*Lab3 Path Planning with RRTs*

*Method*

*We applied RRTs within configuration space, for the reason that there will be no singularity, conflict and unsolvable points on the path when operating path planning in C-space.*

*Pseudo code*

T\_start {} // store the the q within configuration space

Lines{} // store the index the q

Loapmap(‘map’);

Get\_q(start, end) //

***Loop***: for 10000 times

***If*** random number > probability

q\_goal: move toward q\_goal

***Else***

q\_rand: Choose random point within joint limit

q\_close: Choose random and store the index in lines

q\_new: move a distance toward q\_rand from q\_close

***if*** q\_new within joint limit

check collision with joint 1 – joint 6

convert q\_new into workspace by Forward kinematics

iscollided(point1, point2).

Check collision of line between q\_new and q\_close

Separate the line into multiple points

Check collision with iscollided lines by lines

***if*** line between q\_new and q\_close

put into T\_start

store index of q\_new and q\_close as index of line

***if*** not

jump out;

***if*** goal on the line of q\_new and q\_close or goal == new

q\_new: // q\_goal as q\_new

path: index of lines linked from q\_end to q\_start

stack: store the via points with index in path within C-space

***end***

Based on the statistics of our experiments, the rate of success is around 60%.

The amount of time and final path in each experiments taken by calculation are different, due to, first, different environmental setting causing different computational complexity to our program; second, the random procedure also played a role in this phenomenon.

The computational complexity will rise exponentially when the environment changes. So, the time for calculation will increase and the rate of success will drop. If the environment becomes too complicated, there might be no solution. If not, the path is still solvable for very long time.

We got different results and different operation time, as running the planner with same destinations for multiple times. In that, the random sampling caused different path and results. Sometimes, it might fail in the positions which can be solved previously.

***Analysis***

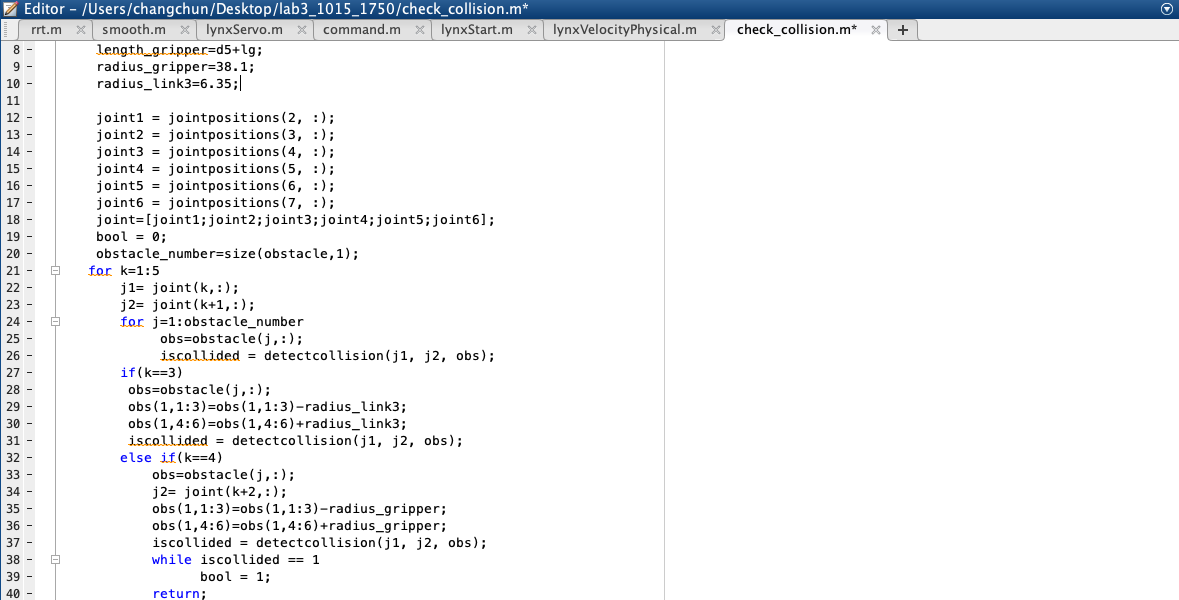
We would try to implement the high efficient way of getting configurations of q\_start and q\_goal. Due to inherent limitation of lynx robot arm, the position of end effector can only be on the plan constructed by the first three links. Therefore, it is hard to get the exact wanted position.

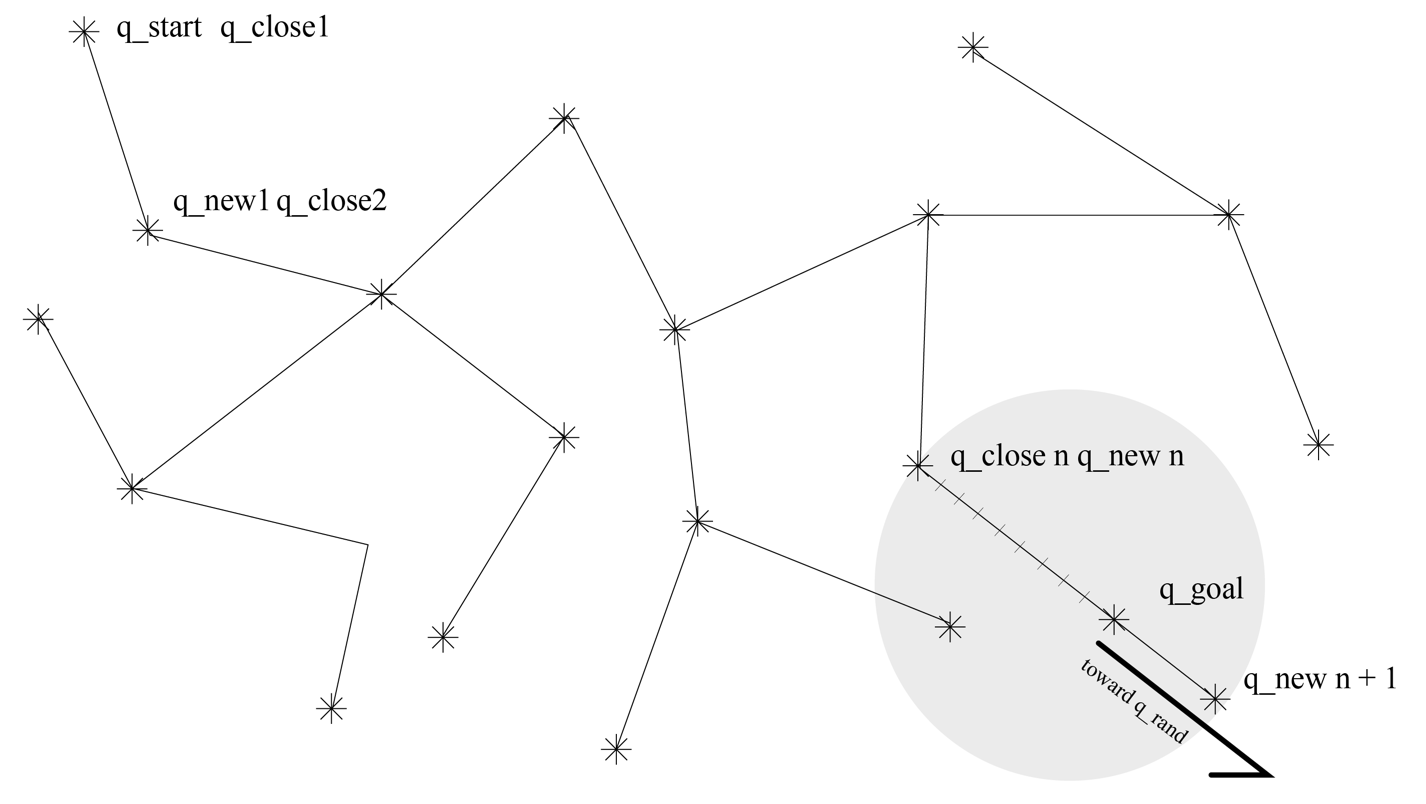
In this lab experiments, we implemented both way: by inverse kinematics with random configurations and given position to get first and final configurations; or by mapping position of end effector by using forward kinematics with random configurations of four joints, and they took lots of time if the constraints was tight.

Second, we would try different method of sampling such as KD-Trees. In that, although we gave the planner a force from start point to goal point with probability 30% to increase the efficiency. Each experiments still took only 10% of sampling as its via-points.

Third, we would implement some ways to smooth the paths. The end effector moved only a small portion due to the logic of the RRTs. So, sometimes planner may swirl in an area before moving toward to the end point. This movement wasted time and energy. It would be more power and time efficient if the smoothing path can be applied.

Summary: We applied RRTs within configuration space, because this way guarantees the paths with no unsolvable points and singularity. However, as a tradeoff, the rate of success in simulation and speed of calculation cannot compete with RRTs with workspace. But, we notices that RRTs within workspace may not always succeed without calibration in real world, due to error accumulation caused by factors such as gravity when using inverse kinematics. On the other hand, RRTs within C-space is more reliable because of the higher accuracy of joint out from configuration space. Once the simulation is generated, the path in real world is good at avoidance the obstacles on the environment.



**