Credits to the information contained in this guide go to:

[1] Coulter, R. Craig. Implementation of the pure pursuit path tracking algorithm. No. CMU-RI-TR-92- CARNEGIE-MELLON UNIV PITTSBURGH PA ROBOTICS INST, 1992

[2] "Introduction to Intelligent System (430.457, Spring 2015) by Prof. Songhwai Oh.

Pure Pursuit Algorithm

The pure pursuit approach is a method of geometrically determining the curvature that will drive the vehicle to a chosen path point, termed the look-ahead point. An arc that joins the current robot position and the goal point is constructed. And chord length of this arc is the look-ahead distance. This length acts as the third constraint in determining a unique arc that joins the two points.

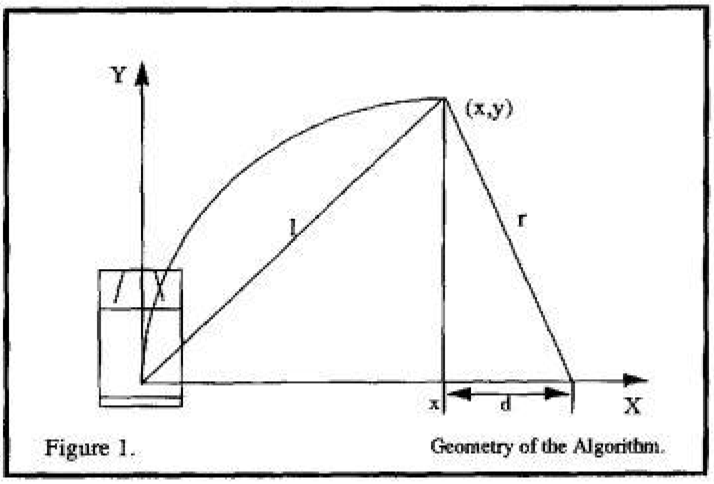

Figure 1

Consider Figure 1. The mobile robot is pictured, with the axes of its coordinate system drawn. The Y axis is mobile robot's heading. And point (x, y) which is one look-ahead distance l from the robot is one of points on the path. The following two equations hold. The first is from the geometry of the smaller right triangle in Figure 1. The second from the summing of line segments on the x axis. In the figure the length of x+d on the X axis and r on the hypotenuse of right triangle seems not be the same. But actually these are the same length r and constitute two sides of pie shape with arc. You can think point (x+d, 0) on the X axis as a pivot point of the arc.

$$x^2 + y^2 = l^2$$

$$x + d = r$$

Equation 1

The following equations relate the curvature of the arc to the look-ahead distance.

$$d = r - x$$

$$(r - x)^2 + y^2 = r^2$$

$$r^2 - 2rx + x^2 + y^2 = r^2$$

$$2rx = l^2$$

$$r = \frac{l^2}{2x}$$

$$\gamma = \frac{2x}{l^2}$$

Equation 2

Equation 2 is just a relation between x offset of the goal point from the robot's coordinate and look-ahead distance l. Using final equation, we can calculate curvature of the arc which joins robot's position and the look-ahead point. And this curvature can be used to determine a proportional constant between linear velocity and angular velocity as following equation.

$$w = \gamma v$$

Equation 3

In implementation, the positions of robot and look-ahead point are set according to global coordinate. So, first, you have to transform them from global coordinate frame to relative coordinate frame in which the robot position becomes its origin. After the transformation, you can simply compute linear and angular velocity from the above equation.

Heuristics

Simply implementing pure pursuit algorithm is not sufficient to control mobile robot move smoothly. Sometimes, this algorithm shows poor performance generating large wave like trajectory as Figure 2. This phenomenon occurs when the heading of robot at current points deviates large from the look ahead direction. And also occurs at the point where the path bent abruptly. So we have to add some heuristics to handle this problem.

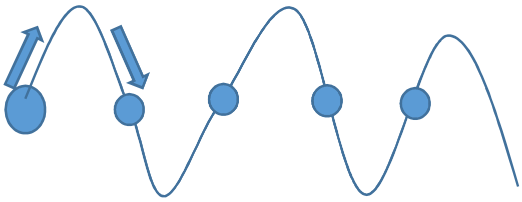

Figure 2

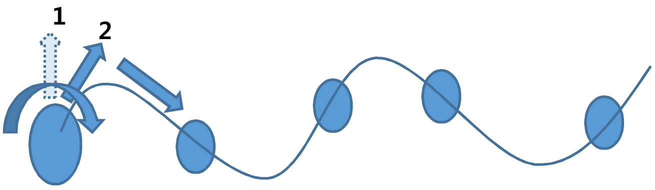

Figure 3

Pure pursuit works inefficiently when the deviation of heading of a robot from the look-ahead direction is too large. In that case, it would be better to turn sharply toward the look-ahead point at a current point. This procedure is illustrated in Figure 3. Parameters you have to design are threshold of angle difference above which the robot stops and turn to look-ahead direction and its angular velocity. And also you have to design robot's linear velocity when following the pure pursuit. Consider constraints when you design parameters. Whole procedure is summarized in algorithm 1.

Algorithm 1

1. Compute the angle difference between robot's heading and look-ahead point direction.

2. If the angle difference is larger than threshold1,

3. Turn fast toward look-ahead point direction.

4. If the angle difference is larger than threshold2 (< threshold1),

5. Turn slowly toward look-ahead point direction

6. Otherwise, do pure pursuit.