Path Plan and Obstacle Avoidance

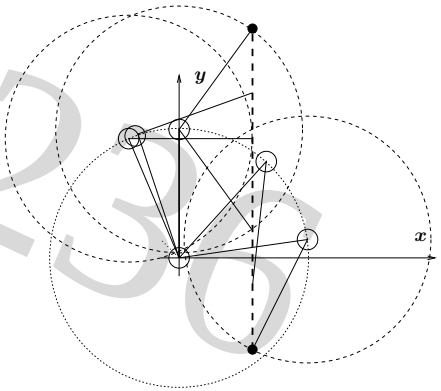
- Path Planning - determining the path of the end-effector in order to accomplish a given task.
- There may be different (intermediate) paths that accomplishes the same tasks. Use a performance measure to select which path to take.
  - Minimum energy
  - Minimum distance
  - Maximum manipulability, $\sqrt{\det(\boldsymbol{J}\boldsymbol{J}^{\boldsymbol{T}})}$
  - Minimum time

Path Plan and Obstacle Avoidance

- Obstacle avoidance is primary concern in path planning. End-effector as well as the rest of the manipulator should not interfere with the workspace.
- Efficient path planning and obstacle avoidance algorithms are important especially with online applications.

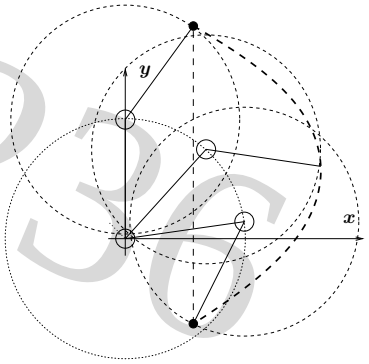
Basic Path Planning

- Given an initial point A and a final point B, move the end-effector in a straight line from A to B.
- Visually easier to plan.
- Possibility of locked joints, hard to control.



Basic Path Planning

- Hard to accomplish given that control is usually done in the joint space. Easier to implement joint space control.
- May take up larger amount of workspace, run into obstacles?
- Can use via points to force "straight" motion.



Obstacle Avoidance

- Potential functions are used to implement obstacle avoidance.
- A potential function gives an indication of how close the the manipulator is from object(s) in the workspace.
- To avoid obstacles, manipulator position is determined such that the potential functions minimum.
- Cost functions may also be defined in relation to obstacle distance, and solved for the minima for obstacle avoidance.

Cost Functions

- Cost function for one object.

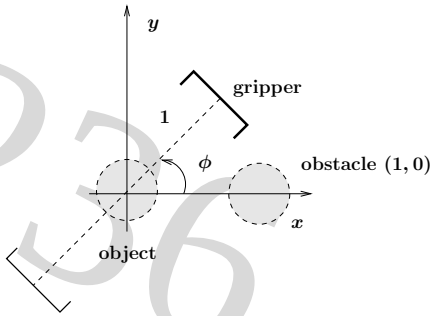
$$cost = \frac{1}{(\text{distance to object})^2} = \frac{1}{(p_x^m - p_x^{obj})^2 + (p_y^m - p_y^{obj})^2 + (p_z^m - p_z^{obj})^2}$$

- For multiple objects.

$$cost = \frac{W_1}{dist_{obj1}^2} + \frac{W_2}{dist_{obj2}^2} + \ldots$$

Cost Functions

- Example. Proper gripper orientation for grasping objects.
- Gripper centered on the object.
- Gripper fingers should avoid the obstacle.



Cost Functions

- Cost function considering only one finger.

$$cost = \frac{1}{(p_x^e - p_x^{obs})^2 + (p_y^e - p_y^{obs})^2} = \frac{1}{(\cos\phi - 1)^2 + (\sin\phi - 0)^2}$$

- Solution.

$$\underset{\phi}{\text{argmin}}\, cost = \pm\pi$$

Cost Functions

- Cost function with both fingers.

$$\begin{aligned} cost &= \frac{1}{dist_{grip1}^2} + \frac{1}{dist_{grip2}^2} \\ &= \frac{1}{(\cos\phi - 1)^2 + (\sin\phi)^2} \\ &\quad + \frac{1}{(-\cos\phi - 1)^2 + (-\sin\phi)^2} \end{aligned}$$

$$\operatorname*{argmin}_{\phi} cost = \pm\frac{\pi}{2}$$

Image Processing Basics

- What is an image?
  Matrix of values describing an object based on location.
- Color (rgb, cmyk, hsv), greyscale, bw.
- Concept of resolution, dpi, pixels, row/column pixels.
- Different storage formats (jpeg, bmp, raw).

Basic Image Operations

- Image is a bunch of numbers, what do you do with the numbers?
- Basic image operations?
- Loading an image - imread.
- Histogram - gives valuable information about pixel distribution.

Basic Image Operations

- Thresholding - converts image to "bw" given some threshold.
- Template convolution - mapping original image to a new image using weighted sum of neighboring pixels.

Other Helpful Image Operations

- **Averaging, median filter - reduce noise, low pass effect.**
- **Edge detection.**
- **Growing, shrinking objects.**

Image Acquisition and Processing System

- **Webcam - mplayer image capture.**
- **Octave or C image processing.**
- **Octave or C manipulator command generation.**
- **RVM1 execution.**