Transformations in Detail

Algorithms and Data Structures 2 – Motion Planning and its applications

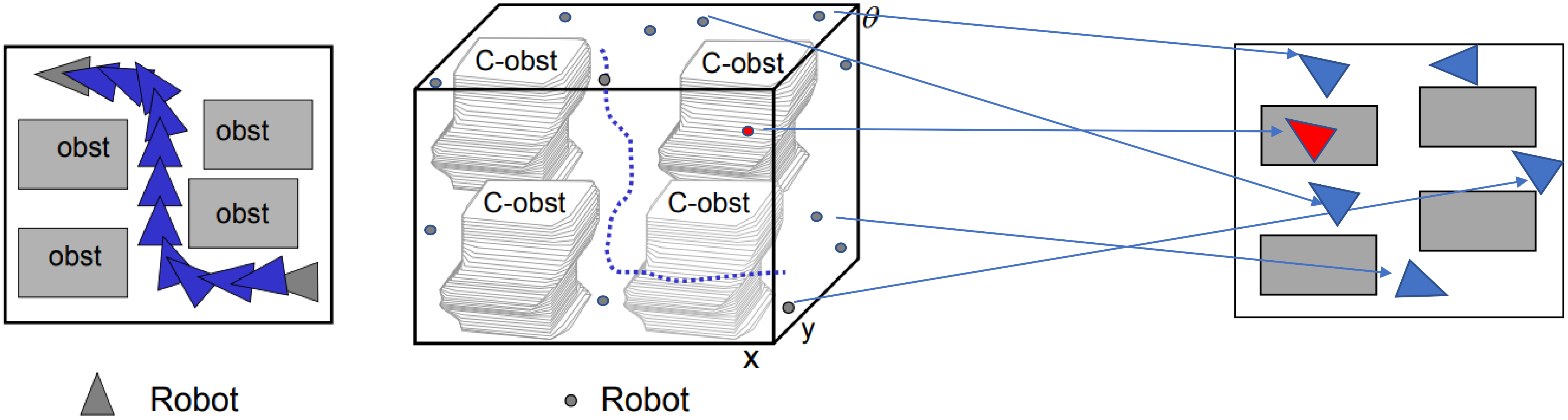
University of Applied Sciences Stuttgart

Dr. Daniel Schneider

Interpolation in the workspace

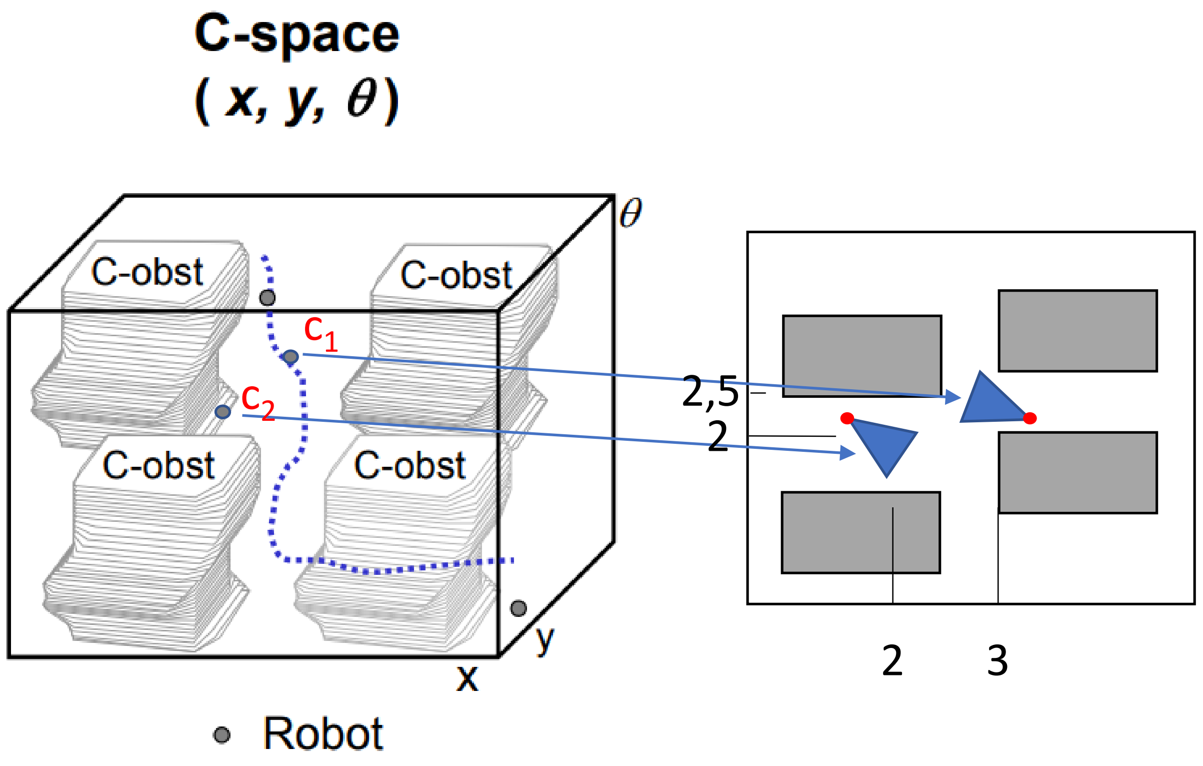
Workspace **(x, y)**

C-space **(x, y, θ)**



Sources:
Configuration Space Configuration Space for Motion Planning– Prof. Seth Teller – http://courses.csail.mit.edu/6.141/spring2010/pub/lectures/Lec10-ConfigurationSpace.pdf

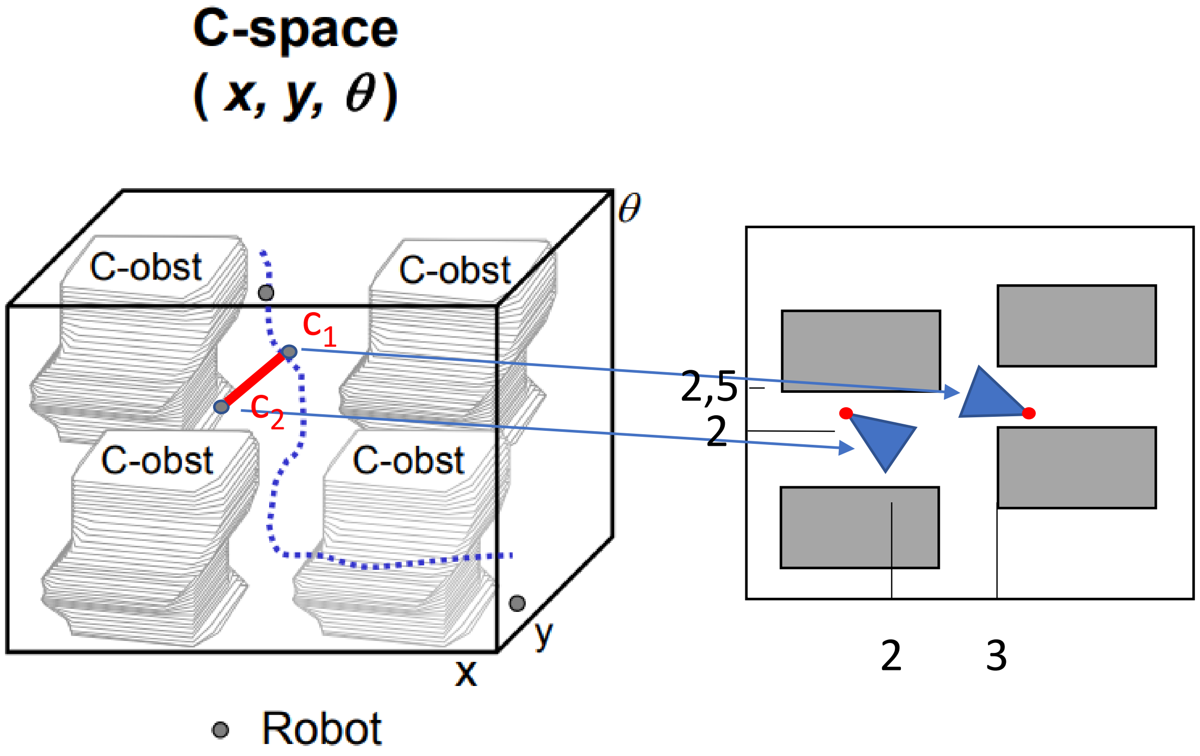
Interpolation in the workspace



C_1=(2/2/135°)

C_2=(3/2,5/0°)

How to “draw the line”

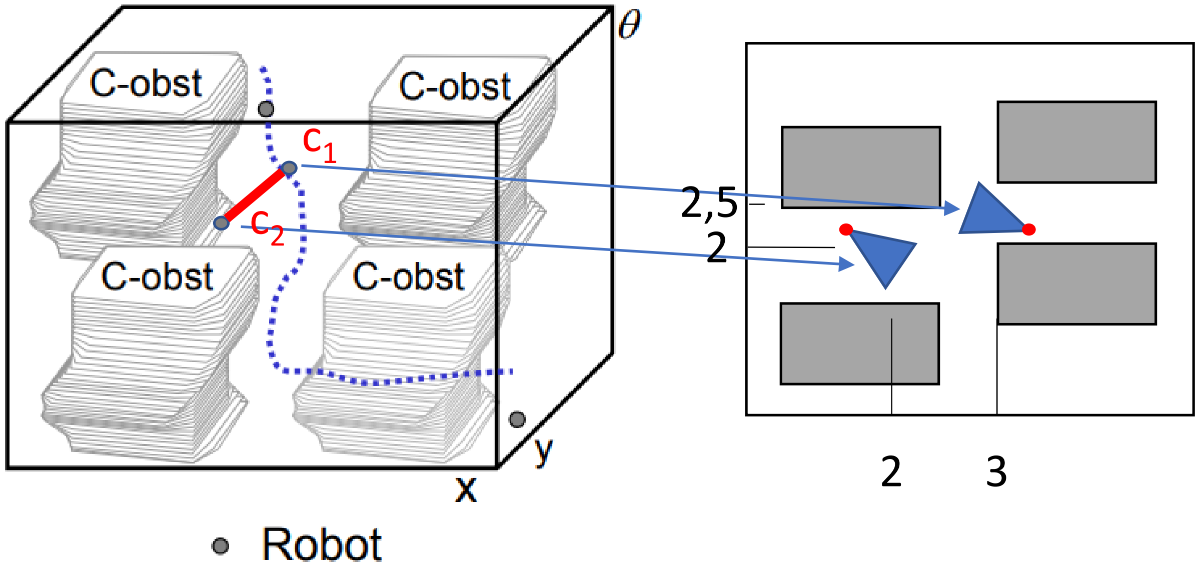


C_1=(2/2/135°)

C_2=(3/2,5/0°)

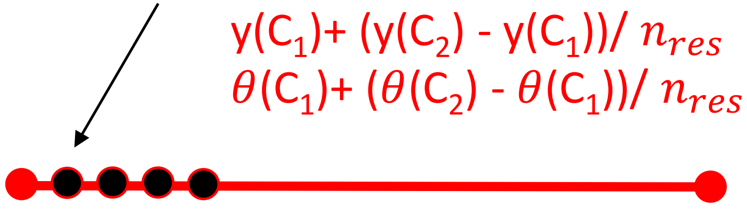
Linear interpolation

C-space
(x, y, θ)



1. Define a resolution n_{res}. (e.g. 1000)
2. Compute the difference in each DOFs
3. Discretize the DOF in n_{res} parts.

$C_i = x(C_1) + (x(C_2) - x(C_1)) / n_{res}$
$y(C_1) + (y(C_2) - y(C_1)) / n_{res}$
$\theta(C_1) + (\theta(C_2) - \theta(C_1)) / n_{res}$



$C_1 = (2/2/135°)$

$C_2 = (3/2{,}5/0°)$

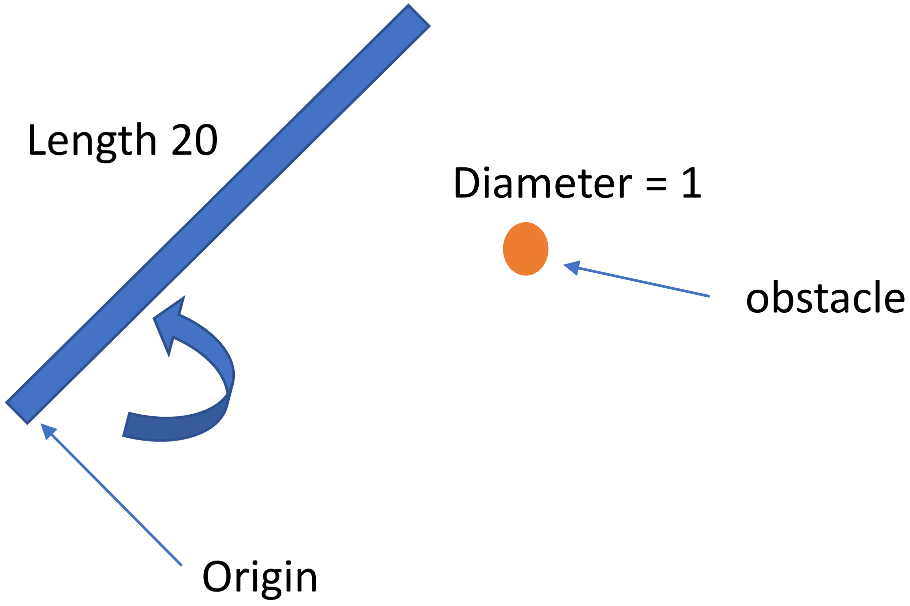
Linear interpolation with fixed resolution

Is linear interpolation the best way?

- No, but it is the simplest way to check for feasibility of the line.
- As it is the simplest and easiest to implement approach → **We will use it in the practical work study.**

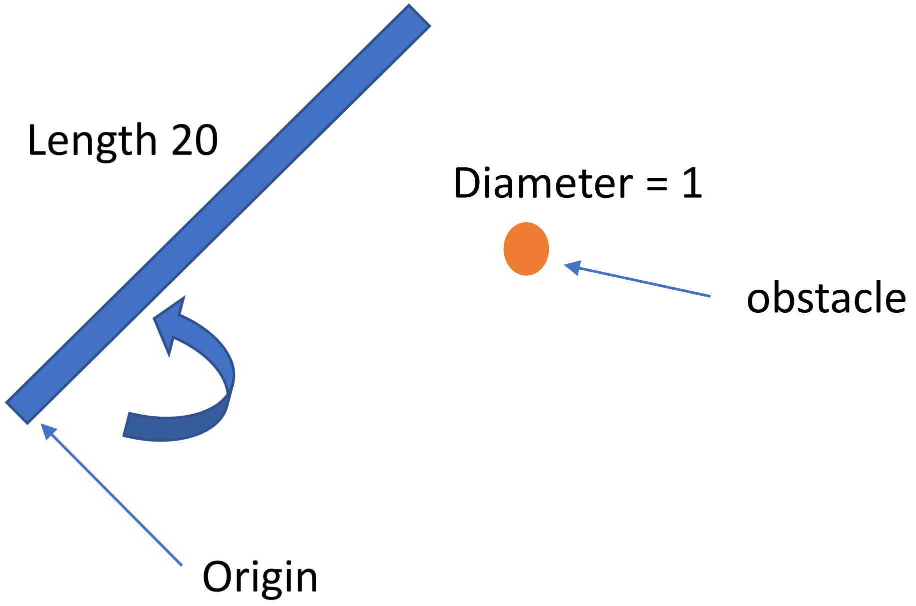
Why is it bad?

The resolution depends on the robot. Assume we have the following robot:



Why is it bad?

For the translational (if you move 20 units), a resolution 20 is enough, but for the angle 1/20 of
360° is not enough as it could might miss the obstacle.



Impacts

- You need to make the resolution very high (How much? More that enough) in order to make sure all rotations are captured and don't miss obstacles.
- You need more collision detection calls to verify a line in $\mathcal{C}$
- It increases performance

Better approach

- In practical applications the precision is not defined by a defined resolution.
- There is a concrete precision (in units) given for the workspace.

For example:

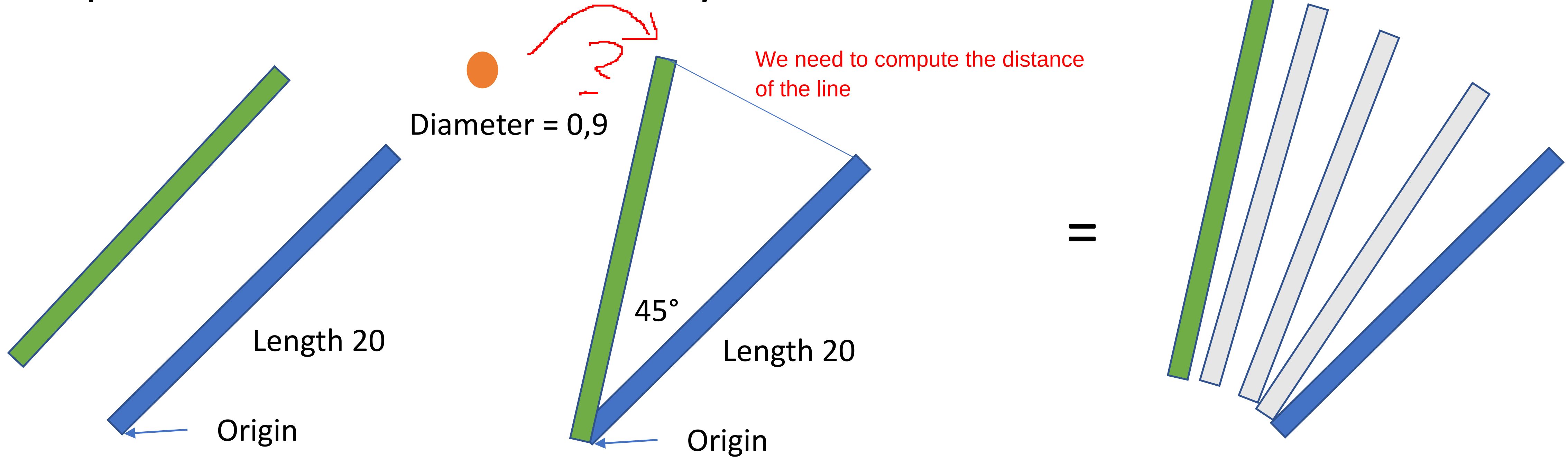
In the workspace every obstacle of 1 mm thickness has to be captured.

1. Measure the maximal translation of any point of the robot by the rotation.
   → Compute needed resolution
2. Measure the maximal translation of any point of the robot by the translation
   → Compute needed resolution
3. Take the max resolution needed (assume resolution is an Integer)

Better approach

Scenario:

In the workspace every obstacle of 0,9 unit thickness has to be captured. The robot is moved by 5 units and rotated about 45°



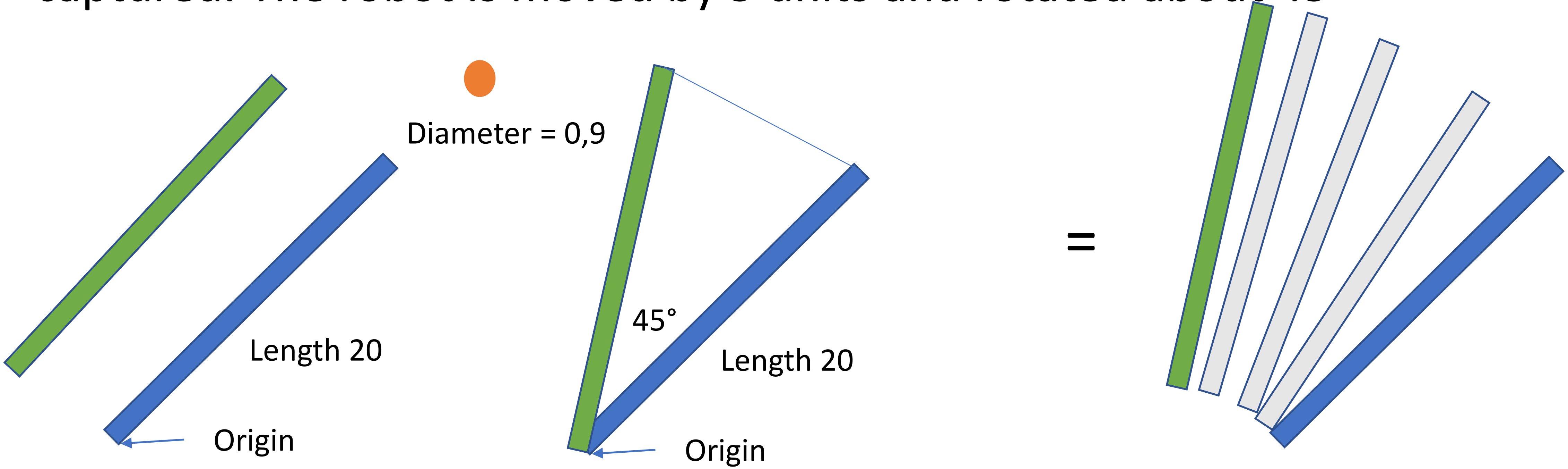
Better approach

Exercise:

What minimal resolution (INT, round up) that is needed for this movement? And how many collision detections are needed? Hint: Assume that the collision detection for the start (blue) is already done.

Scenario:

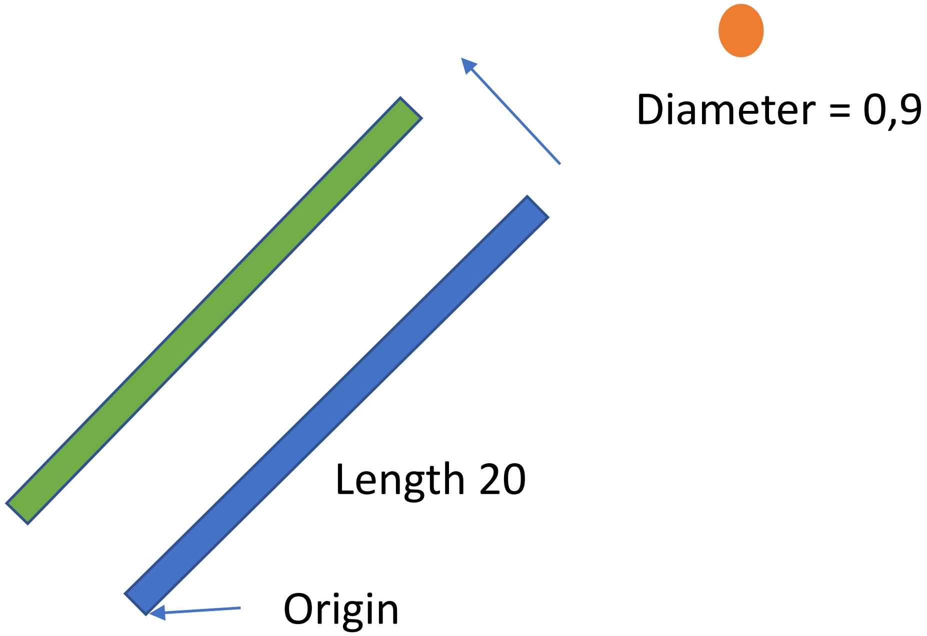
In the workspace every obstacle of 0,9 unit thickness has to be captured. The robot is moved by 5 units and rotated about 45°



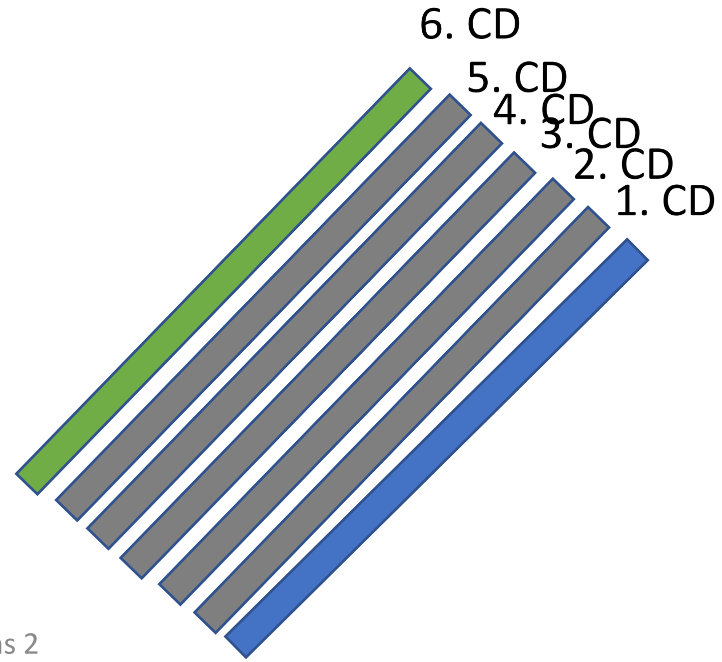
Better approach

Solution:

1. Translation



1. Robot moves 5 units.
2. 5 units/0,9 diameter = 5,5 steps
3. Resolution/CD = 6

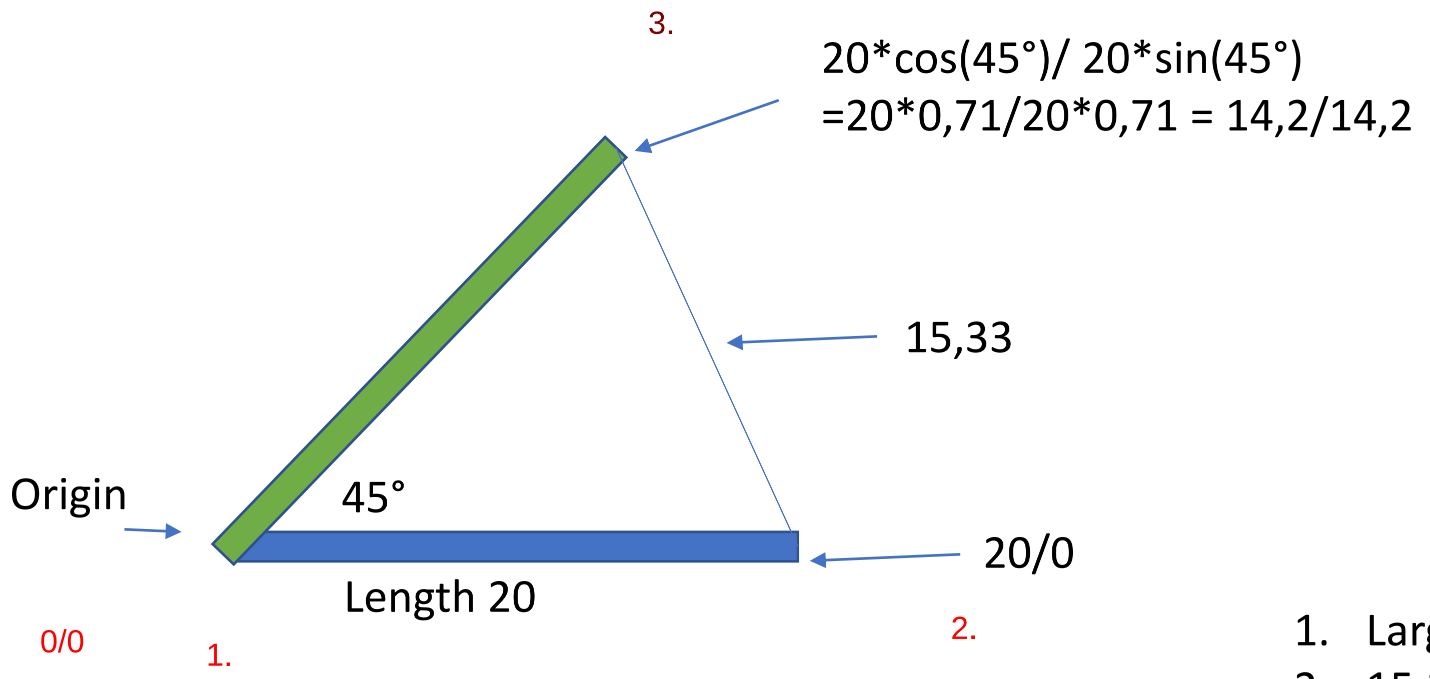


sqrt((14,2-20)^2 + (14,2-0)^2) = 15.33

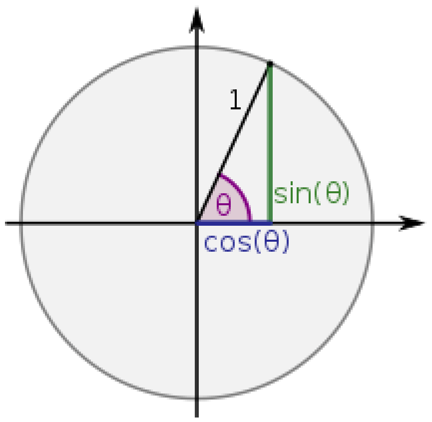
Better approach

Solution:

2. Rotation



Unitcircle

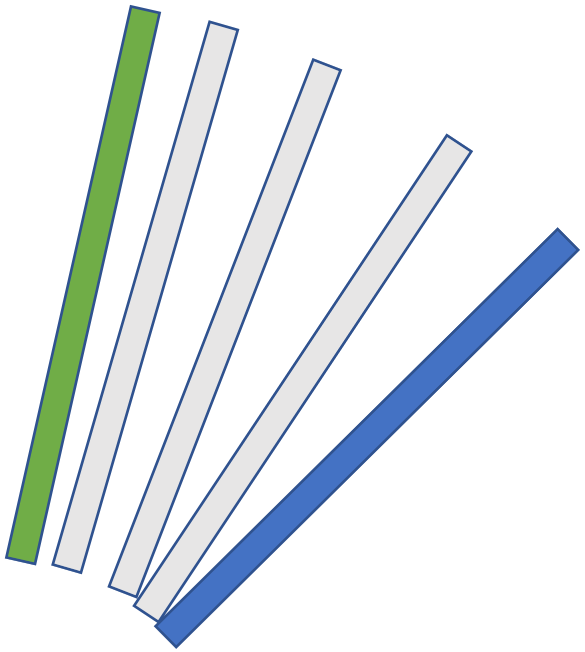


1. Largest distance is 15,33
2. 15,33 units/0,9 diameter = 17,03 steps
3. Resolution/CD = 18

Better approach

Solution:

Combined

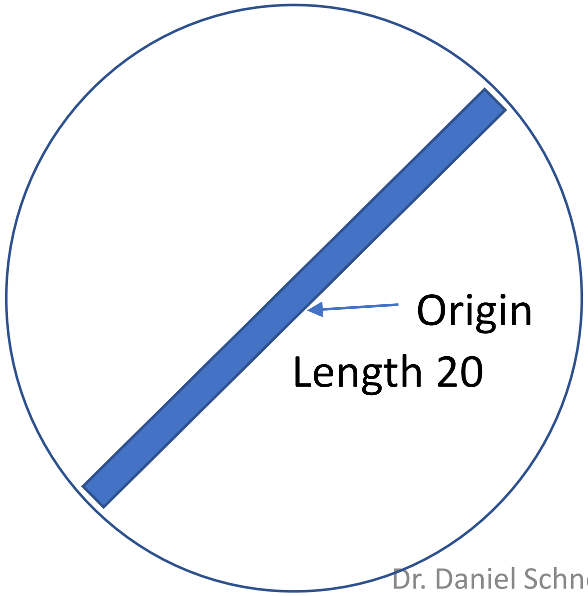


max(6,18) = 18
A resolution of 18 is needed → This means 18 Collision calls.

Better approach

Some topic to notice:

In the workspace every obstacle of 1 mm thickness has to be captured.



Note:

- This depends on the definition of the origin.
- This depends also on the geometry of the robot.
- Set the origin in a way that the max rotation is minimal (Minimal bounding sphere)

Better approach

Advantages:
- Resolution is defined by the movement between two configurations.
- Less collision detections needed.

Disadvantages:
- More complex to implement than linear approach.
- Minimal bounding sphere algorithm needed.
- A bit more computing time to compute the resolution on the fly. But can be ignored.