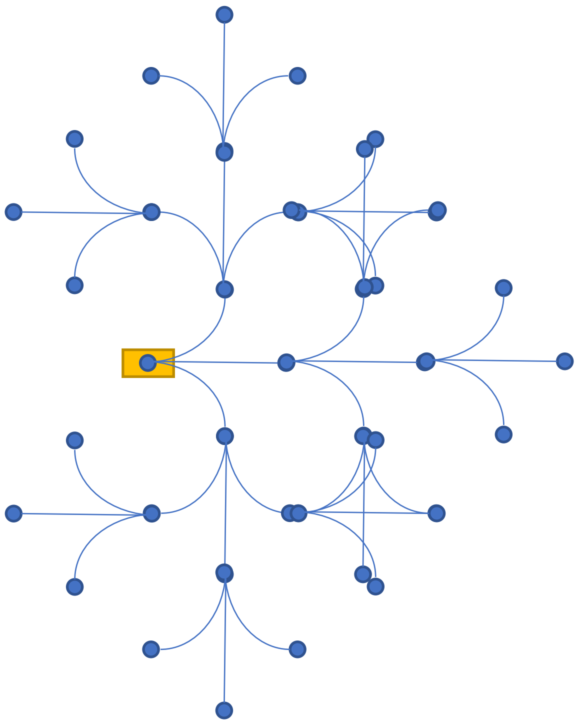
Questions

1. 3. Make a figure similar to Figure 14.6b, but for three stages of the Reeds-Shepp car.



2. Explain why it is more difficult to define a distance metric that leads to good performance for a non-holonomic motion planning problem than a holonomic one. Use a car as an example that illustrates this difficulty and include diagrams.

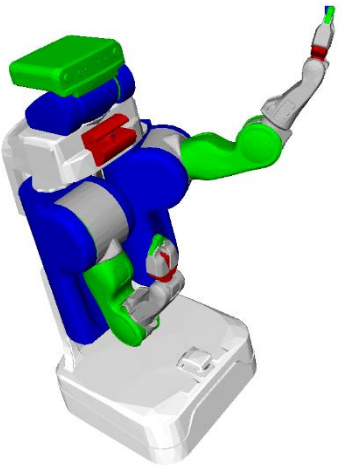
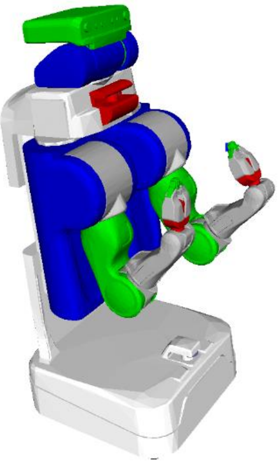
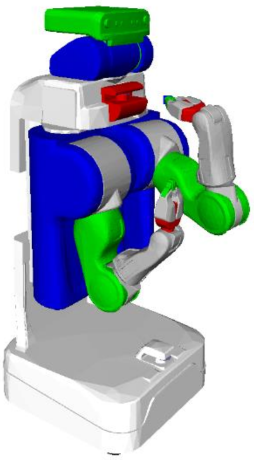
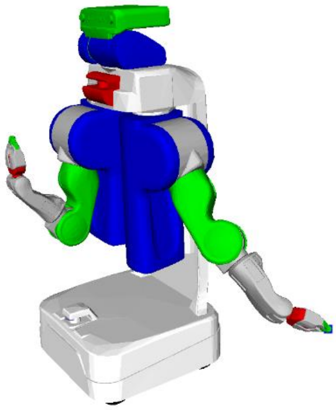
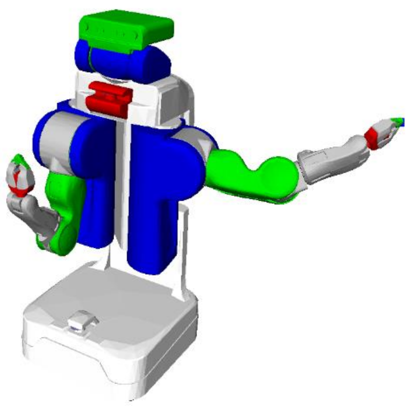
   Non-holonomic problem is harder to move around since its available motion is limited. For example, a car is limited in forward, reverse, and turning in curve with a maximum curvature which means move to a location sideway is harder for a car that it needs to go forward, turn a bit, then backward. In comparison, a holonomic vehicle could simply rotate to the direction of goal then move forward. Because of that, a distance metric for non-holonomic should consider which step is available and it is not suitable to use a simple Euclidean distance metric that holonomic vehicle uses.

3. Consider a robot finger making a point contact on a plane made of homogeneous material. The friction coefficient between the finger and the plane is unknown and we would like to determine it. Assume that the robot can apply any force at the contact point and can sense the position of the contact point perfectly. Assuming the standard Coulomb friction model, write down an algorithm in pseudo-code for applying a series of forces at the contact that would determine the friction coefficient.

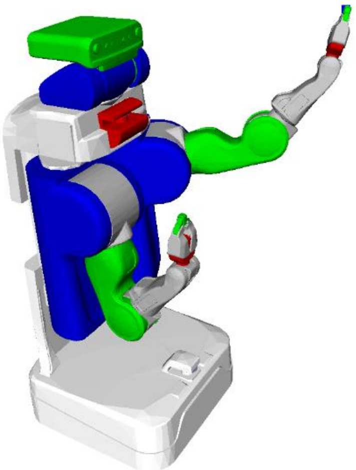
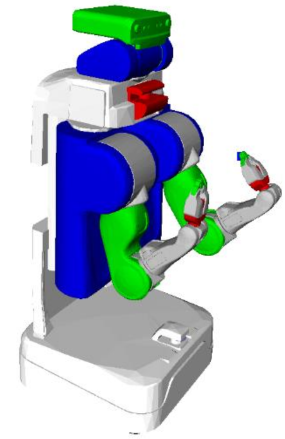
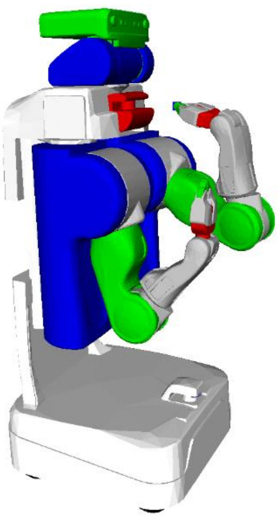
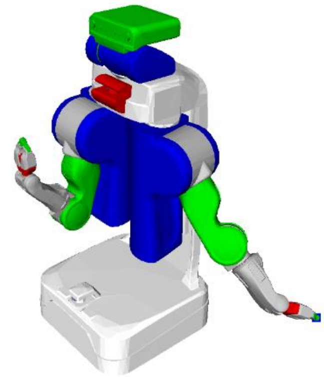
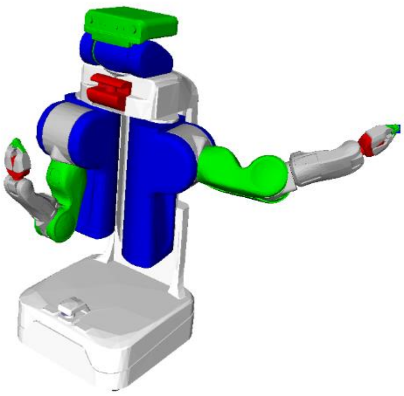
```
Assuming the plane is fixed.
Given a constant force with direction in the normal direction of the plane
Tangent force = 0
While True
    Increase the tangent force that finger applied
    If position of contact point changed
        Friction coefficient = current tangent force / normal force
        Return Friction coefficient
```

Implementation

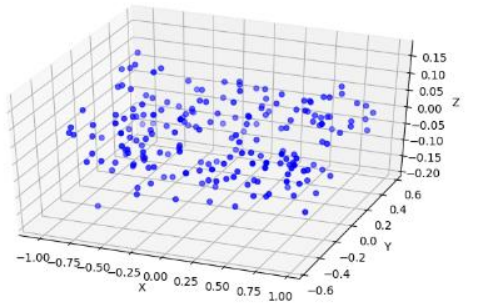
1. (c)Iterative Jacobian Pseudo-Inverse Inverse Kinematics

Target 0	Target 1
	
Target 2	Target 3
	
Target 4	
	

(d)IK with left null-space of the Jacobian pseudo-inverse

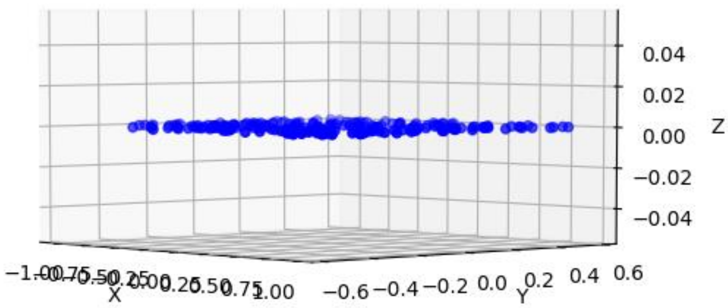
Target 0	Target 1
	
Target 2	Target 3
	
Target 4	
	

2. (a) PCA



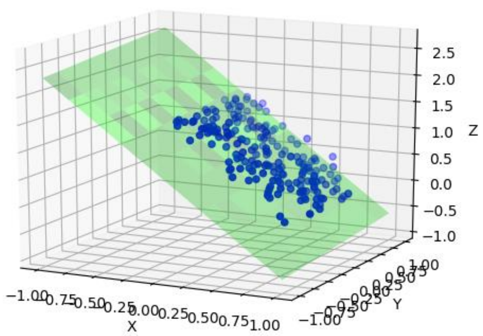
$$V^T = \begin{bmatrix} -0.52935 & -0.00254 & 0.84840 \\ -0.05809 & -0.99754 & -0.03923 \\ 0.84641 & -0.07005 & 0.52790 \end{bmatrix}$$

(b) Noise elimination

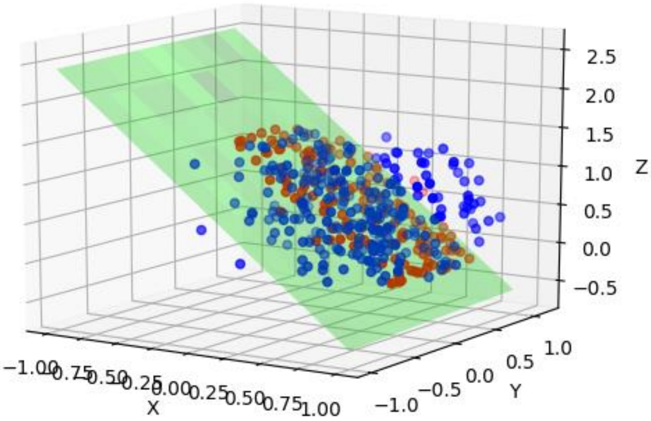


$$V_s^T = \begin{bmatrix} -0.52935 & -0.00254 & 0.84840 \\ -0.05809 & -0.99754 & -0.03923 \\ 0 & 0 & 0 \end{bmatrix}$$

(c) Plane fitting

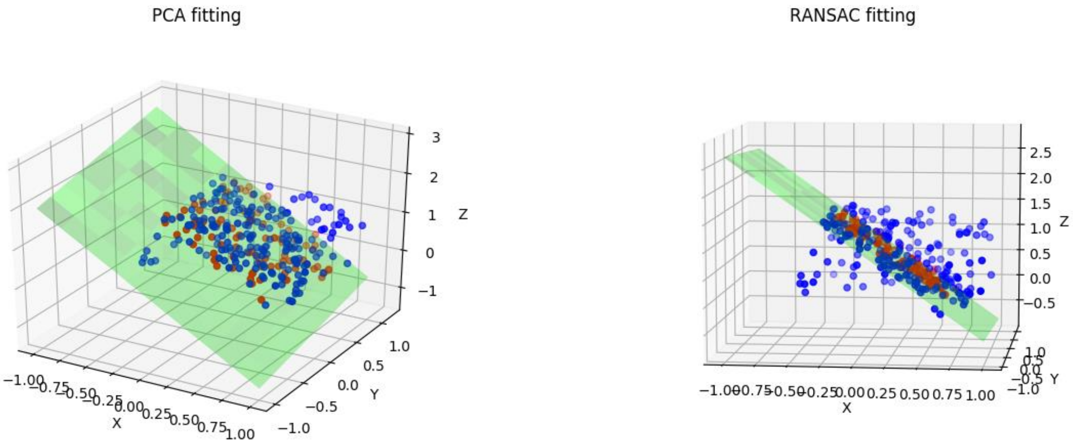


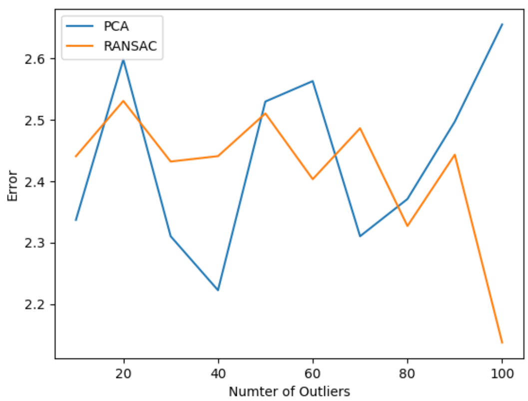
3. RANSAC



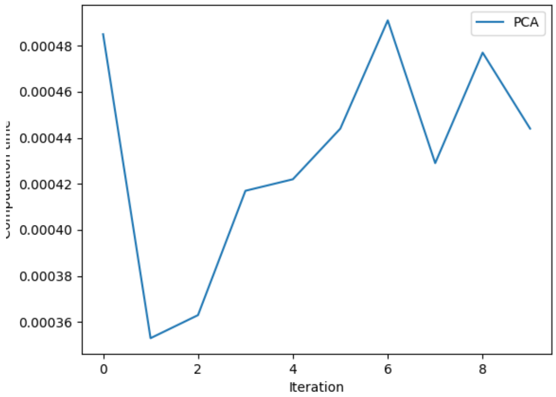
Plane equation: $0.8315(x - 0.2319) + 0.0271(y - 0.4361) + 0.5548(z - 0.5628)$

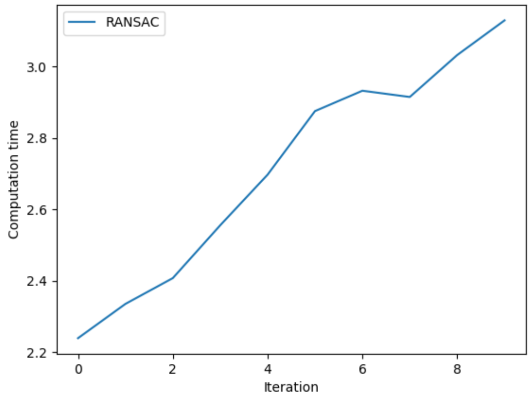
4. PCA vs RANSAC





Number of Outliers vs Error





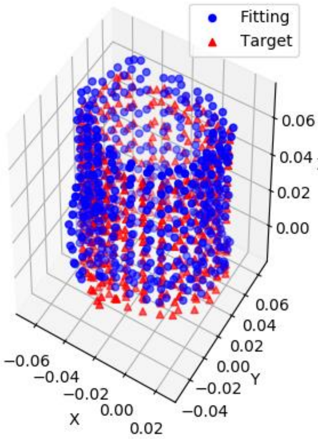
Computation Time of PCA and RANSAC

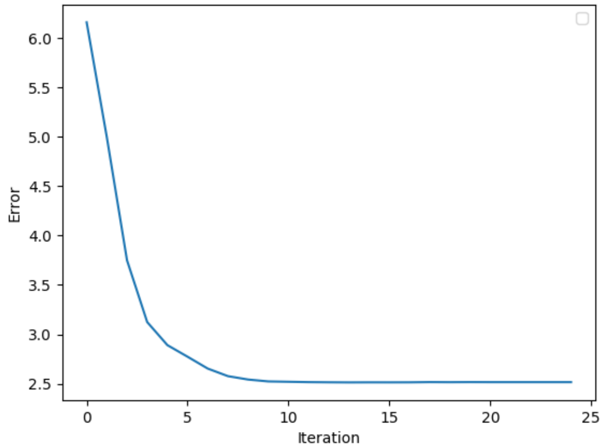
Discussion

In terms of computation time, PCA is faster than RANSAC since RANSAC needs to do random sampling to find the best model it can find comparing to PCA which does the matrix computation directly. In terms of error, RANSAC perform better when number of outliers increases, because RANSAC is able to drop outliers by finding a model that fits a given minimum number of consensus points.

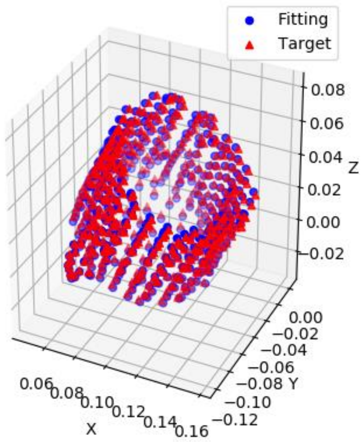
5. ICP

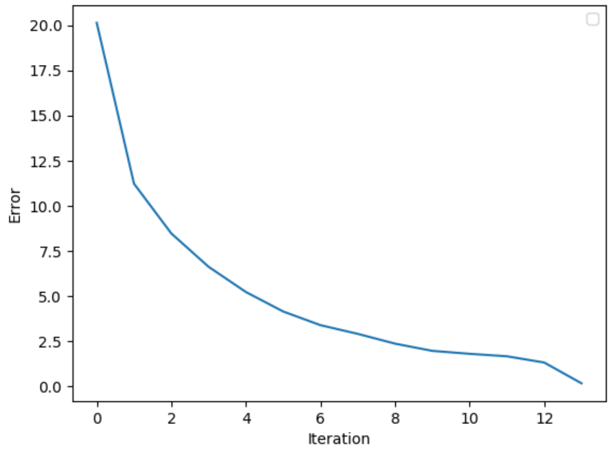
Target 0



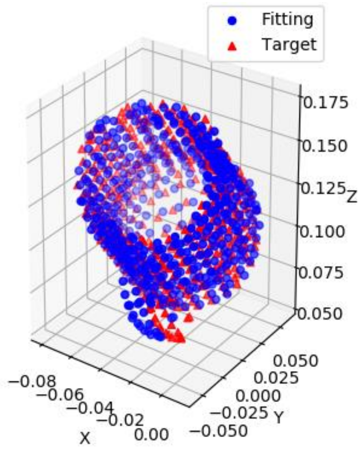


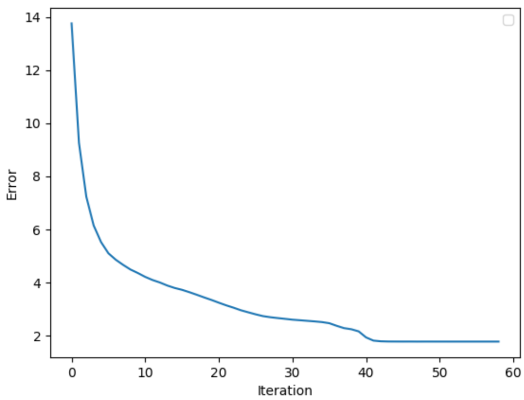
Target 1



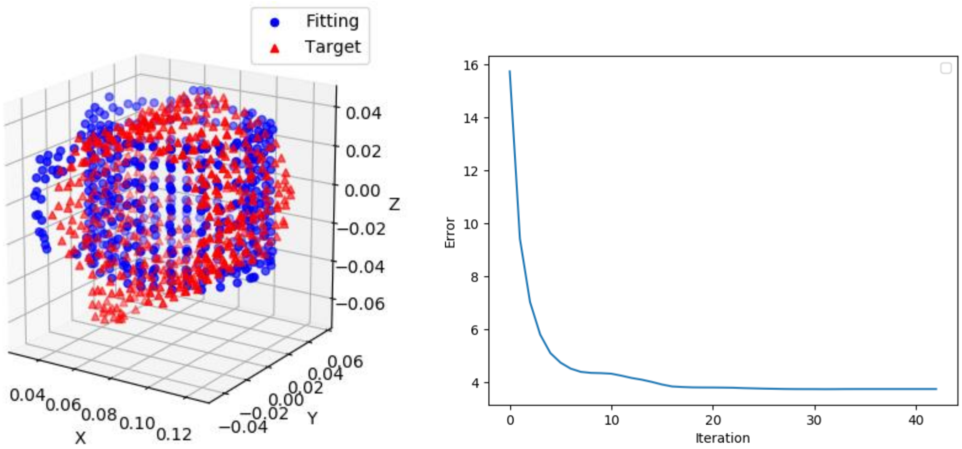


Target 2





Target 3



Discussion

From the better result, target 0 to 2, ICP cannot fit the smaller group of point cloud which is the points that represent the handle of mug, and it seems like the variant of ICP implemented tends to be stuck at the local minimum.

The target 3 is the hardest for ICP since the difference of the pose of the source and the target is largest and the strategy to find closest point is based on Euclidean distance which can not capture the information of surface.