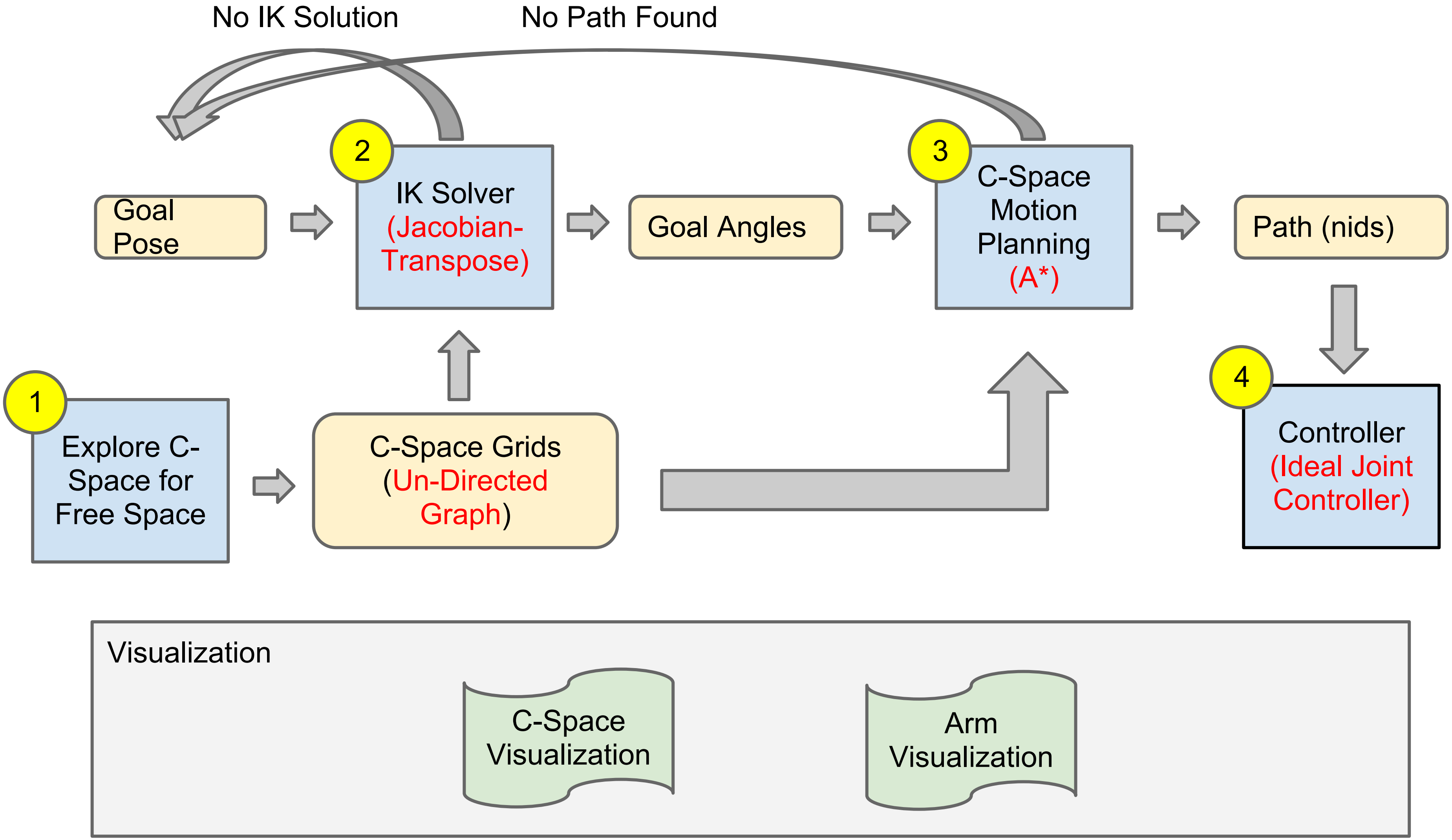
Winter 2012 CSE599J
Personal Robotics Clinic:
Algorithms and Applications

Arm Motion Planning
Final Presentation

Liang-Ting Jiang

Arm Motion Planning Framework



No IK Solution

No Path Found

Goal Pose

2 IK Solver (Jacobian-Transpose)

Goal Angles

3 C-Space Motion Planning (A*)

Path (nids)

1 Explore C-Space for Free Space

C-Space Grids (Un-Directed Graph)

4 Controller (Ideal Joint Controller)

Visualization

C-Space Visualization

Arm Visualization

1

Explore C-Space

(Unit: degree)

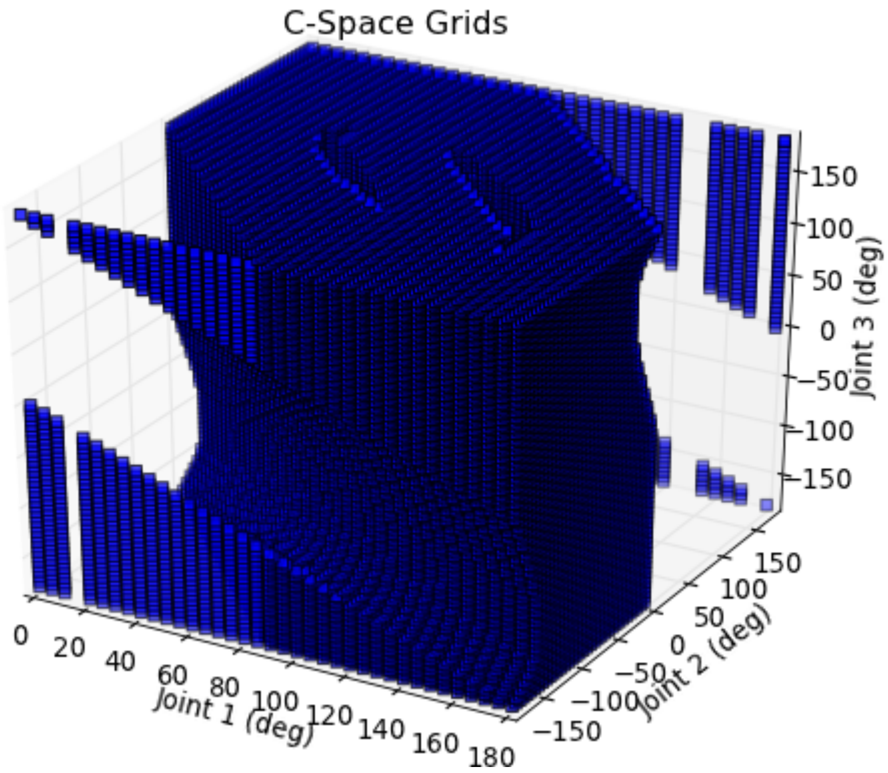
Grid Size = 5

Joint1: x (0 ~ 180)

Joint2: y (-180 ~ 180)

Joint3: z (-180 ~ 180)

```
for each (x,y,z)
{
    ForwardKinematics(x,y,z)
    CheckCollision()
}
```

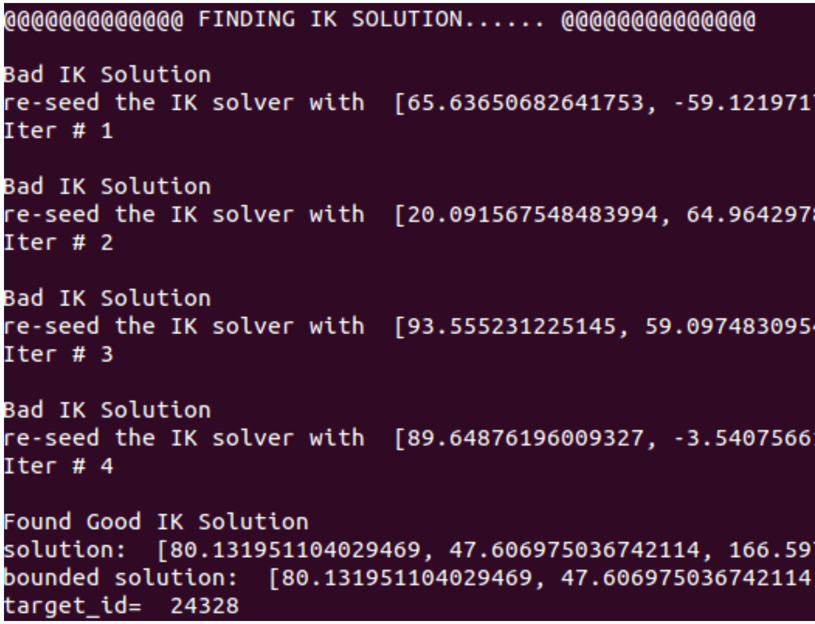


2

IK Solver (Jacobian-Transpose)

(MAX_ITER = 20)

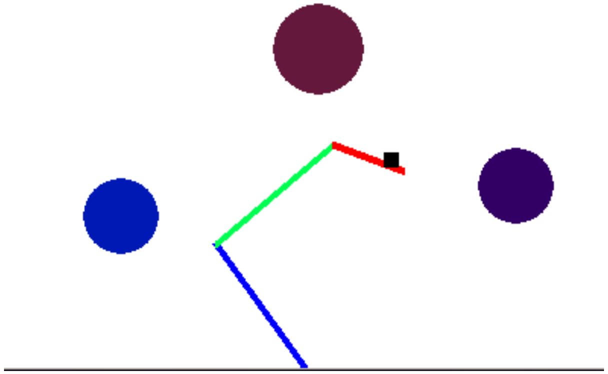
```
IterativeJTransIK()
{
    SeedCurrentAngles()
    while count < MAX_ITER
    {
        JTransIK()
        if Collision():
            SeedRandom()S //bad solution
        else:
            return //valid solution
    }

    return //no solution
}
```

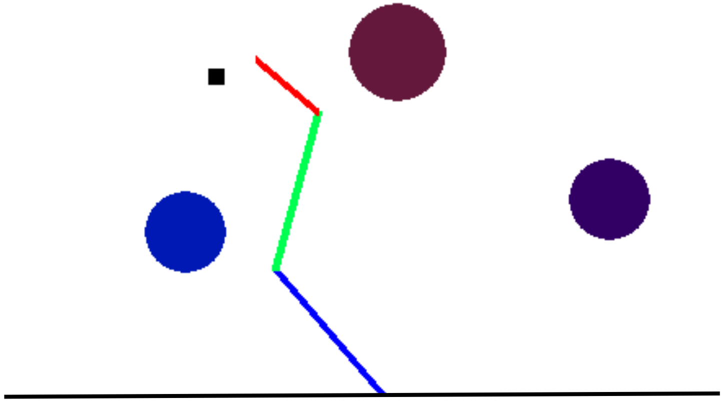


```
@@@@@@@@@@@@@ FINDING IK SOLUTION...... @@@@@@@@@@@@@@

Bad IK Solution
re-seed the IK solver with  [65.63650682641753, -59.121971
Iter # 1

Bad IK Solution
re-seed the IK solver with  [20.091567548483994, 64.964297
Iter # 2

Bad IK Solution
re-seed the IK solver with  [93.555231225145, 59.097483095
Iter # 3

Bad IK Solution
re-seed the IK solver with  [89.64876196009327, -3.5407566
Iter # 4

Found Good IK Solution
solution:  [80.131951104029469, 47.606975036742114, 166.59
bounded solution:  [80.131951104029469, 47.606975036742114
target_id=  24328
```

3

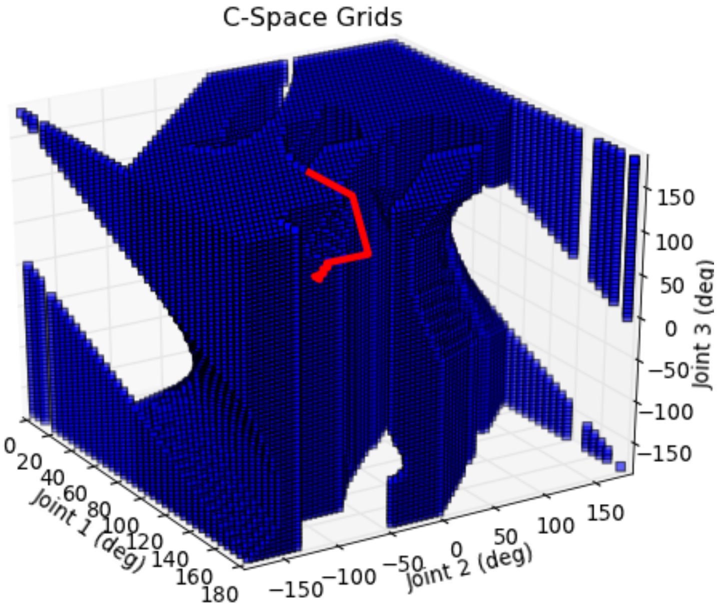
C-Space Motion Planning (A*)





- Search in the 'Free' C-Space
- Path consists of node IDs

path = [start, nid1, nid2, ..., nidX, goal]



4

Controller (Ideal Joint Controller)

- Move the Joints according to the planned Path
- Node ID -> [angle1, angle2, angle3]

DEMO

Issues:

- Slow C-Space Exploration with fine grid size
- Conservative and slow collision checking (draw->check pixels)
- End pose error: discretization of C-Space

Solution:

- C-Space Grids: OctoMap (http://octomap.sourceforge.net/)
- Better collision checking algorithm