.09106801, -0.86077386, 1.09776243], [82.56021697, -0.17291401, 1.09776243], [83.02936593, -0.34582803, 1.09776243], [83.49851489, -0.51874204, 1.09776243], [83.96766385, -0.68975795, 1.09776243], [84.43681281, -0.86077386, 1.09776243], [84.90596177, -0.17291401, 1.09776243], [85.37511073, -0.34582803, 1.09776243], [85.84425969, -0.51874204, 1.09776243], [86.31340865, -0.68975795, 1.09776243], [86.78255761, -0.86077386, 1.09776243], [87.25170657, -0.17291401, 1.09776243], [87.72085553, -0.34582803, 1.09776243], [88.19000449, -0.51874204, 1.09776243], [88.65915345, -0.68975795, 1.09776243], [89.12830241, -0.86077386, 1.09776243], [89.59745137, -0.17291401, 1.09776243], [90.06660033, -0.34582803, 1.09776243], [90.53574929, -0.51874204, 1.09776243], [91.00489825, -0.68975795, 1.09776243], [91.47404721, -0.86077386, 1.09776243], [91.94319617, -0.17291401, 1.09776243], [92.41234513, -0.34582803, 1.09776243], [92.88149409, -0.51874204, 1.09776243], [93.35064305, -0.68975795, 1.09776243], [93.81979201, -0.86077386, 1.09776243], [94.28894097, -0.17291401, 1.09776243], [94.75808993, -0.34582803, 1.09776243], [95.22723889, -0.51874204, 1.09776243], [95.69638785, -0.68975795, 1.09776243], [96.16553681, -0.86077386, 1.09776243], [96.63468577, -0.17291401, 1.09776243], [97.10383473, -0.34582803, 1.09776243], [97.57298369, -0.51874204, 1.09776243], [98.04213265, -0.68975795, 1.09776243], [98.51128161, -0.86077386, 1.09776243], [98.98043057, -0.17291401, 1.09776243], [99.44957953, -0.34582803, 1.09776243], [99.91872849, -0.51874204, 1.09776243], [100.38787745, -0.68975795, 1.09776243], [100.85702641, -0.86077386, 1.09776243], [101.32617537, -0.17291401, 1.09776243], [101.79532433, -0.34582803, 1.09776243], [102.26447329, -0.51874204, 1.09776243], [102.73362225, -0.68975795, 1.09776243], [103.20277121, -0.86077386, 1.09776243], [103.67192017, -0.17291401, 1.09776243], [104.14106913, -0.34582803, 1.09776243], [104.61021809, -0.51874204, 1.09776243], [105.07936705, -0.68975795, 1.09776243], [105.54851601, -0.86077386, 1.09776243], [106.01766497, -0.17291401, 1.09776243], [106.48681393, -0.34582803, 1.09776243], [106.95596289, -0.51874204, 1.09776243], [107.42511185, -0.68975795, 1.09776243], [107.89426081, -0.86077386, 1.09776243], [108.36340977, -0.17291401, 1.09776243], [108.83255873, -0.34582803, 1.09776243], [109.30170769, -0.51874204, 1.09776243], [109.77085665, -0.68975795, 1.09776243], [110.24000561, -0.86077386, 1.09776243], [110.70915457, -0.17291401, 1.09776243], [111.17830353, -0.34582803, 1.09776243], [111.64745249, -0.51874204, 1.09776243], [112.11660145, -0.68975795, 1.09776243], [112.58575041, -0.86077386, 1.09776243], [113.05489937, -0.17291401, 1.09776243], [113.52404833, -0.34582803, 1.09776243], [113.99319729, -0.51874204, 1.09776243], [114.46234625, -0.68975795, 1.09776243], [114.93149521, -0.86077386, 1.09776243], [115.40064417, -0.17291401, 1.09776243], [115.86979313, -0.34582803, 1.09776243], [116.33894209, -0.51874204, 1.09776243], [116.80809105, -0.68975795, 1.09776243], [117.27724001, -0.86077386, 1.09776243], [117.74638897, -0.17291401, 1.09776243], [118.21553793, -0.34582803, 1.09776243], [118.68468689, -0.51874204, 1.09776243], [119.15383585, -0.68975795, 1.09776243], [119.62298481, -0.86077386, 1.09776243], [120.09213377, -0.17291401, 1.09776243], [120.56128273, -0.34582803, 1.09776243], [121.03043169, -0.51874204, 1.09776243], [121.49958065, -0.68975795, 1.09776243], [121.96872961, -0.86077386, 1.09776243], [122.43787857, -0.17291401, 1.09776243], [122.90702753, -0.34582803, 1.09776243], [123.37617649, -0.51874204, 1.09776243], [123.84532545, -0.68975795, 1.09776243], [124.31447441, -0.86077386, 1.09776243], [124.78362337, -0.17291401, 1.09776243], [125.25277233, -0.34582803, 1.09776243], [125.72192129, -0.51874204, 1.09776243], [126.19107025, -0.68975795, 1.09776243], [126.66021921, -0.86077386, 1.09776243], [127.12936817, -0.17291401, 1.09776243], [127.59851713, -0.34582803, 1.09776243], [128.06766609, -0.51874204, 1.09776243], [128.53681505, -0.68975795, 1.09776243], [129.00596401, -0.86077386, 1.09776243], [129.47511297, -0.17291401, 1.09776243], [129.94426193, -0.34582803, 1.09776243], [130.41341089, -0.51874204, 1.09776243], [130.88255985, -0.68975795, 1.09776243], [131.35170881, -0.86077386, 1.09776243], [131.82085777, -0.17291401, 1.09776243], [132.29000673, -0.34582803, 1.09776243], [132.75915569, -0.51874204, 1.09776243], [133.22830465, -0.68975795, 1.09776243], [133.69745361, -0.86077386, 1.09776243], [134.16660257, -0.17291401, 1.09776243], [134.63575153, -0.34582803, 1.09776243], [135.10490049, -0.51874204, 1.09776243], [135.57404945, -0.68975795, 1.09776243], [136.04319841, -0.86077386, 1.09776243], [136.51234737, -0.17291401, 1.09776243], [136.98149633, -0.34582803, 1.09776243], [137.45064529, -0.51874204, 1.09776243], [137.91979425, -0.68975795, 1.09776243], [138.38894321, -0.86077386, 1.09776243], [138.85809217, -0.17291401, 1.09776243], [139.32724113, -0.34582803, 1.09776243], [139.79639009, -0.51874204, 1.09776243], [140.26553905, -0.68975795, 1.09776243], [140.73468801, -0.86077386, 1.09776243], [141.20383697, -0.17291401, 1.09776243], [141.67298593, -0.34582803, 1.09776243], [142.14213489, -0.51874204, 1.09776243], [142.61128385, -0.68975795, 1.09776243], [143.08043281, -0.86077386, 1.09776243], [143.54958177, -0.17291401, 1.09776243], [144.01873073, -0.34582803, 1.09776243], [144.48787969, -0.51874204, 1.09776243], [144.95702865, -0.68975795, 1.09776243], [145.42617761, -0.86077386, 1.09776243], [145.89532657, -0.17291401, 1.09776243], [146.36447553, -0.34582803, 1.09776243], [146.83362449, -0.51874204, 1.09776243], [147.30277345, -0.68975795, 1.09776243], [147.77192241, -0.86077386, 1.09776243], [148.24107137, -0.17291401, 1.09776243], [148.71022033, -0.34582803, 1.09776243], [149.17936929, -0.51874204, 1.09776243], [149.64851825, -0.68975795, 1.09776243], [150.11766721, -0.86077386, 1.09776243], [150.58681617
```

[illegible]

collision detected between robot sphere at (6.09279488, -1.88622458) with radius 0.65768694 and robot sphere at (7.01880968, -1.50873738) with radius 0.65768694
collision detected between robot sphere at (6.55580228, -1.69748098) with radius 0.65768694 and robot sphere at (7.01880968, -1.50873738) with radius 0.65768694
collision detected between robot sphere at (6.55580228, -1.69748098) with radius 0.65768694 and robot sphere at (7.48181709, -1.31999378) with radius 0.65768694
collision detected between robot sphere at (7.01880968, -1.50873738) with radius 0.65768694 and robot sphere at (7.48181709, -1.31999378) with radius 0.65768694
collision detected between robot sphere at (7.01880968, -1.50873738) with radius 0.65768694 and robot sphere at (7.94482449, -1.13125018) with radius 0.65768694
collision detected between robot sphere at (7.48181709, -1.31999378) with radius 0.65768694 and robot sphere at (7.94482449, -1.13125018) with radius 0.65768694
collision detected between robot sphere at (7.48181709, -1.31999378) with radius 0.65768694 and robot sphere at (8.40783189, -0.94250658) with radius 0.65768694
collision detected between robot sphere at (7.94482449, -1.13125018) with radius 0.65768694 and robot sphere at (8.40783189, -0.94250658) with radius 0.65768694

Manipulation

```
In [7]: import numpy as np
import sympy

p = np.array([[ 1, -6, -9]])
n = np.array([[0, 0, 1]])

# Create variables
list_of_variables = "wax, wbx, way, wby, waz, wbz, vax, vbx, vay, vby, vaz, vbz"
wax, wbx, way, wby, waz, wbz, vax, vbx, vay, vby, vaz, vbz = sympy.symbols(list_of_variables)

# Setup symbolic matrix multiplication
p_cross_n = np.cross(p, n)
wrench = np.concatenate([p_cross_n, n], axis=1)
sym_wrench = sympy.Matrix(wrench)
sym_diff_bodies = sympy.Matrix([[wax-wbx],
                                [way-wby],
                                [waz-wbz],
                                [vax-vbx],
                                [vay-vby],
                                [vaz-vbz]])

sym_mat_eq = sym_wrench * sym_diff_bodies

# Get symbolic constraint
sym_eq = sym_mat_eq[0,0]

print(sym_eq)
```

vaz - vbz - 6*wax - way + 6*wbx + wby