CS 237B: Principles of Robot Autonomy II
Problem Set 2

Name: Pei-Chen Wu
SUID: pcwu1023

Problem 1: Form and Force Closure

(i) Explain why every form closure grasp is also a force closure grasp!

Form closure is stricter than force closure. For form closure, the object is kinematically constrained. Alternatively, a force closure grasp uses forces applied at contact points to be able to resist any external wrench. Force closure typically rely on friction and generally require fewer contact points than are required for form closure. Therefore, every form closure grasp is also a force closure grasp.

(ii) To achieve form closure for a 2D object, you need at least 4 contacts. To achieve form closure with a 3D object, you need at least 7 contacts. Explain in your own words why this is the case!

A 3D object will have 6 DOF. For j contacts, the j forces need to positively span the 6 dimensional space $\mathbb{R}^6$.The space $\mathbb{R}^n$ can be positively spanned by n+1 vectors, but no fewer. Therefore, for 3D object, you need at least 6+1 contacts. For planar object, the space $\mathbb{R}^3$ can be positively spanned by 3+1 vectors, but no fewer. Therefore, at least 4 contacts will be needed for form closure in 2D.

(iii) Form closure Subsets

Using graphical planar method, if there is no CoR labeled consistently, then the feasible velocity cone consists of only the zero velocity point, and the part is immobilized by the stationary contacts.
Subsets 1,2,3,5 and 1,3,4,5 will have form closure.

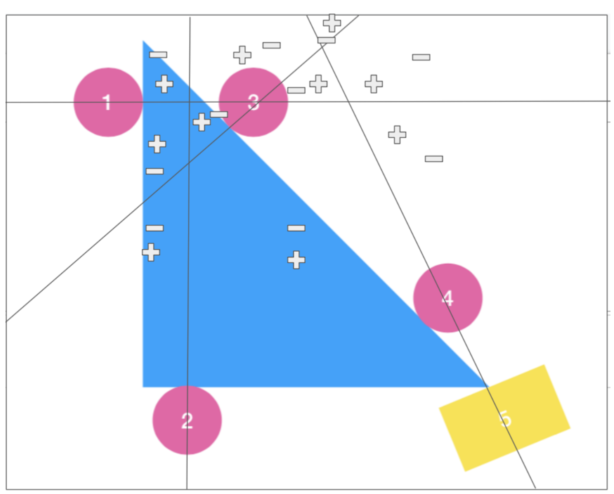

Figure 1: Form Closure 1: 1,2,3,5

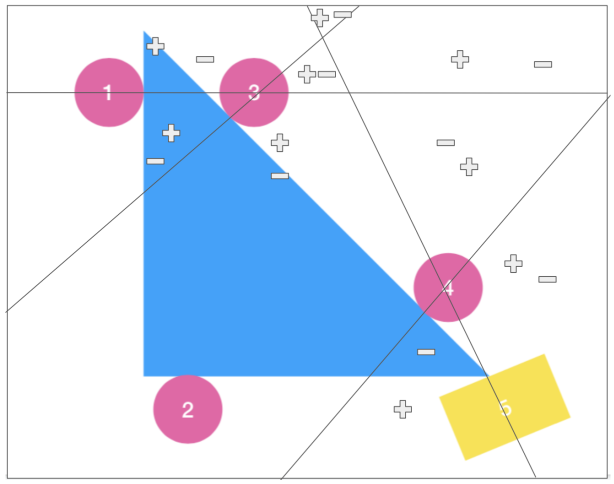

Figure 2: Form Closure 2: 1,3,4,5

(iv) CODE

(v) CODE

(vi) Force closure: the range of μ

From the problem statement, we will have 4 wrenches and we can construct the following F matrix.

$$\begin{bmatrix} -\mu & \mu & 0 & -1 \\ 1 & 1 & -1 & 0 \\ 0 & 0 & -c & h \end{bmatrix}$$

Figure 3: F matrix

From force closure condition $Fk = 0$, we will have the following 3 equations.
$-\mu K_1 + \mu K_2 - K_4 = 0$
$K_1 + K_2 = K_3$
$-cK_3 + hK_4 = 0$

Rearrange the above equations, we will get $\mu = \frac{K_4}{-K_1+K_2}$ and $\frac{c}{h} = \frac{K_4}{K_1+K_2}$
Since all the K_i and μ need to be positive.
We can get $\mu > \frac{c}{h}$.
Plugging in $c = 1/4$ and $h = 1/2$, we will get, $\mu > 0.5$

Problem 2: Grasp Force Optimization

(i) Show how to write force equilibrium constraints to grasp map and f.

$$\Phi = \begin{bmatrix} T^{(1)} & T^{(2)} & ... & T^{(M)} \\ P^{(1)}_{(\times)}T^{(1)} & P^{(2)}_{(\times)}T^{(2)} & ... & P^{(M)}_{(\times)}T^{(M)} \end{bmatrix}$$

$$f = \begin{bmatrix} f^{(1)} \\ f^{(2)} \\ ... \\ f^{(M)} \end{bmatrix}$$

$$w^{ext} = \begin{bmatrix} f^{ext} \\ \tau^{ext} \end{bmatrix}$$

For making above equations to be SOCP form, we will have to expand $f^(i)$, $T^{(i)}$ and $P^{(i)}_{\times}$ to be either 3D ($\in \mathbb{R}^6$) or 2D ($\in \mathbb{R}^3$) accordingly.

(ii) Show how to reset the objective to a linear form by adding second order cone constraints.

We can minimize an additional scalar variable s. In $h^T x$, we can set up one additional variable s append to the end of x and add 1 to the end of h as following, which ends up minimizing s only.

$$h = \begin{bmatrix} 0 \\ 0 \\ ... \\ 1 \end{bmatrix}, h \in \mathbb{R}^{3M+1}, \mathbb{R}^{2M+1}$$

$$x = \begin{bmatrix} f^{(1)}_x \\ f^{(1)}_y \\ ... \\ f^{(M)}_x \\ f^{(M)}_y \\ s \end{bmatrix}, x \in \mathbb{R}^{3M+1}, \mathbb{R}^{2M+1}$$

For minimizing the variable s, we will have additional M constraints for all the $||f^{(i)}|| \leq s$. The full SOCP form is defined in the next problem.

(iii) Write out all the variables so we can send them to the SOCP solver directly.

I only wrote out the matrices in spatial case.

$$x = \begin{bmatrix} f^{(1)}_x \\ f^{(1)}_y \\ f^{(1)}_z \\ ... \\ f^{(M)}_x \\ f^{(M)}_y \\ f^{(M)}_z \\ s \end{bmatrix}, x \in \mathbb{R}^{3M+1}$$

$$h = \begin{bmatrix} 0 \\ 0 \\ ... \\ 1 \end{bmatrix}, h \in \mathbb{R}^{3M+1}$$

Total number of cone constraints A_i will be 2M.
For $||f^{(i)}|| \leq s$ constraint, A_i will be the following,

$$A_1 = \begin{bmatrix} 1 & 0 & 0 & ... & 0 \\ 0 & 1 & 0 & ... & 0 \\ 0 & 0 & 1 & ... & 0 \end{bmatrix}, A_1 \in \mathbb{R}^{3\times(3M+1)}$$

$$A_2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & ... & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & ... & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & ... & 0 \end{bmatrix}, A_2 \in \mathbb{R}^{3\times(3M+1)}$$

For friction cone constraints, A_i will be the following,

$$A_1 = \begin{bmatrix} 1 & 0 & 0 & ... & 0 \\ 0 & 1 & 0 & ... & 0 \\ 0 & 0 & 0 & ... & 0 \end{bmatrix}, A_1 \in \mathbb{R}^{3\times(3M+1)}$$

$$A_2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & ... & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & ... & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & ... & 0 \end{bmatrix}, A_2 \in \mathbb{R}^{3\times(3M+1)}$$

b_i is zero vector for both friction cone and $||f^{(i)}|| \leq s$ constraints.

$$b_i = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, b_i \in \mathbb{R}^3$$

c_i is the same for all $||f^{(i)}|| \leq s$ constraints.

$$c_i = \begin{bmatrix} 0 \\ 0 \\ ... \\ 1 \end{bmatrix}, c_i \in \mathbb{R}^{3M+1}$$

For friction cone constraints,

$$c_1 = \begin{bmatrix} 0 \\ 0 \\ \mu_1 \\ ... \\ 0 \end{bmatrix}, c_1 \in \mathbb{R}^{3M+1}$$

$$c_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \mu_2 \\ ... \\ 0 \end{bmatrix}, c_2 \in \mathbb{R}^{3M+1}$$

d_i is zero for both friction cone and $||f^{(i)}|| \leq s$ constraints.
F is Φ. $T^{(i)}$ and $P^{(i)}_{(\times)}T^{(i)}$ will be $\in \mathbb{R}^{3\times 3}$ and $g = -w^{ext}$.

$$F = \begin{bmatrix} T^{(1)} & T^{(2)} & ... & T^{(M)} \\ P^{(1)}_{(\times)}T^{(1)} & P^{(2)}_{(\times)}T^{(2)} & ... & P^{(M)}_{(\times)}T^{(M)} \end{bmatrix}$$

$$g = -\begin{bmatrix} f^{ext} \\ \tau^{ext} \end{bmatrix}$$

(iv) CODE

(v) CODE

Problem 3: Learning Intuitive Physics

(i) CODE

(ii) CODE

(iii) CODE

(iv) Report the training and validation losses.

The training and validation losses of the network with embedded physics model are 0.0309 and 1.537 respectively.

```
Epoch 00050: saving model to trained_models/cp-050.ckpt
20/20 [==============================] - 89s 4s/step - loss: 0.0309 - val_loss: 1.5347
```

Figure 4: Network with embedded physics model

(v) How confident is its predictions of the materials class (p_class)? Why are some μ_i negative? Can you think of a way to force the neural network to output positive μ_i?

Most of the highest probability of p_class is about 0.1 to 0.3, which is not very high as I expected. The predicted μ is $\sum_i P_i \mu_i$, so after P class times μ class, the predicted μ becomes close to the ground truth acceleration, which performs pretty good.

For having all positive μ_{class}, we can constraint the weights for the last second layer (before acceleration law) to be non negative. Setting kernal constraint to be "tf.keras.constraints.NonNeg", so the μ_i will all be positive numbers. The following pictures show the results of both with and without non negative constraints. However, with non-negative constraint, the model has a bit higher validation loss.

Figure 5: with non-negative constrain

Figure 6: without non-negative constraint

Figure 7: Highest P class probability is 0.322

(vi) CODE

(vii) Report baseline model training and validation losses. How does it compare to the physics network?

Baseline model's training and validation losses are 0.069 and 2.5657 respectively. Comparing with physics network, adding acceleration law is better for the model to learn the physical properties. From my results, both training and validation losses are lower when we have physics model embedded in the CNN model.

```
Epoch 00050: saving model to trained_models/cp-050.ckpt
20/20 [==============================] - 87s 4s/step - loss: 0.0609 - val_loss: 2.5657
```

Figure 8: Baseline model loss