EECS 106B Project 4 - Grasp Planning

Prabhman Dhaliwal Matthew Hallac Jaiveer Singh

***Abstract*—In this paper, we explore a series of metrics to quantitatively predict the success of a grasp with known vertices on an object with known mesh, and compare these predictions with experimental results. The metrics are then applied as part of a novel grasp generation and planning algorithm, which randomly samples vertices on an object mesh to return a perceived best grasp and a gripper configuration that maximizes clearance around the obstacle.**

***Index Terms*—Friction cone - Discretization - Gravity Resistance - Ferrari Canny - Robust Force Closure - Grasp Planning - Contact - Force Closure - Convex - Optimization - Metrics - Wrench - Monte Carlo - Sampling - Game Theory**

I. METHODS

For the purposes of this experiment, we explored three metrics: Gravity Resistance, Ferrari-Canny, and Robust Force Closure. These techniques have been previously referenced in the literature, but the goal of this experiment was to validate the metrics' predictive power against experimental results obtained using a Sawyer robot arm.

A. Helper Routines

1) Generating Grasp Maps: A basic framework of helper functions is critical to the construction of each of these metrics. Specifically, we prepared a subroutine named *get_grasp_map()* to perform the namesake calculation. The essential math behind this function involves transforming the contact frame to the world frame:

$$G = \left[\begin{array}{ccc} \mathrm{Ad}^T_{g^{-1}_{oc_1}} B_{c_1} & \cdots & \mathrm{Ad}^T_{g^{-1}_{oc_k}} B_{c_k} \end{array} \right] \tag{1}$$

2) Friction Cone Discretization: In this experiment, we modelled the forces imparted by a finger using the soft-contact model. This model allows for limited frictional forces in all spatial directions, as well as a torque component, to be imparted by a single finger. As a result, the forces that can be imparted by each of the gripper's fingers lie in a 3-dimensional friction cone, a shape that is nonconvex and thus somewhat difficult to optimize over. However, by effectively discretizing the friction cone into a 'friction pyramid', we were able to convert the relevant problem into a form solvable by the **cvxpy** optimization solver. Essentially, the framework relies on constructing a set of vectors that span the edges of the pyramid, as well as a single vertical vector, and ensures that the resultant force lies within the 3D space spanned by a linear combination of those vectors with positive coefficients.

This optimization problem is solved inside a routine named *contact_forces_exist()*, which ultimately solves problems of the following form:

$$\begin{gathered} \widehat{f} = \underset{f}{\operatorname{argmin}} \|f\|_2^2 \\ \text{st. } f \in FC \\ Gf = w \end{gathered} \tag{2}$$

This function uses *get_grasp_map()* for G, and calculates the minimum input force needed by our gripper to appropriately produce a desired input wrench.

B. Metrics

1) Gravity Resistance metric: The gravity resistance metric is a deterministic metric that improves upon standard force closure by computing how well our grasp is able to resist the force of gravity, since gravity is the source of a large portion of applied forces on any grasp configuration. This metric returns the minimum magnitude force needed to counteract the wrench applied by gravity, which itself is calculated using a straightforward application of the *contact_forces_exist()* subroutine referenced above. Concisely, in equation (2), w is $-W_g$, the wrench of the force of gravity. Smaller scores are better for this metric; a score of infinity implies that no suitable forces are found.

2) Ferrari Canny Metric: Ferrari Canny is a stochastic extension of the force closure metric that quantifies how much effort is needed to maintain force closure without slippage. The original paper, "Planning Optimal Grasps" by Carlo Ferrari and John Canny, state this problem as: "Given the position of the gripper and the object to be grasped, how can we say if this is a good grasp?" Recall that the original force closure concept is a balancing act; the application of any external wrenches and torques are balanced by the fingers forces. By considering the efficiency of a wrench applied in the worst case, and minimizing the force needed in this worst case, we arrive at the value of our metric.

The most important takeaway from this algorithm is that it allows us to quantify which grasp configurations are better than others. While we can think of an arbitrarily large number of ways to apply finger forces such that our grip doesn't slip on the object, this algorithm warrants that the best grasp applies the least amount of force needed, mimicking human grasps. This has numerous advantages, which includes low deformation of the object and lower power consumption to gear the motors. Conceptually, this algorithm is formalizing the relationship between the magnitude of the applied finger forces and the magnitude of the maximum wrench to be resisted. When the ratio between these two quantities are minimized respectively while still satisfying force closure, we have achieved what this algorithm considers the optimal grasp quality. This algorithm operates in two main steps:

- We define a local quality metric, dubbed $LQ(w)$ as a function of wrench space, that measures how efficiently a given wrench w can be resisted. This is the best possible ratio of applied finger forces to resisting the worst case wrench mentioned previously. The lower the force needed, the better the efficiency. We modify this problem in a more code-friendly manner:

$$\begin{aligned} \widehat{LQ}(\widehat{w}) = \min_f \|f\|_2^2 \\ \text{st. } f \in FC \\ Gf = \widehat{w} \end{aligned} \tag{3}$$

- Second, the quality Q returned by our solver is the wrench that minimizes $LQ(w)$ for the worst possible wrenches our system can apply. We want to consider the w that minimizes the worst efficiency, since we typically don't have the granular control to determine which wrenches affect our system the most. It is possible to minimize this with respect to the directions of the wrenches because this quality metric is scale invariant, however this was not done in our implemented code. Notably, we also changed this overall optimization problem to be more code-friendly, where we reduced

$$Q = \min_{\boldsymbol{w}} LQ_{\boldsymbol{w}} \tag{4}$$

equivalently to:

$$Q = \min_{w, \|w\|=1} \frac{1}{\sqrt{\widehat{LQ}(w)}} \tag{5}$$

An additional caveat to this algorithm, and what makes it stochastic in application, is how we solve for Q. An exact solution to this problem is difficult to solve because it assumes we are able to effectively model the wrenches concisely. This assumption was one of the main reasons we decided to do worst case analysis in step 1, and is computationally infeasible in practice. To remedy this, we will randomly sample N 6D vectors representing our wrench forces from a unit cube. These samples come from a standard normal distribution to simulate these wrenches and pick the wrench that minimizes Q. As the number of samples increases towards infinity, we algorithmically cover all the possible situations for wrenches, and thus better approximate the worst case wrench situation. Overall, the variance should tend to 0 the more samples we take. A consideration was given towards precisely computing this solution using the geometry of the level sets of the friction cone, but this was not done in practice, because the payoff was not worth the effort in implementation and runtime. Finally, in our implementation, we used N = 1000 for the number of wrench samples.

3) Robust Force Closure Metric: This metric asks the question "how much can we perturb the grasp and still maintain force closure"? This is another way to qualify what the "best" grasp configuration is. For example, if one of the multiple grasp configurations that maintains force closure is weak, and any moderate perturbation breaks the grasp on the object, then we would like to know if we have stronger options that can withstand tougher random motions. This problem requires a more complex solution than the boolean that our *contact_forces_exist()* function returns; we not only require to know if something is a force closure, but also what is the probability that a random perturbation breaks that force closure. The grasp configuration we are searching for is the one that maintains the highest probability of staying firm to a random disturbance. The end effector position of the grasp is now described fully by a pose μ in $SE(3)$, and we also have a model for the object. With our given inputs, we describe our implementation in these steps:

- We randomly sample a small amount of noise from IID multivariate normals with mean 0 and covariance matrix $I\sigma^2$ to act as our deviation from a configuration in force closure.
- We perturb each end effector pose μ by each sample of noise by adding our sampled noise to the x and y coordinates of our contact points, simulating a (possibly) less optimal hold on the object.
- For each sample, check if our grasp maintained force closure. We can calculate the probability this grasp stays in force closure by taking the number of successful samples over the total number of samples. This probability is noted as:

$$Q(\mu, \mathcal{M}) = \mathbb{P}(F(\hat{\mu}, \mathcal{M}) = 1) \tag{6}$$

where $\mathcal{M}$ is our object model. We always recompute the force closure using our helper function, *contact_forces_exist()*.

We could have alternatively sampled a vector from $se(3)$. However, the input to this step was the pair of vertices, so to do so would require computing the pose of the gripper, applying the sampled delta, and then recomputing the contact vertices. This would take more time and be less convenient than the approach implemented. Similar to Ferrari Canny, we used N = 1000 samples.

C. Custom Grasp Planning Algorithm

The custom grasp plan algorithm starts with computing a pair of contact point vertices for the grasp. The algorithm was as follows:

1) Sample two vertices randomly from the surface of the mesh
2) Ensure that the distance between these vertices is less than the minimum gripper distance and greater than the maximum gripper distance. If not, repeat the step 1 to resample.

3) Compute a grasping metric using the sampled pair of vertices. The grasping metric used for the samples in this paper was the Robust Force Closure metric.
4) Repeat steps 1-3 many times, and save each sample. The grasps in experimental results were generated using 1000 samples.
5) Of all the samples, return the pair of vertices that corresponds to the highest metric.

Once a pair of vertices is generated, the next step is to find an approach angle for the gripper. The set of potential poses for the gripper is a ring around the axis between the two vertices. The gripper cannot approach from underneath the object, so only angles approaching from the upper hemisphere are considered. To find the best approach angle, iterate over the possible angles within this set of potential poses, and find the one that has the largest buffer, or the most space on both sides from a failed grasp. To test if an approach angle is a success or failure, check if there are any collisions with the object in a line segment running along the finger and along the base of the fingers. If there is, the approach angle is a failure. If not, the approach angle is a success. The pseudocode for this algorithm is provided below.

Algorithm 1 Custom Grasp Algorithm

```
Input: Object mesh object_mesh, 2 3D points vertices
Output: 4x4 rigid transform specifying pose of end-effector
1: Set dist equal to the distance between the vertex $v_1$ and vertex $v_2$
2: Set MaxGrip equal to the maximum distance the gripper can extend
3: Set MinGrip equal to the minimum distance the gripper can close
4: Set GripperLength equal to the length of the finger of a gripper
5: if (dist $>$ MaxGrip or dist $<$ MinGrip) then
6:     return None
7: end if
8: $d\theta = 0.001$
9: Initialize empty list thetas
10: Initialize empty list valid
11: Compute a transformation matrix $G$ that transforms to the coordinate frame with the origin at one of the vertices and the z-axis pointed toward the other vertex. Compute such that the y axis is as vertical as it can be (This is implemented in code as the function $look\_at\_general()$)

12: for $\theta = 0...\pi$ with step $d\theta$ do
13:     Compute the x, y, and z positions of the points at the end of the finger for each vertex as follows:
14:     $x_1$ = GripperLength * $\cos\theta$
15:     $y_1$ = GripperLength * $\sin\theta$
16:     $z_1$ = 0
17:     $p_1$ = ($x_1$, $y_1$, $z_1$, 1)
18:     $x_2$ = GripperLength * $\cos\theta$
19:     $y_1$ = GripperLength * $\sin\theta$
20:     $z_1$ = dist
21:     $p_2$ = ($x_2$, $y_2$, $z_2$, 1)
22:     BasePt1 = $G * p_1$
23:     BasePt2 = $G * p_2$
24:     Check if there are any intersecions between the line segments between $v_1$ and BasePt1 and the mesh, and $v_2$ and BasePt2 and the mesh.
25:     Add (theta, BasePt1, and BasePt2) to the thetas array and a boolean intersections variable to the valid array.
26: end for
27: Find the longest sequence of True in the valid array
28: Find the transformation to the theta corresponding to the index of the center of the longest sequence in the valid array, return this transformation.
=0
```

By finding the theta with the greatest amount of valid angles surrounding it, we ensure that the gripper will be resistant to noise along the axis of angles.

II. Experimental Results

A. Grasp Metric Values

The metrics in the following tables were gathered from the grasps provided. For each object and each grasp, each metric was computed. The success rate is the percentage of trials in which the object was successfully lifted and moved, and was generated using the results of 5 trials. For the robust force closure metric, the standard deviation of the perturbances was 5mm and the number of samples used was 1000. Note that for the Gravity Resistance metric, a lower score corresponds to a better grasp, and for the Ferrari-Canny and the Robust Force Closure metrics, a higher score corresponds to a better grasp.

Metrics for Pawn Grasps				
Grasp	Gravity Resistance	Ferrari Canny	Robust FC	Success Rate
1	223.641	4.28×10^{-4}	0.242	100%
2	63.961	1.04×10^{-3}	0.290	0%
3	7.931	9.14×10^{-3}	0.454	80%
4	5.093	1.32×10^{-2}	0.314	20%
5	8.973	9.77×10^{-3}	0.465	80%
Metrics for Nozzle Grasps				
Grasp	Gravity Resistance	Ferrari Canny	Robust FC	Success Rate
1	5.901	5.11×10^{-3}	0.368	100%
2	6.042	1.23×10^{-2}	0.046	0%
3	5.952	4.59×10^{-3}	0.337	100%
4	6.467	6.84×10^{-3}	0.067	0%
5	5.901	5.91×10^{-3}	0.130	60%

B. Custom Grasps

Figures 1, 2, and 3 show examples of grasps from our custom grasp algorithm as described above on the pawn object. Figures 4, 5, 6 show examples of grasps from our custom grasp algorithm on the nozzle object. The metrics

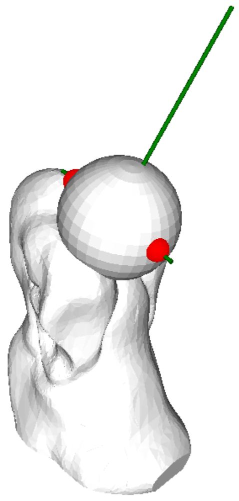
Fig. 2. Custom grasp 2 for the pawn object. This grasp had a gravity resistance score of 7.307, a Ferrari-Canny score of 1.08×10^{-2}, and a robust force closure score of 0.46. The approach angle chosen was approximately 112.13 degrees.

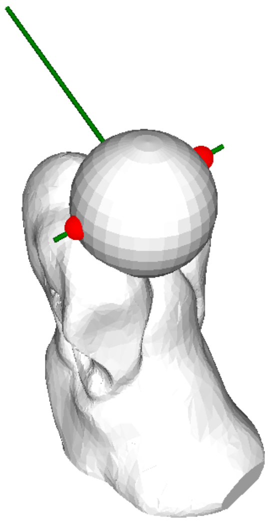
Fig. 1. Custom grasp 1 for the pawn object. This grasp had a gravity resistance score of 10.428, a Ferrari-Canny score of 1.05×10^{-2}, and a robust force closure score of 0.49. The approach angle chosen was approximately 18.16 degrees.

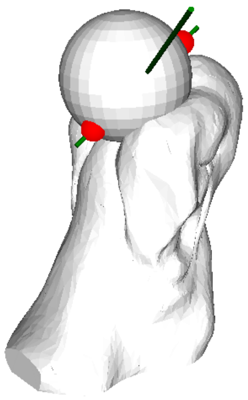
Fig. 3. Custom grasp 3 for the pawn object. This grasp had a gravity resistance score of 3.850, a Ferrari-Canny score of 5.42×10^{-3}, and a robust force closure score of 0.45. The approach angle chosen was approximately 34.43 degrees.

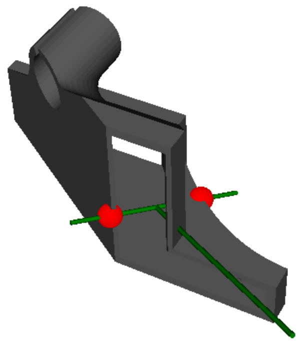
Fig. 4. Custom grasp 1 for the nozzle object. This grasp had a gravity resistance score of 5.159, a Ferrari-Canny score of 1.1×10^{-2}, and a robust force closure score of 0.4. The approach angle chosen was approximately 164.10 degrees.

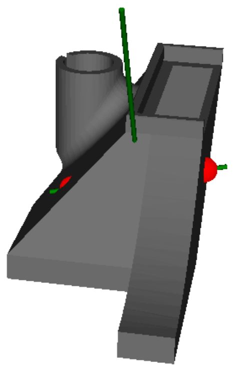
Fig. 5. Custom grasp 2 for the nozzle object. This grasp had a gravity resistance score of 2.427, a Ferrari-Canny score of infinity, and a robust force closure score of 0.27. The approach angle chosen was approximately 179.97 degrees.

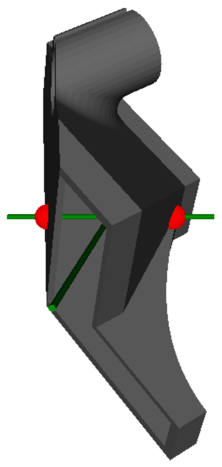
Fig. 6. Custom grasp 3 for the nozzle object. This grasp had a gravity resistance score of 5.916, a Ferrari-Canny score of 4.8×10^{-3}, and a robust force closure score of 0.28. The approach angle chosen was approximately 35.75 degrees.

III. Discussion

A. Difficulties and Edge Cases of Custom Grasp Planning Algorithm

In designing the grasp planning algorithm, one concern was ensuring that the randomly-sampled vertices considered as locations for the fingers to grasp were appropriately within the range of the gripper's actuation distance. Ultimately, our solution was simply to increase the number of candidate pairs randomly generated from the mesh, until this pool was large enough that a specific successful grasp could be chosen. This approach does have some scalability concerns; a significantly larger object than the two considered in this experiment might naturally lend itself to candidate vertex pairs that are further apart on average, in such a way that even the increase iteration count fails to produce a sufficiently close-together pair.

B. Physical Principles Differentiating Success from Failure

Intuitively our grasps metrics are trying to find contact points such that the object does not slip out of the grip of the robot. Physically, in the videos, the gripper is most successful when it has a solid grasp on points that have lots of surface area around it. Meaning, grabbing an object from the corner or at an edge makes it more difficult to maintain a solid grasp. This can be seen which the pawn grasp, where grasp 1 has solid contact on two points that are in the dead center of the round part of the top of the pawn, and grasp two fails because it tries to grab the round part where there is a lot of curvature. This is equivalent to trying to pincer a ball at the edge of a ball. Your fingers will be close to one another and have difficulty finding enough area to clasp onto. Our metrics predict that these sorts of contact points are more prone to failure, and this matches the physical intuition: it is important to have solid contact points.

C. *Predictive Power of Grasp Metrics*

In broad terms, the Robust Force Closure metric seemed to best predict the relative success of grasps by the produced numerical scores. In the case of the nozzle object, the two grasps with perfect experimental success had scores of > 0.3, while the two grasps with complete experimental failure had scores of < 0.1. On the pawn object, Robust Force Closure offered scores of > 0.45 on two of the more successful grasps and < 0.35 on two of the least successful grasps.

Gravity Resistance had reasonable success on the nozzle object, albeit with a razor-thin margin; the three successful grasps had scores < 5.96, while the failing grasps had scores > 6.0. On the pawn object, however, this metric struggled to produce meaningful results. The lowest score (and thus presumed best grasp) by this metric was Grasp 4, which had only a 20% success rate. The relative roundness of the pawn object forced many grasps to only tangentially touch the object, in a way that makes this gravity-based approach vulnerable to misclassifying good grasps as poorer ones.

Ferrari-Canny seemed to be the overall worst metric in our suite of tests. There was no discernible relationship between the scores produced by this metric and the success rate when tested on the Nozzle object; on the Pawn object, this metric correctly separated the successful Grasps 3 and 5 from the catastrophically bad Grasp 2, but misclassified each of Grasp 1 and 4.

One case deserving of special note is Grasp 1 on the pawn object; interestingly, this grasp was scored poorly by all metrics and yet had the highest experimental success. This example serves as a reminder that while there is a general association between metrics and experimental success, this relationship is by no means perfect. Beyond the weaknesses already mentioned, the single biggest problem of all of these metrics is that they fail to account for the 3D structure of the gripper and its approach vector. Pawn Grasp 4 is an excellent example of this, in which a seemingly good grasp is virtually unattainable without collision, thus leading to the observed low experimental success.

D. *Strengths and Limitations of Custom Grasp Planning Algorithm*

Our custom grasp planning algorithm uses an intelligent approach to identify the 'angle of attack' that produces the best margin around the gripper tool. By conducting a first-pass over all possible approach angles in the 'top hemisphere' (that is, skipping any angles that involve the gripper approaching from below), our algorithm first establishes an understanding of the total region of feasibility before returning the midpoint of the longest contiguous successful segment. This approach has the same asymptotic complexity as a naive approach of returning the first success found, but has a key advantage. By ensuring a 'buffer' around the angle chosen, we reduce the chance that the thickness of the gripper causes an unplanned collision with the object and ruins the grasp. Though we lack any real-world experimental results with our algorithm, we hypothesize that the most likely failure case is with an object designed to be held at the base. Such objects are naturally difficult to reach because of the fact that robot grippers are typically bulky, and as such cannot easily insert themselves low to the planar surface on which the object rests. However, our heuristic limiting the approach angles to the positive hemisphere would likely struggle on these cases.

IV. Bibliography

References

[1] Murray, Richard M., et al. A Mathematical Introduction to Robotic Manipulation. CRC, 1994.
[2] Mahler, Jeffrey. A Cloud-Based Network of 3D Objects for Robust Grasp Planning Using a Multi-Armed Bandit Model with Correlated Rewards. IEEE.
[3] Ferrari, Carlo, and John F. Canny. "Planning optimal grasps." ICRA. Vol. 3. 1992.

V. Appendix

A. *Future Improvements*

This assignment was the most straightforward of all the projects we had done, and the documentation is relatively concise. One thing that would be helpful if it was explained more is how to do the sampling for Ferrari-Canny. It was made clear after a comment from Jay in discord, but before that we didn't have a clue how to do it. This assignment was concise and helped us understand grasping from what was taught in class. There was little confusion in what was asked of us.

B. *Code Repository*

The GitHub Classroom team name for this report is Get A Grip. GitHub Link