

1 Configuration Space Path Planning

- Reference: 1) A Simple Motion Planning Algorithm for General Purpose Manipulators by T. Lozano-Perez, 2) Siegwart, section 6.2.1
- Fast, simple to implement
- Can handle multiple DOF Robots, complicated non-convex obstacles
- Can plan motion in cluttered environments
- A key component of task level programming “Plan Path (collision free) from start to goal”
- Use an approximate, discrete, quantized approach to reduce mathematical complexity. No attempt to model object surfaces
- Configuration: Set of parameters that completely specify the position of an object. We can use joint space for N-DOF Robot as its configuration. Cartesian not unique
- C-Space: Set of all possible configurations
- Key idea for Path Planning: Map obstacles into the robot’s C-Space. Creates regions of C-Space that contain obstacles and free-space. Then plan path in C-Space.
- Problem: Joint Space can still be high dimensional.
- Solution: Use “slices” of C-Space for path planning. Projections of N-DOF space into set of N-1 Dimensional Projections.
- 2-D C-Space creation and Path Planning is simple. Represent 3-D as 2-D slices and plan in these spaces.
- Basic Idea: Determine range of legal joint values for a joint parameter given ranges of the previous joints.
- Advantages: Planning becomes simple search in 1 and 2-D space. Disadvantages: Loss of accuracy (granularity of projection), increased storage as DOF increases.
- Tradeoff: Simple vs. intersecting high DOF manifolds of objects and manipulators.

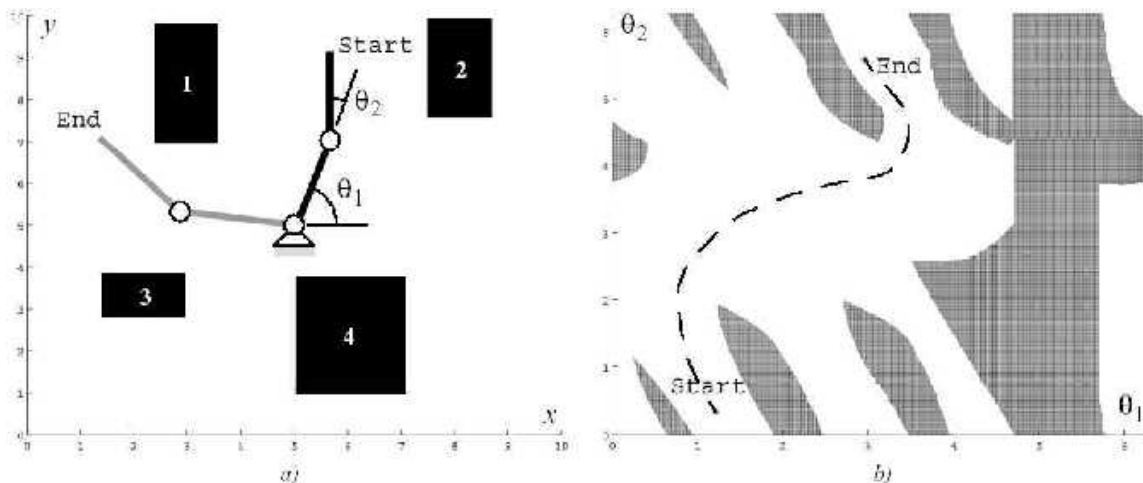


Figure 1: Configuration Space for Two Link Manipulator with obstacles

2 Configuration Space Path Planning Examples

Below are some simple examples of how Configuration Space (CSPACE) path planning works. We will be using a two-link, planar, pick-and-place manipulator as in figure 1. This design allows us to avoid computing arm collisions with the obstacles - we just have to worry about the gripper (assumed to be a point) moving into an obstacle.

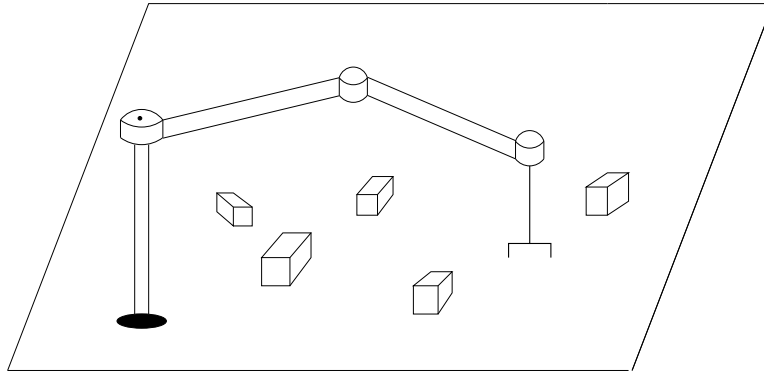


Figure 2: Two Link Manipulator used in examples

We give the system Cartesian start and goal positions. Each Cartesian position has 2 inverse kinematic solutions, so the path planner can plan 4 distinct paths from each distinct start and goal position. A set of known obstacles is given as part of the environment. Here is the algorithm:

1. Partition each of the robot's joints into discrete regions, say every 5 degrees. This creates a 72×72 discrete angular grid.
2. Iterate over all combinations of joint 1 and joint 2 (we use the center of each 5 degree range as the test point). If the forward kinematics for these joint angles lies inside an obstacle, we classify this part of the 72×72 grid as *forbidden*, otherwise it is a legal configuration.
3. Given a Cartesian start and end position, do the following:
 - (a) Using Inverse Kinematics, find the solutions to (θ_1, θ_2) for the start and goal position. There will be 2 solutions for each position.
 - (b) Choosing 1 of the 4 possible solutions, perform a breadth-first search in the joint-space grid from $(\theta_{1start}, \theta_{2start})$ to $(\theta_{1goal}, \theta_{2goal})$. Mark the path.

We will use a simple Breadth First Search in Configuration Space of the manipulator to find a path in free space between start and goal positions represented as joint configurations. Pseudo-code below.

```

FINDPATH( $\theta_{1start}, \theta_{2start}, \theta_{1goal}, \theta_{2goal}$ )
if(( $\theta_{1start}, \theta_{2start}$ )  $\equiv$  ( $\theta_{1goal}, \theta_{2goal}$ )) then stop-GOAL=FOUND
ADD(( $\theta_{1start}, \theta_{2start}$ )) to QUEUE and mark as VISITED
While QUEUE NOT EMPTY and GOAL NOT FOUND do
  POP top item in QUEUE and assign to ( $\theta_1, \theta_2$ )
  if(( $\theta_1, \theta_2$ )  $\equiv$  ( $\theta_{1goal}, \theta_{2goal}$ )) then GOAL=FOUND
  else ADD to QUEUE each Free Space UNVISITED 4-neighbor of ( $\theta_1, \theta_2$ )
  Mark each neighbor added to QUEUE as VISITED
  and Remember which node Opened this node (its predecessor, ( $\theta_1, \theta_2$ ))
If GOAL==FOUND
Recreate path by following predecessor chain from ( $\theta_{1start}, \theta_{2start}$ ) to ( $\theta_{1goal}, \theta_{2goal}$ )
  
```

There are 4 examples that follow showing the joint-space and Cartesian-space paths for the same Cartesian start and goal positions using each of the 4 possible inverse kinematic solutions. In each example, the manipulator has link lengths $L_1 = 5$ and $L_2 = 3$.

In the joint-space diagrams, the path from start to goal is shown with asterisks. In the Cartesian-space plots, the path and obstacles are shown along with the manipulator links superimposed at every 5th point on the path.

```

      THETA1 THETA2
start  53   127
goal   294   58

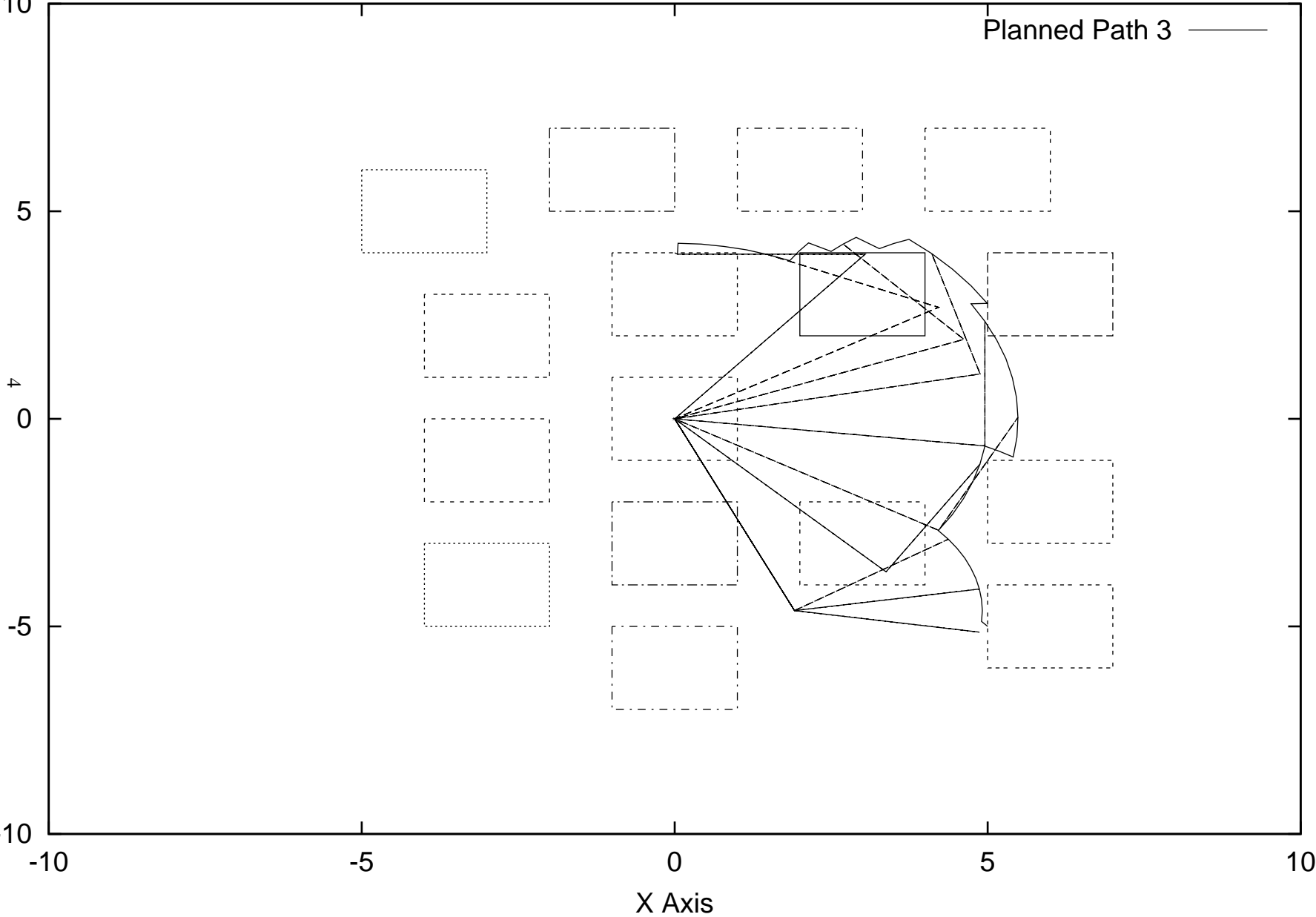
```

```

Theta 1:
      Theta2
0 5      11111111 * 11111111      11111 11111111
5 10     11111111 * 1111111111      11 11111111
10 15    1111111 *111111111111      11111111
15 20    111      ***11111111      1111111
20 25    1      ***111      1
25 30    1111111      ***
30 35    11111111 111      *      1
35 40    11111111 111111      * 111      11111111
40 45    11111111 1111111111      *11111111      11111111 1
45 50    1111111 1111111111      *11111111      11111111 1111
50 55    1111 111111111      *4111111111      1111111111 111111
55 60    1 11 11 11111111      11111111 11111111 11111111
60 65    11111 1111111111      1111 1111111111
65 70    1111111111 1111111111      111 1111111111
70 75    111111111 11111111      11111111 1111111
75 80    11111111 111111      111111111111 11
80 85    111 11111111      1111111111 1
85 90    111 111 11111111      111111 1111
90 95    11111111 11 111 1111 11111 11111 1111111111
95 100   111111111 1111 111111 11 1111111111
100 105  111111111 11111111 111111 1111111111
105 110  11111111 1111111111 11111111 11111111
110 115  111111111 111111111 11111111111 1
115 120  1 111111111 111111111 1111111111 1111
120 125  111111111 111111111 1111111111 111111111
125 130  11 111 1111111116 1111111111
130 135  111111111 11111111
135 140  11111111 111111
140 145  111 1 11 111
145 150  11111 11111111
150 155  111111 111111111
155 160  11111111 111111111
160 165  111111111 1 11111111
165 170  11111111111 1111 111
170 175  11111111111 1111111111
175 180  111111111 1111111111
180 185  1 11111111 1111111111
185 190  1111 1111 1 111111111
190 195  11111111 1 1111111
195 200  1111111111 11 1111
200 205  1111111111 11
205 210  111111 111
210 215  11 1111
215 220  111 111111111
220 225  111111111 111111111
225 230  11 11111111 111111111
230 235  11111 111111111 11111111
235 240  111111111 111111111 11111111
240 245  111111111 111111111 11111111
245 250  11111111 1111111111 111111
250 255  111 11111111 111 1
255 260  11111 11111 11111
260 265  11 11111 111111111
265 270  11111 1111 111 111111111
270 275  1111111 111 1111 111111111
275 280  111111111 11111 1111111
280 285  11111111111 111111
285 290  1111111 1111111 111
290 295  3***** 111 111111111 111111111
295 300  11111111 * 111111111 111111111
300 305  11111111 111 * 111111111 111111111
305 310  11111111 111111 * 111111111 111111
310 315  11111111 111111111* 11111111 11 1
315 320  111111 111111111*** 11111111 1111
320 325  1111 11111111 * 111 111111
325 330  1 11 * 111111111
330 335  * 111111111
335 340  * 111 51111111
340 345  * 11111111 111111
345 350  * 11111111111 11
350 355  11111* 11 111111111 1
355 360  11111111** 11111 11111111 1111
Theta 1 0 20 40 60 80 100 120 140 160 180 200 220 240 260 280 300 320 340
      Theta 2

```

Cartesian Start:(0,4.1) Goal:(4.9,-5): (q1start,q2start)=(53,127) (q1goal,q2goal) = (294, 58)

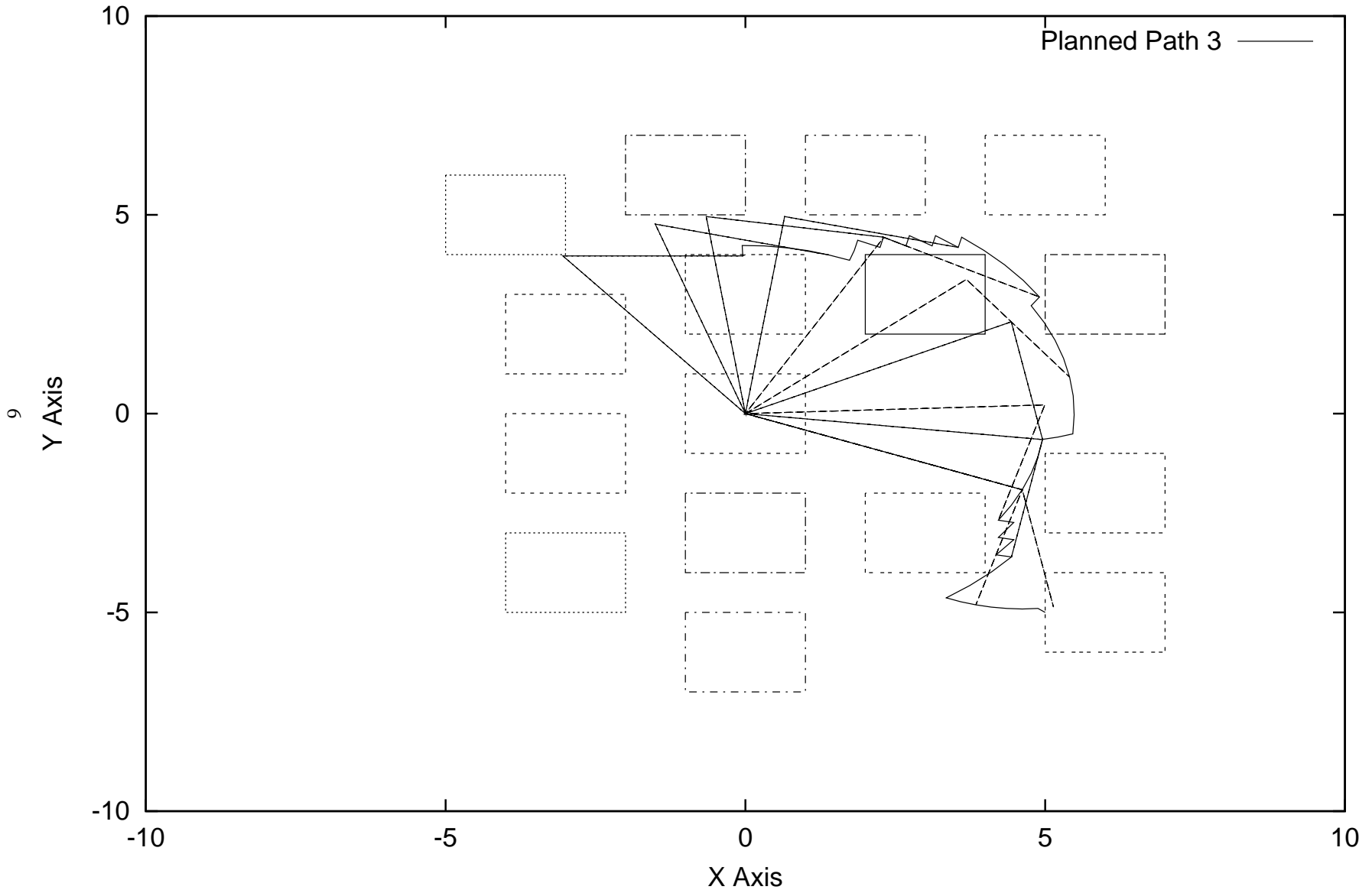


```

      THETA1 THETA2
start 127 233
goal  336 302
Theta 1:
  0  5      11111111  1111111  11111** 11111111
  5 10      11111111  111111111  11 * 11111111
 10 15      1111111  111111111111  * 11111111
 15 20      111  1111111  *111111
 20 25      1  111  *1
 25 30      1111111  ***
 30 35      11111111  111  * 1
 35 40      11111111  111111  111  * 11111111
 40 45      111111111  111111111  11111111  * 11111111  1
 45 50      111111  111111111  11111111  * 11111111  1111
 50 55      1111  11111111  411111111  *111111111  111111
 55 60      1  11  11  111111111  *1111111  111111111
 60 65      11111  111111111  **1111  111111111
 65 70      111111111  11111111111  111  * 11111111
 70 75      111111111  11111111  11111111  * 111111
 75 80      11111111  11111  111111111111* 11
 80 85      111  11111  1111111111** 1
 85 90      1111 111 111 1111 111111** 1111
 90 95      11111111  11 111 1111 11111** 11111111
 95 100     11111111  1111 11111 11** 11111111
100 105     111111111  11111111  111111  *** 11111111
105 110     1111111  1111111111  1111111  * 111111
110 115     11111111  11111111  1111111111* 1
115 120     1  11111111  111111111* 1111
120 125     11111111  111111111* 11111111
125 130     11 111 1111111116* 111111111
130 135     11111111  11111111
135 140     11111111  11111
140 145     111  111  1  11 111
145 150     11111  11111111
150 155     111111  111111111
155 160     11111111  11111111
160 165     111111111  1 1111111
165 170     1111111111  1111 111
170 175     11111111111  111111111
175 180     1111111111  11111111111
180 185     1 1111111 1111111111
185 190     1111 1111 1 11111111
190 195     11111111  1 111111
195 200     1111111111  11 1111
200 205     1111111111  11
205 210     11111 111
210 215     11 1111
215 220     111 11111111
220 225     11111111  11111111
225 230     11 11111111  11111111
230 235     11111 111111111  11111111
235 240     11111111  11111111  11111111
240 245     111111111  11111111  11111111
245 250     11111111  1111111111  11111
250 255     111 1111111  111 1
255 260     11111 11111
260 265     11 11111 11111111
265 270     11111 1111 111 11111111
270 275     11111111  111 1111 11111111
275 280     11111111  11111 111111
280 285     1111111111  11111
285 290     1111111  1111111  111
290 295     3 111 111111111  11111111
295 300     1111111  11111111  11111111
300 305     11111111  111 11111111  11111111
305 310     11111111  111111  11111111  11111
310 315     11111111  111111111  11111111  11 1
315 320     111111  111111111  11111111  1111
320 325     1111 11111111  111 111111
325 330     1 11 11111111
330 335     11111111
335 340     111 *****51111111
340 345     1111111  * 111111
345 350     11111111111* 11
350 355     11111 11 11111111** 1
355 360     11111111  11111 1111111** 1111
      0  20  40  60  80 100 120 140 160 180 200 220 240 260 280 300 320 340
      Theta 2

```

Cartesian Start:(0,4.1) Goal:(4.9,-5): (q1start,q2start)=(127,233) (q1goal,q2goal) = 336,302



```

THETA1 THETA2
start 53 127
goal 336 302

```

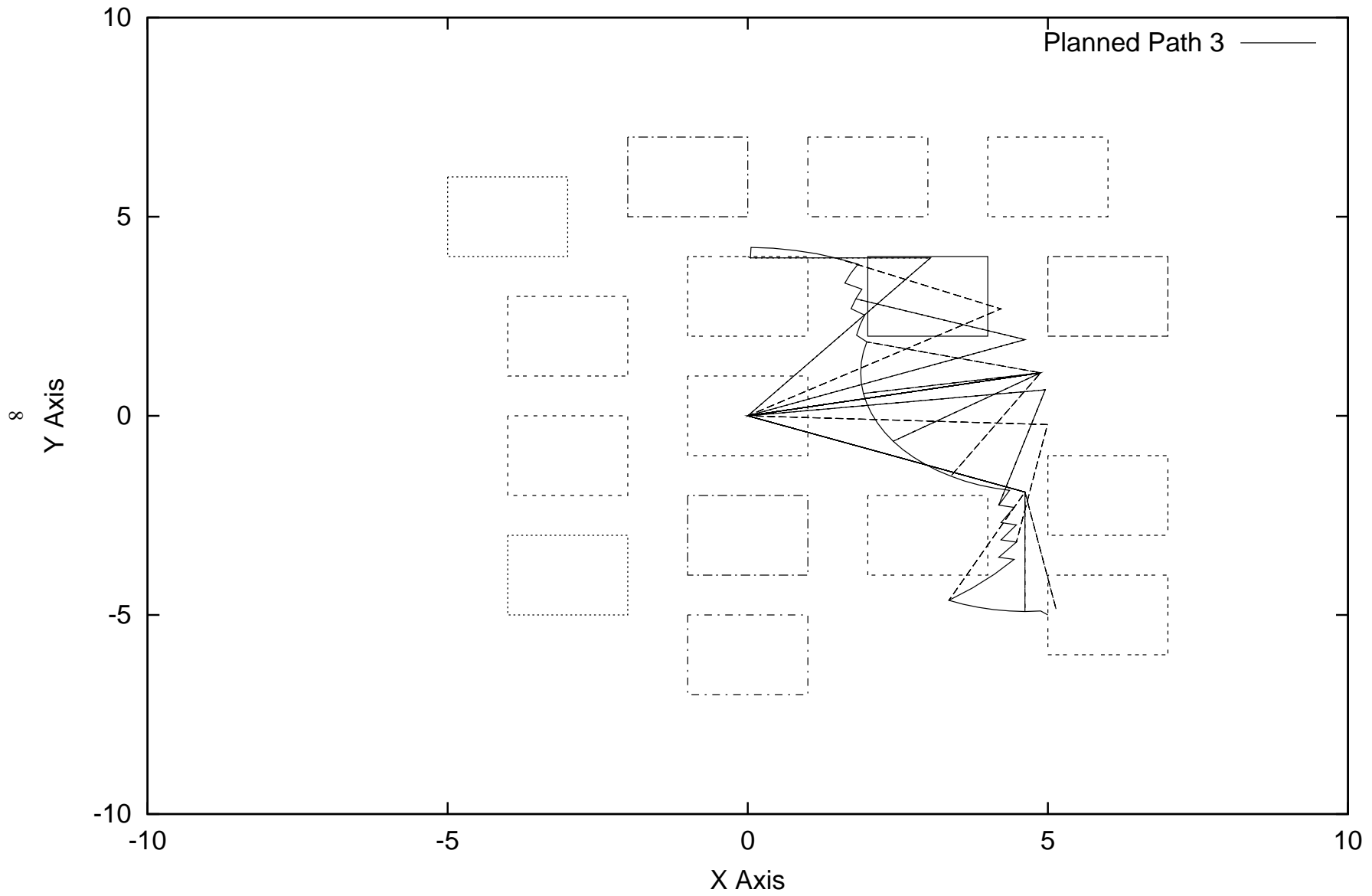


```

THETA1 THETA2

```

Cartesian Start:(0,4.1) Goal:(4.9,-5): (q1start,q2start)=(53,127) (q1goal,q2goal) = 336,302




```

start 127 233
goal 294 57
Theta 2
0 5 11111111 11111111 11111 11111111
5 10 11111111 11111111 11 11111111
10 15 1111111 11111111111 111111111
15 20 111 11111111 1111111
20 25 1 111 1 11111111
25 30 11111111
30 35 11111111 111 1
35 40 11111111 1111111 111 11111111
40 45 11111111 111111111 11111111 11111111 1
45 50 1111111 111111111 11111111 11111111 1111
50 55 1111 11111111 411111111 111111111 1111111
55 60 1 11 11 111111111 1111111 111111111
60 65 11111 111111111 11111111 1111 1111111111
65 70 111111111 1111111111 111 11111111
70 75 111111111 1111111 1111111 1111111
75 80 11111111 11111 11111111111 11
80 85 111 1111 11111111 1
85 90 1111 111 1111111 1111
90 95 1111111 11 111 1111 11111 111111111
95 100 11111111 1111 11111 11 11111111
100 105 11111111 1111111 11111 11111111
105 110 1111111 111111111 1111111 1111111
110 115 11111111 11111111 1111111111 1
115 120 1 11111111 111111111 1111
120 125 11111111 11111111 111111111
125 130 11 111 111111116* 111111111
130 135 11111111* 1111111
135 140 11111111* 11111
140 145 111 * 1 11 111
145 150 11111 * 11111111
150 155 11111 * 11111111
155 160 11111111 ***** 11111111
160 165 111111111 ***1 11111111
165 170 1111111111 *****1111 111
170 175 1111111111 *111111111
175 180 1111111111 * 1111111111
180 185 1 1111111111111111111111
185 190 1111 1111 *1 111111111
190 195 11111111 * 1 1111111
195 200 11111111111 * 11 1111
200 205 11111111111 * 11
205 210 11111 * 111
210 215 11 ***** 1111
215 220 ***111 111111111
220 225 *11111111 11111111
225 230 11 *11111111 11111111
230 235 11111 *11111111 11111111
235 240 11111111 *11111111 11111111
240 245 111111111 *11111111 11111111
245 250 11111111 *111111111 11111
250 255 111 * 1111111 111 1
255 260 *** 11111 11111
260 265 **11 11111 111111111
265 270 **11111 1111 111 111111111
270 275 **1111111 111 1111 111111111
275 280 **111111111 11111 1111111
280 285 *11111111111 11111
285 290 * 1111111 1111111 111
290 295 3***** 111 111111111 11111111
295 300 1111111 11111111 111111111
300 305 11111111 111 11111111 11111111
305 310 11111111 1111111 111111111 11111
310 315 11111111 111111111 1111111 11
315 320 111111 111111111 1111111 1111
320 325 1111 11111111 111 111111
325 330 1 11 11111111
330 335 111111111
335 340 111 51111111
340 345 1111111 111111
345 350 1111111111 11
350 355 11111 11 11111111 1
355 360 11111111 11111 111111 1111
0 20 40 60 80 100 120 140 160 180 200 220 240 260 280 300 320 340
Theta 2

```

Cartesian Start:(0,4.1) Goal:(4.9,-5): (q1start,q2start)=(127,233) (q1goal,q2goal) = 294, 57)

