1. Solution to problem 1 as belows:

① Homework 1 Problem 1

ⓐ The cart always points forward.
  (Steering angle cannot be changed)
  we have $\dfrac{dy}{dx} = \tan\theta$

  $\Rightarrow \dfrac{\dot{y}}{\dot{x}} = \dfrac{\sin\theta}{\cos\theta}$

  $\Rightarrow \left[\dot{x}\sin\theta - \dot{y}\cos\theta = 0 \right]$ —— ①

  This is a Pfaffian constraint.

  —— × ——

ⓑ Configuration transition equations can be
  written as,

  $\dot{\theta} = 0$ ∵ Cannot be changed — Constraint
  $\dot{x} = u_s \cos(\theta)$
  $\dot{y} = u_s \sin(\theta)$

  —— × ——

ⓒ From equation ①
  $\dot{x}\sin\theta - \dot{y}\cos\theta = 0$

  Integrating the above equation,

  $\displaystyle\int \dfrac{dx}{dt}\sin\theta \, dt - \int \dfrac{dy}{dt}\cos\theta \, dt = 0$
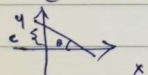
②

$$\Rightarrow \int \sin\theta \, dx - \int \cos\theta \, dy = 0$$

$$\Rightarrow \quad x\sin\theta - y\cos\theta + C = 0$$

This is holonomic because it can be expressed as $f(q,t) = 0$.

———x———

ⓓ $y\cos\theta = x\sin\theta + C$

$$y = x\tan\theta + \frac{C}{\cos\theta}$$

This is of the form $y = mx + C$ which is an equation of a straight line with slope m & 'c' as the intercept on y axis. →

∴ We can reduce the above equation to be a straight line with slope '$\tan\theta$' making θ with +ve x axis & $(C/\cos\theta)$ intercept on y axis.

2. The inflated object in the C space and the minimum length path can be seen as follows.
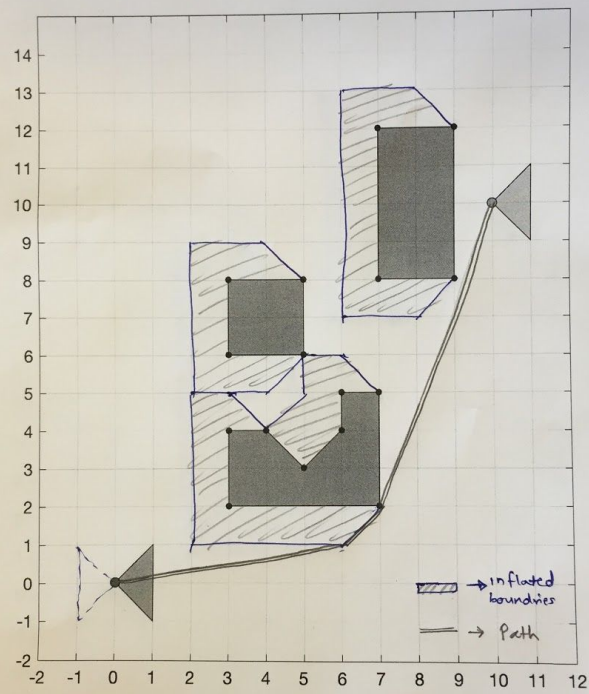

Figure 1: Workspace for a translating triangular robot. The start position is shown in red and the goal position is shown in green. The three workspace obstacles are shown in blue.

3. Code attached: run python rrt.py . should read input_maze.txt (Not working , hardcoded the values) , creates obstacles, builds the Tree in real time, finds a path. Comments inline.

The tree and the path is shown as below:
Number of Nodes < 11000
Attached files: rrt.py, RRT_includes.py (Some include stuff) input_maze.txt
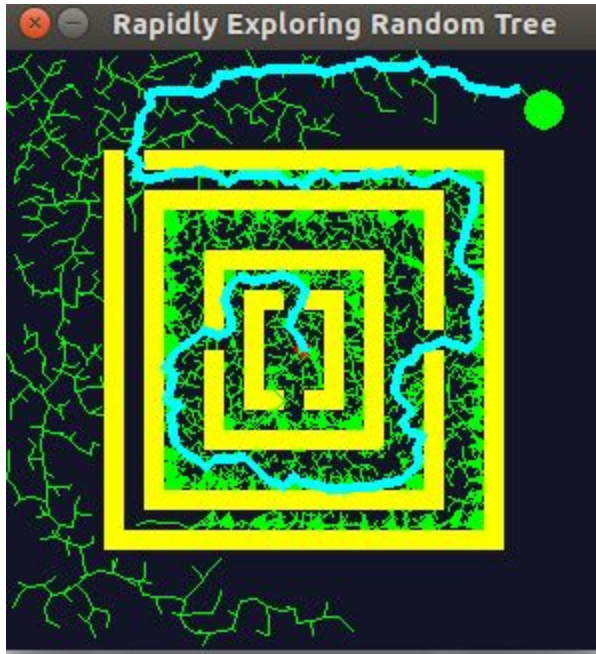
Created input_triangle.txt from problem 2 and is attached too. Following is the path and the Tree when RRT is run on the problem 2 configuration space.

Number of nodes < 200
Attached files: rrt_2.py, RRT_includes.py (Some include stuff) input_triangle.txt