Mobile Manipulation - Assignment 4: Null Space and Grasp Maps

L. Carter Price

March 2020

1 Part I

1. Why does the commanded base motion not match the resulting base motion when we multiply by the null space projector?

   The base does not match the set velocities because the Null-space projection is not the identity in the first three rows. Therefore, when it is multiplied by the goal joint velocities, the joint velocities change.

2 Part II

2.1 Reuleaux' Method for frictionless contacts

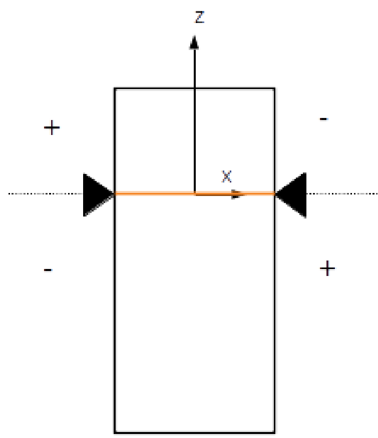

Figure 1: Reuleaux' Method for friction-less contacts

1. Only the IRCs along the normal line are possible according to Reuleaux' method (the orange line), because this is where the "rotation regions" overlap.

2. This does not actually give the correct answer. These contacts cannot generate rotations along the normal line. See the image 2.

3. The IRCs that are possible are at infinity (+Z and -Z) because the contacts can generate linear motion in the x direction.

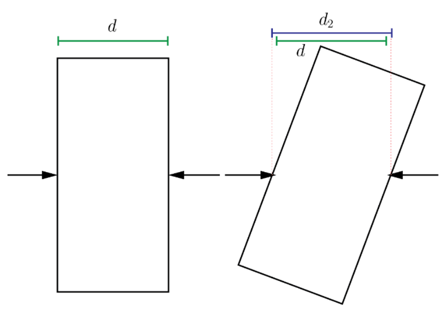

Figure 2: impossible to have IRC along contact line

2.2 Grasp map for Frictionless contacts

1.

$$Ad_T1 = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$Ad_T2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

2.

$$G1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$G2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \\ 0 \\ 0 \end{bmatrix}$$

3.

$$G = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & -1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

4. Possible Wrench

$$f = \begin{bmatrix} 2 & 5 \end{bmatrix}$$

$$F = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -3 \\ 0 \\ 0 \end{bmatrix}$$

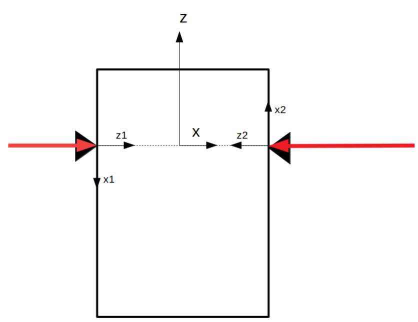

Figure 3: Red arrows indicate contact forces. This is a possible scenario resulting in a force in the negative X_o direction.

Impossible Wrench

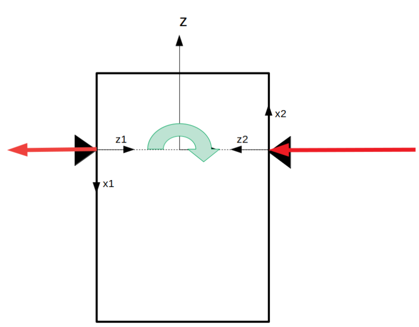

Figure 4: There are 2 problems with wrench. 1 it is impossible to generate a moment about the y axis with the two contacts given. 2. The first contact cannot be less than zero. Mathematically this results in -7 in the $\mathrm{X}_o direction, butthisisimpossible$.

$$f = \begin{bmatrix} -2 & 5 \end{bmatrix}$$

$$F = \begin{bmatrix} 0 \\ -5 \\ 0 \\ -3 \\ 0 \\ 0 \end{bmatrix}$$

2.3 Bonus Question: Point Contact with Friction

1. G_i for each contact

$$G1 = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

$$G2 = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

2. Grasp Map G

$$G = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

3. wrench to balance gravity mg

$$f_c = \begin{bmatrix} -0.5mg & 0 & 0.5mg/\mu \\ 0.5mg & 0 & 0.5mg/\mu \end{bmatrix}$$

$$F = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ mg \end{bmatrix}$$

4. sliding condition mg

$$f_c = \begin{bmatrix} -abs(< 0.5mg) & 0 & < 0.5mg/\mu \\ < 0.5mg & 0 & < 0.5mg/\mu \end{bmatrix}$$

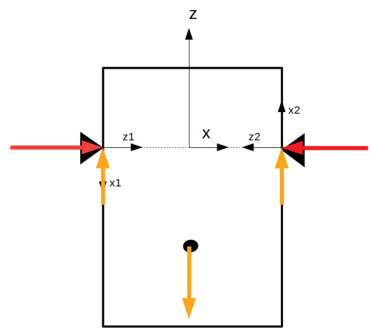

Figure 5: The contact forces in orange counter act the force of gravity and the +Z contact forces (red) are large to enable forces in $+Z_o$ subject to the constraint with the coefficient of friction μ. The red forces balance each other out and the system reaches static equilibrium.

$$F = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ mg > \end{bmatrix}$$

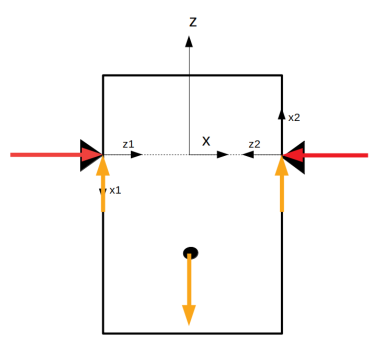

Figure 6: In this case, the contact forces are too small to allow for a friction force that can counter the force of gravity, therefore, the phone slides down. The $+Z_c$ contact force must be less than $0.5mg/\mu$ to allow sliding.

2.4 Grasp Code

```python
#!/usr/bin/env python2
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 16 19:23:02 2020

@author: carter
"""

import numpy as np

#Question 6

#for contact 1
a = np.array([[-1, 0, 0]])
b = np.array([[0, 0, -1],[0,1,0],[1,0,0]])

#for contact 2
a = np.array([[1, 0, 0]])
b = np.array([[0, 0, 1],[0,1,0],[-1,0,0]])

c = np.cross(a,b)
zero = np.zeros((3,3))
temp = np.vstack((zero,b))
Ad = np.hstack((np.vstack((b,c)),temp))

B = np.array([[0],[0],[0],[0],[0],[1]])

out = np.dot(Ad.T,B)


G = np.array([[ 0., 0],
       [ 0., 0],
       [ 0., 0],
       [1., -1],
       [ 0., 0],
       [ 0., 0]])

f = np.array([[-2], [-5]])

F = G.dot(f)

#Question 7

a = np.array([[-1, 0, 0]])
b = np.array([[0, 0, -1],[0,1,0],[1,0,0]])

#for contact 2
a = np.array([[1, 0, 0]])
b = np.array([[0, 0, 1],[0,1,0],[-1,0,0]])

c = np.cross(a,b)
zero = np.zeros((3,3))
temp = np.vstack((zero,b))
Ad = np.hstack((np.vstack((b,c)),temp))

B = np.zeros((6,3))
B[3:,:] = np.identity(3)

out = np.dot(Ad.T,B)
```

```
#combine for parts 3 and 4

G1 = np.array([[ 0.,  0.,  0.],
       [-1.,  0.,  0.],
       [ 0., -1.,  0.],
       [ 0.,  0.,  1.],
       [ 0.,  1.,  0.],
       [-1.,  0.,  0.]])

G2 = np.array([[ 0.,  0.,  0.],
       [-1.,  0.,  0.],
       [ 0.,  1.,  0.],
       [ 0.,  0., -1.],
       [ 0.,  1.,  0.],
       [ 1.,  0.,  0.]])

G = np.hstack((G1,G2))

#calculate contact forces with constraints
m = 1
g = 9.8
mu = 0.2
f1 = 0.5*m*g
f3 = np.sqrt(f1**2)/mu

fc1 = np.array([[-f1],[0],[f3]])
fc2 = np.array([[f1],[0],[f3]])

fc = np.vstack((fc1,fc2))

F = np.dot(G,fc)
print(F)
```