Centre for Robotics

EGB339 Part 2: Robotic Arms

Lecture 5: Path and Trajectory Planning

Chris Lehnert (Lecturer)

Centre for Robotics

Outline

- Subjects covered in this series of lectures
  - Rigid Body Motions (week 8)
  - Forward Kinematics (week 9)
  - Inverse Kinematics (week 10)
  - Velocity Kinematics (week 11)
  - Path and Trajectory Planning (week 12)
  - Revision (week 13)

- Subjects not covered in this series of lectures
  - Dynamics
  - Control
  - Hardware
  - (Artificial) Intelligence
  - …

Watch these online videos

- QUT Robot Academy (by Prof Peter Corke)
  - Paths and Trajectories
    - https://robotacademy.net.au/masterclass/paths-and-trajectories/

Centre for Robotics

Review of Week 11

- The Jacobian is a matrix that is a function of joint position, that linearly relates joint velocity to toolpoint velocity.
- The Forward Velocity Kinematics maps the velocity of the joints to the velocity of the tool.
- The Inverse Velocity Kinematics maps the velocity of the tool to the velocity of the joints.
- From the viewpoint of velocity kinematics, singularity means a configuration of the robot in which the Jacobian matrix becomes rank-deficient.
- The static force/torque analysis can be done with the Jacobian matrix.

Common Questions

- Rank-deficient?
  - Rank of a matrix is the dimension of the largest submatrix that is invertible.

$$\text{Rank}\left(\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}\right)=3 \qquad \text{Rank}\left(\begin{bmatrix}1 & 0 & 0\\0 & 0 & 0\\0 & 0 & 1\end{bmatrix}\right)=2$$

$$\text{Rank}\left(\begin{bmatrix}0 & 0 & 0\\0 & 1 & 0\\0 & 0 & 0\end{bmatrix}\right)=1 \qquad \text{Rank}\left(\begin{bmatrix}0 & 0 & 0\\0 & 0 & 0\\0 & 0 & 0\end{bmatrix}\right)=0$$

Common Questions

- What if Jacobian is non-square?
  - Forward velocity kinematics is not affected
  - For inverse velocity kinematics, two cases:
    - $J_{m\times n}$, m<n, could be infinite solutions
      - May use pseudo-inverse to calculate a solution
        - Pseudo-inverse: $J^{\dagger} = J^T(JJ^T)^{-1}$
        - $\dot{q} = J^{\dagger}\dot{p}$
    - $J_{m\times n}$, m>n, could be no exact solutions
      - May use pseudo-inverse to calculate an approximate solution
        - Pseudo-inverse: $J^{\dagger} = (J^TJ)^{-1}J^T$
        - $\dot{q} = J^{\dagger}\dot{p}$

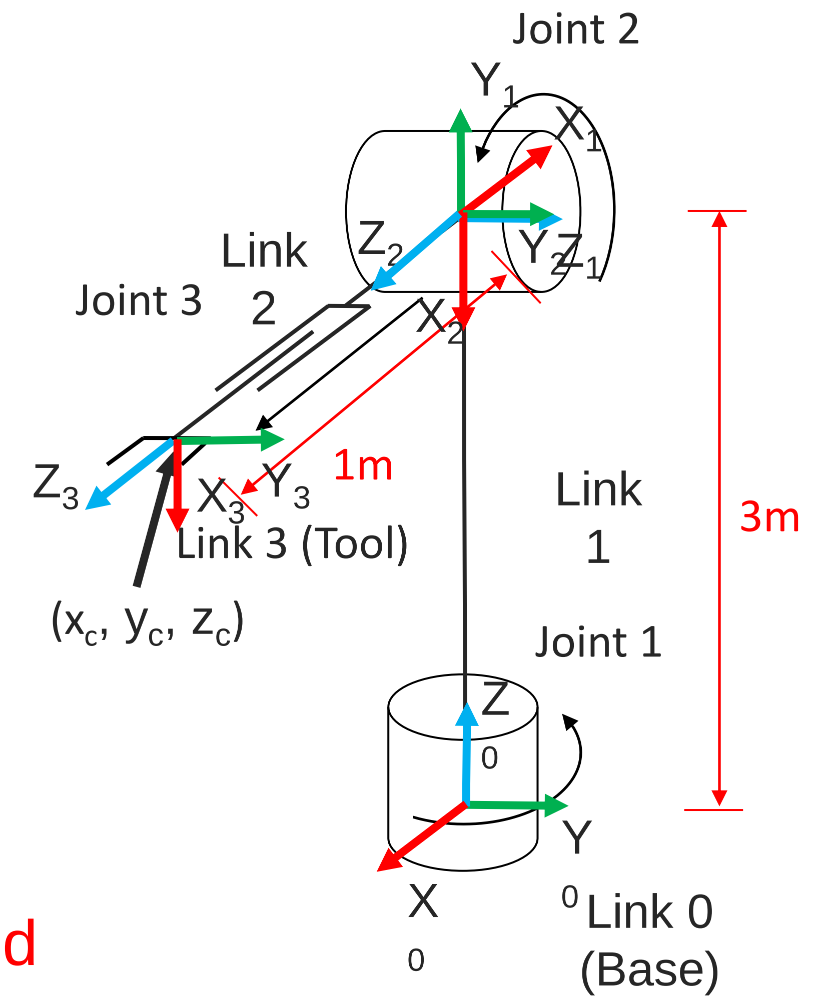
Common Questions

- Find the force/torque **required** at the joints to support a 3 kg (30 N) load at the tool. (q_1=0, q_2=0, q_3=0) i.e., (ϑ_1=0, ϑ_2=-90, d_3=1)

$$J_v = \begin{bmatrix} d_3 s_1 s_2 & -d_3 c_1 c_2 & -c_1 s_2 \\ -d_3 c_1 s_2 & -d_3 s_1 c_2 & -s_1 s_2 \\ 0 & d_3 s_2 & -c_2 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

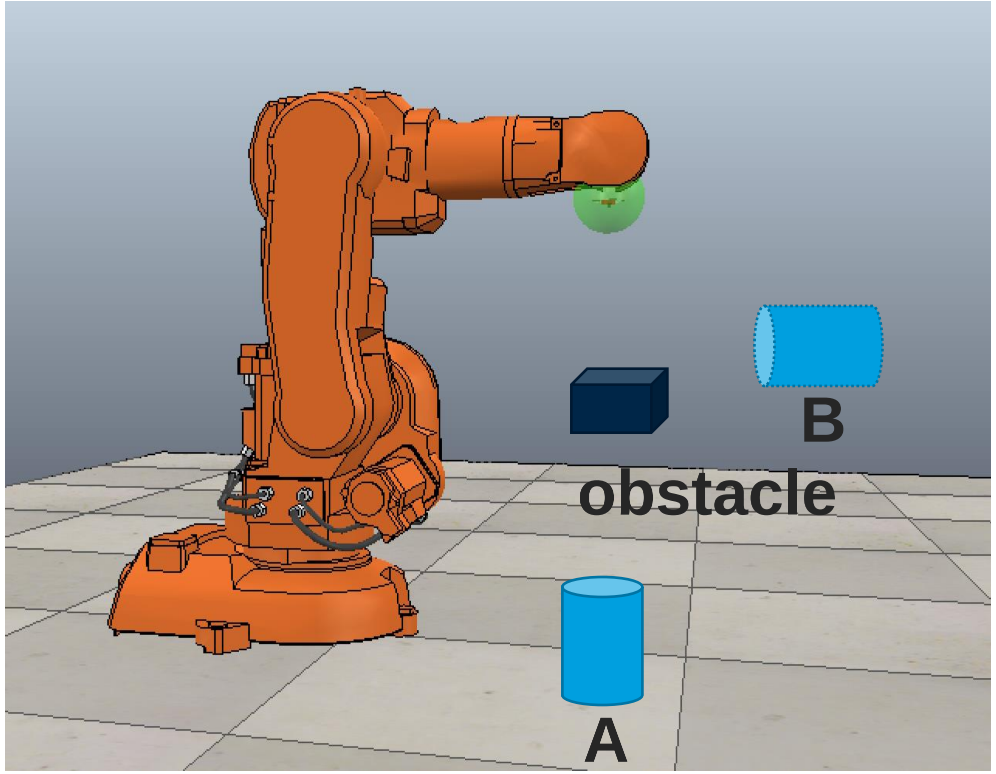
$$\tau = J_v^T F = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -30 \end{bmatrix} = \begin{bmatrix} 0 \\ 30 \\ 0 \end{bmatrix}$$



Joint 2 experiences 30 Nm due to the load, and requires -30 Nm to support the load.

Motivating Problem

- Imagine one of your arms is replaced by a robotic arm. You are supposed to move an object from A to B.
- Now you know where the object is in front of you (homogeneous transformation).
- You know where your “hand” is with respect to your “body” (forward kinematics).
- You know how to move your “hand” to reach the object (inverse kinematics) at a certain speed (velocity kinematics).
- Can you find a path to move the object while avoiding the obstacle?

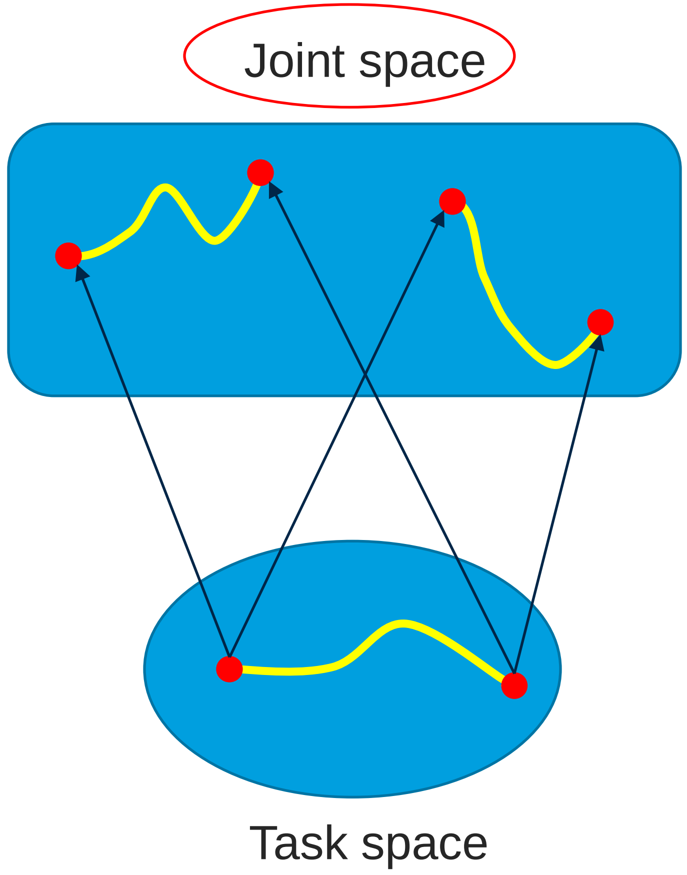


Paths and trajectories

https://robotacademy.net.au/masterclass/paths-and-trajectories/?lesson=109

Path Planning

- Path is a curve connecting a starting configuration q_0 to a goal configuration q_1.
- One path in the joint space corresponds to one path in the task space.
- One path in the task space could correspond to multiple paths in the joint space.
  - Because of multiple inverse kinematics solutions
- Path planning is to find a valid collision free path in the joint space to connect the starting configuration to the goal configuration.

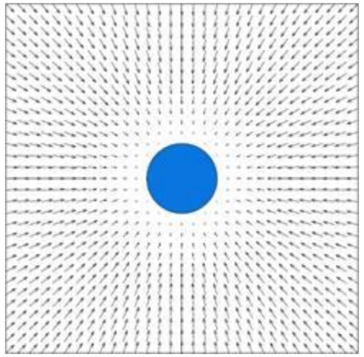


Methods for Path Planning

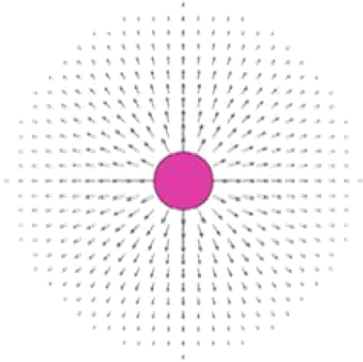
- Artificial Potential Field
- Sampling Based Planning
- Grid Based Planning
- Reward Based Planning

Artificial Potential Field

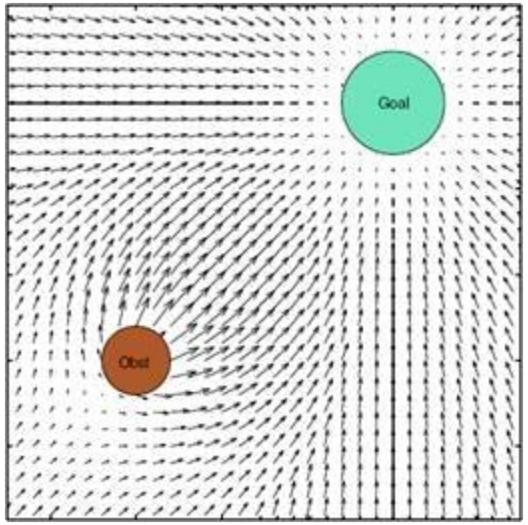
- Treat the goal as an attractive potential field.
- Treat obstacles as repulsive potential field.
- Sum potential fields
- Follow the force (direction and magnitude) of the combined potential field
  - Should lead to the goal while avoiding obstacles



Attractive



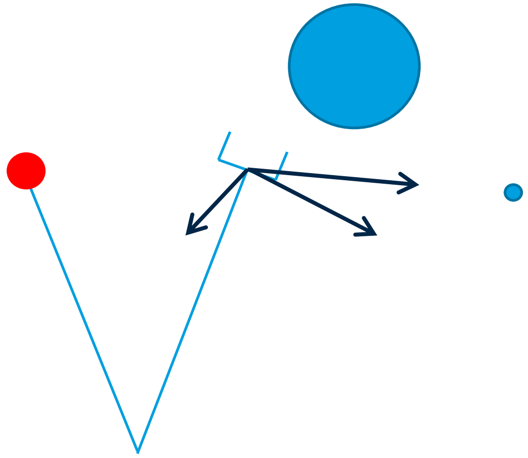
Repulsive



Combined

Centre for Robotics

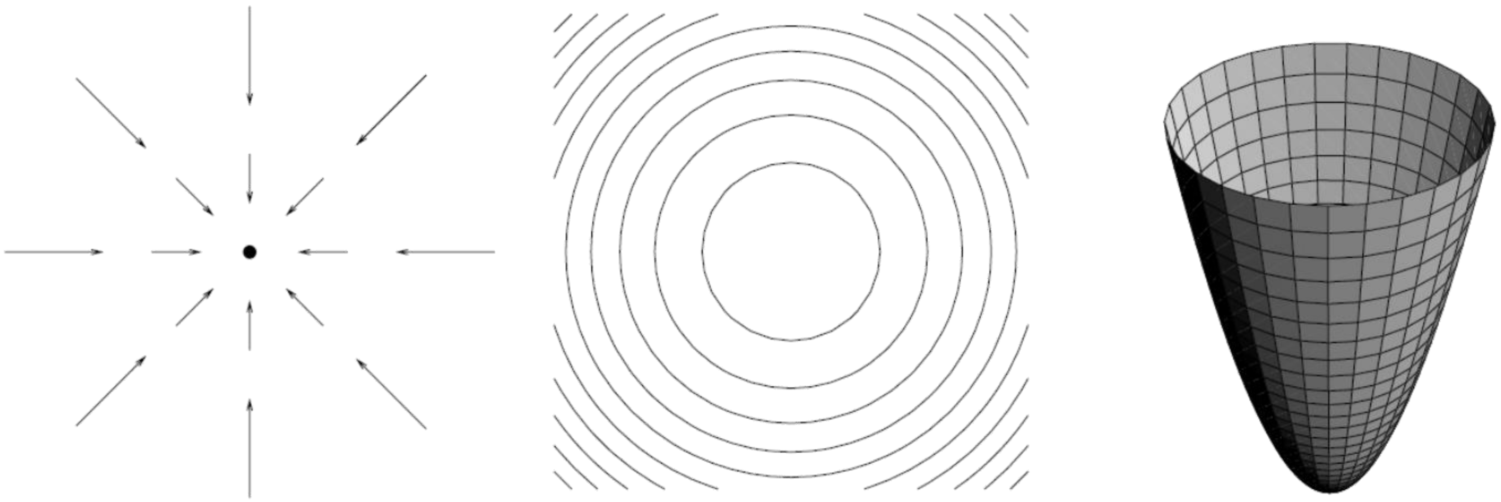
Artificial Potential Field



- Treat the goal as an attractive virtual force.
- Treat obstacles as repulsive virtual forces.
- Sum virtual forces on toolpoint.
- Use J^T to change virtual force on toolpoint to virtual torques at joints. Remember $\tau = J^T F$.
- Follow virtual torques in joint space to goal while avoiding obstacles.
  - Can use gradient descent method to perform this iteratively.
- **YOU DO NOT NEED TO DO ANY OF THIS TO MAKE A PERFECT PRAC ROBOT. But it may help you think about it.**

Attractive Potential Functions

- Conical $$U(q) = \zeta d(q, q_{goal})$$
- Quadratic $$U(q) = \frac{1}{2}\zeta d^2(q, q_{goal})$$

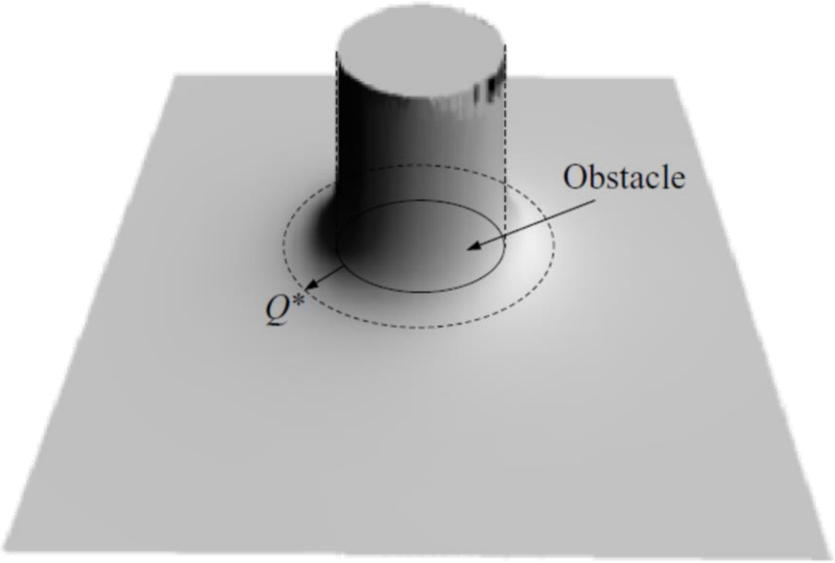


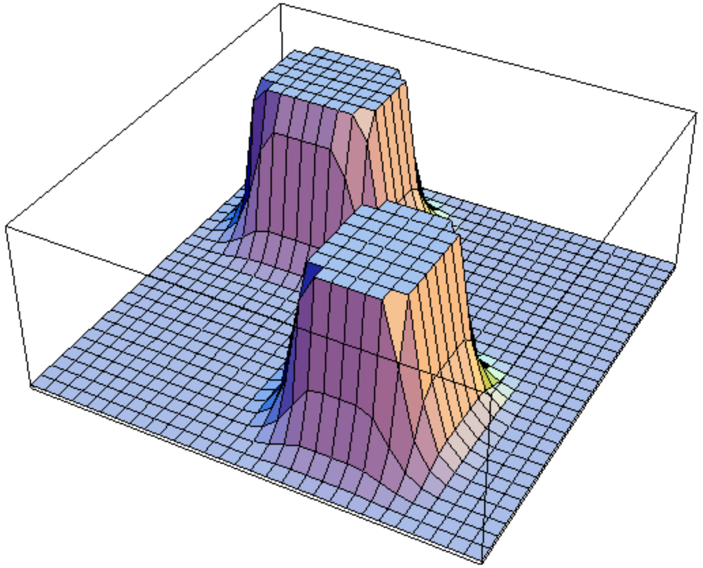
(a) (b) (c)

$\zeta = scalar,$ d(∗)=distance function, $q-$ robot position

Repulsive Potential Functions

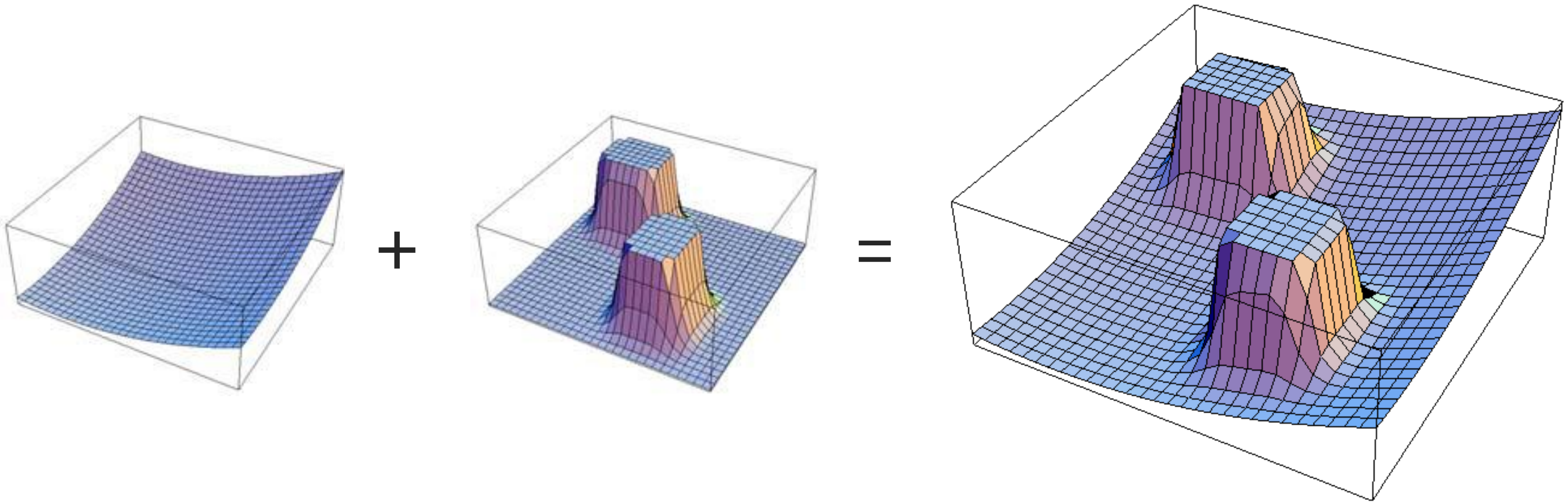
$$U(q) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{d(q)} - \frac{1}{Q^*}\right)^2 & , d(q) \le Q^* \\ 0 & , d(q) > Q^* \end{cases}$$





$\eta = scalar$, $Q^* =$ scalar obstacle buffer, d($*$)=distance function

Total Potential Field



Computing Virtual Forces – Gradient of Field

Location of toolpoint (which is a function of the joint values q)

Location of goal

- Attractive field for goal
  - Force directed towards goal, magnitude proportional to distance.
  - For a goal at *p*: $F_{att}(q) = -\zeta(o_i(q) - p)$

- Repulsive field for obstacles
  - Force directed away from obstacle, magnitude proportional to inverse square of distance
  - For an obstacle at *b*:

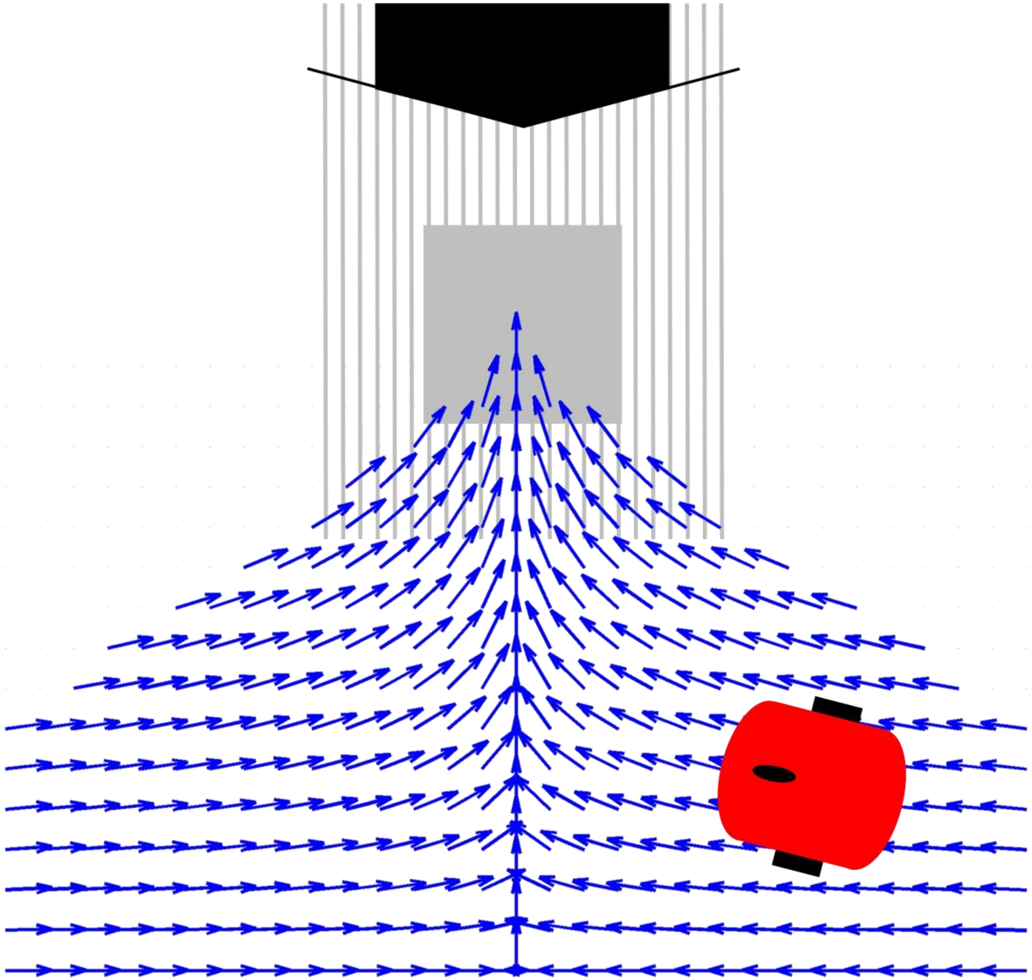
Location of obstacle

$$F_{rep}(q) = \eta \frac{(o_i(q) - b)}{\|o_i(q) - b\|^3}$$

Double vertical lines are the "norm" – take the square root of the sum of the squares of each vector element

$$\boxed{\|a\| = \sqrt{a_1^2 + a_2^2 + \cdots + a_n^2}}$$

Artificial Potential Field in Practice



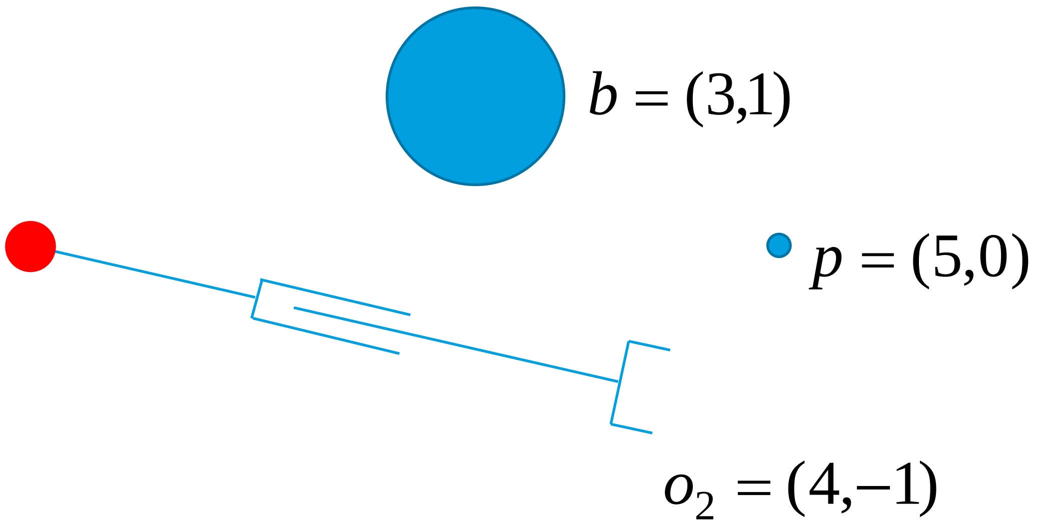
Artificial Potential Field Parameters

- Need to set **goal and obstacle gains** ζ **and** η.
  - Ratio of gains is more important than absolute values.

- Set up **gradient descent step size** α and **time step** Δt in motion generation algorithm.
  - Choose a value that moves toolpoint at a safe speed.
  - Large steps can hit obstacles, but reduce compute time.
  - Use cubic polynomial between points – will practice in the tutorial.

Artificial Potential Field - Example

- Calculate the net potential field force when the robot is at (4, -1). Use $\zeta = 1$ and $\eta = 3$.

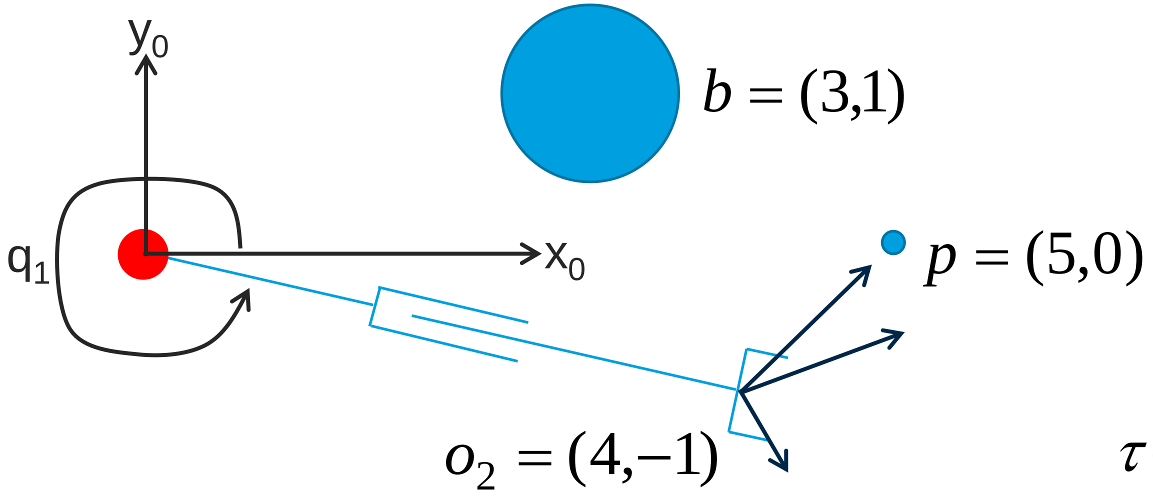


$$F_{att}(q) = -\zeta\left(o_i(q) - p\right)$$

$$F_{rep}(q) = \eta \frac{\left(o_i(q) - b\right)}{\left\| o_i(q) - b \right\|^3}$$

Artificial Potential Field - Example

- Now convert the force to joint torques, given:



$$F_{att}(q) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

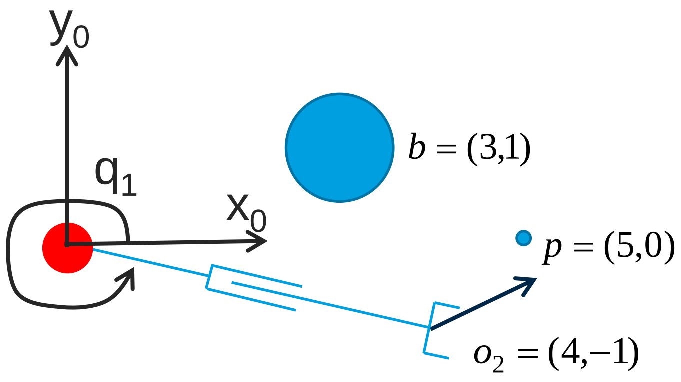
$$F_{rep}(q) = \begin{bmatrix} 0.26 \\ -0.52 \end{bmatrix}$$

$$F(q) = \begin{bmatrix} 1.26 \\ 0.48 \end{bmatrix}$$

$$\tau(q) = J^T F(q)$$

$$J = \begin{bmatrix} -q_2 s_1 & c_1 \\ q_2 c_1 & s_1 \end{bmatrix}$$

Artificial Potential Field - Example



$$\tau(q) = J^T F(q)$$

$$J = \begin{bmatrix} -q_2 s_1 & c_1 \\ q_2 c_1 & s_1 \end{bmatrix}$$

$$F(q) = \begin{bmatrix} 1.26 \\ 0.48 \end{bmatrix}$$

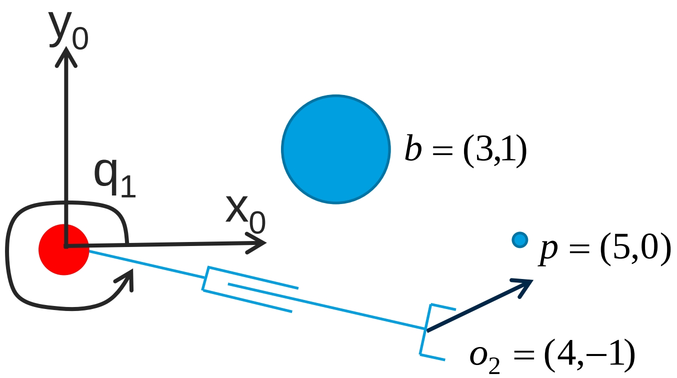
$q_1 = atan2(4, -1) = -0.245$ rad

$q_2 = \sqrt{16 + 1} = 4.12$ m

$$J = \begin{bmatrix} 1 & 0.970 \\ 4 & -0.243 \end{bmatrix}$$

$$\tau = \begin{bmatrix} 1 & 4 \\ 0.970 & -0.243 \end{bmatrix} \begin{bmatrix} 1.26 \\ 0.48 \end{bmatrix} = \begin{bmatrix} 3.18 \\ 1.11 \end{bmatrix}$$

Artificial Potential Field - Example



$$q_1 = \operatorname{atan}2(4,-1) = -0.245$$

$$q_2 = \sqrt{16+1} = 4.12$$

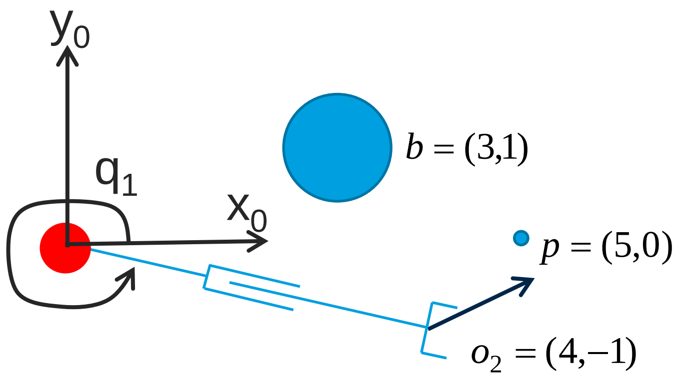
$$\tau = \begin{bmatrix} 1 & 4 \\ 0.970 & -0.243 \end{bmatrix} \begin{bmatrix} 1.26 \\ 0.48 \end{bmatrix} = \begin{bmatrix} 3.18 \\ 1.11 \end{bmatrix}$$

- Using a gradient descent step size of $\alpha = 0.1$ and a time step of $\Delta t = 0.1s$, find the new joint positions and velocities.

$$q(t+\Delta t) = q(t) + \alpha \frac{\tau(q)}{\|\tau(q)\|}$$

$$\dot{q}(t+\Delta t) = \frac{q(t+\Delta t) - q(t)}{\Delta t} = \frac{\alpha}{\Delta t} \frac{\tau(q)}{\|\tau(q)\|}$$

Artificial Potential Field - Example



$$q_1 = \operatorname{atan2}(4,-1) = -0.245$$

$$q_2 = \sqrt{16+1} = 4.12$$

$$\tau = \begin{bmatrix} 3.18 \\ 1.11 \end{bmatrix}$$

$$\alpha = 0.1$$

$$\Delta t = 0.1s$$

$$q(t+\Delta t) = q(t) + \alpha \frac{\tau(q)}{\|\tau(q)\|}$$

$$\begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} -0.245 \\ 4.12 \end{bmatrix} + \frac{0.1}{3.36} \begin{bmatrix} 3.18 \\ 1.11 \end{bmatrix} = \begin{bmatrix} -0.15 \\ 4.15 \end{bmatrix}$$

$$\dot{q}(t+\Delta t) = \frac{\alpha}{\Delta t} \frac{\tau(q)}{\|\tau(q)\|} = \begin{bmatrix} 0.946 \\ 0.330 \end{bmatrix}$$

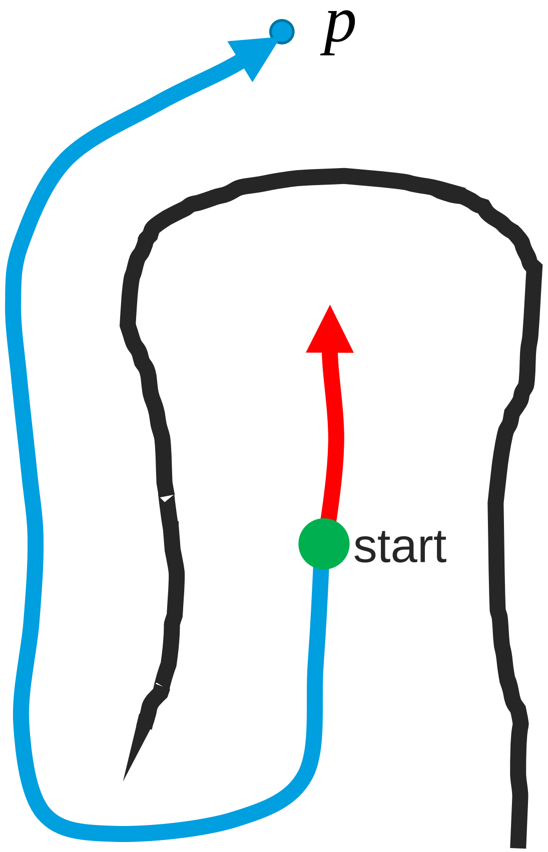
- Using the current and next positions and velocities, you could go on to construct a cubic polynomial trajectory.

Artificial Potential Field Features

- Useful for planning paths around obstacles.
  - Works with dynamic obstacles too.

- Won't land precisely at goal.
  - Use a simpler trajectory planner once close to goal.

- Can get stuck in **local minima**.
  - Places where attractive and repulsive forces cancel to zero.
  - Basins where obstacles "herd" toolpoint to centre.
  - Classic rookie mistake



Trajectory

- A **path** and a **schedule** for getting from A to B
  - ➥ there is a notion of time or speed

Desirable Properties of a Trajectory

- Spatial Accuracy
  - Arrives at the desired location (welding)
- Temporal Accuracy
  - Arrives at the right time (picking up from conveyor belt)
- Temporal Efficiency
  - Doesn’t waste time getting to the desired location (draglines)
- Smoothness
  - A continuous function with a finite first derivative (duty cycle)

Different Trajectory Types

- Cubic Polynomial Trajectories
- Quintic Polynomial Trajectories
- Linear Segments with Parabolic Blends (LSPB)
- Minimum Time Trajectories (Bang-Bang)

Cubic Polynomial

- Describing (joint) position as a cubic polynomial gives **spatial** and **temporal** accuracy, and **smoothness**.

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

$$\dot{q}(t) = a_1 + 2a_2 t + 3a_3 t^2$$

$$\ddot{q}(t) = 2a_2 + 6a_3 t$$

- Find constants by setting initial and final positions and velocities and choosing a trajectory time.

- **<u>4 variables, need to write 4 independent equations</u>**

Cubic Polynomial - Example

- Use a cubic polynomial to describe motion from 0 to 30 degrees in 3 seconds, with zero start and stop velocities.
- First write the two position equations:

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

$$0 = a_0$$

$$30 = a_0 + 3a_1 + 9a_2 + 27a_3$$

- Then write the two velocity equations:

$$\dot{q}(t) = a_1 + 2a_2 t + 3a_3 t^2$$

$$0 = a_1$$

$$0 = a_1 + 6a_2 + 27a_3$$

- Solve for a_0, a_1, a_2 and a_3.

Cubic Polynomial - Example

- Solve for a_0, a_1, a_2 and a_3.

$$0 = a_0$$
$$30 = a_0 + 3a_1 + 9a_2 + 27a_3$$
$$0 = a_1$$
$$0 = a_1 + 6a_2 + 27a_3$$

- $a_0 = 0$, $a_1 = 0$

$$30 = 9a_2 + 27a_3$$
$$0 = 6a_2 + 27a_3$$

- $a_2 = 10$, $a_3 = -60/27 = -2.22$

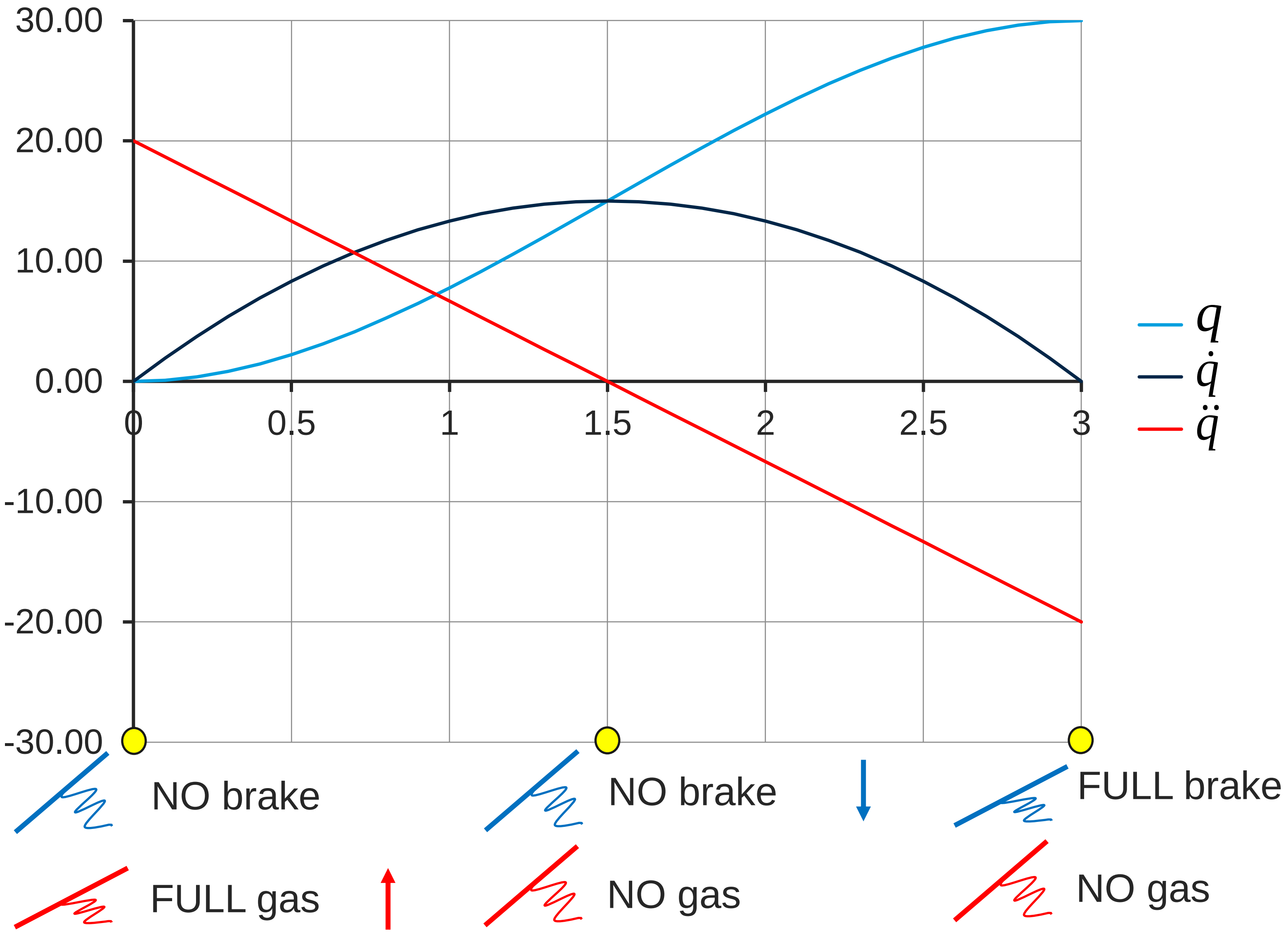
Cubic Polynomial - Example

- $a_0 = 0$, $a_1 = 0$
- $a_2 = 10$, $a_3 = -60/27 = -2.22$
- Write the equations for $q_i(t), \dot{q}_i(t)$

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$
$$\dot{q}(t) = a_1 + 2a_2 t + 3a_3 t^2$$

$$q(t) = 10t^2 - 2.22t^3$$
$$\dot{q}(t) = 20t - 6.66t^2$$

Cubic Polynomial - Example



Cubic Polynomial Drawbacks

- Doesn’t readily facilitate minimum time operations.
  - Not using full actuator capability.

- Need to choose end and start velocities when linking trajectories (to avoid discontinuities).
  - Can use higher order polynomial to ensure smooth connections.

- Infinite jerk (derivative of acceleration) at start and end.
  - Some loss of smoothness.

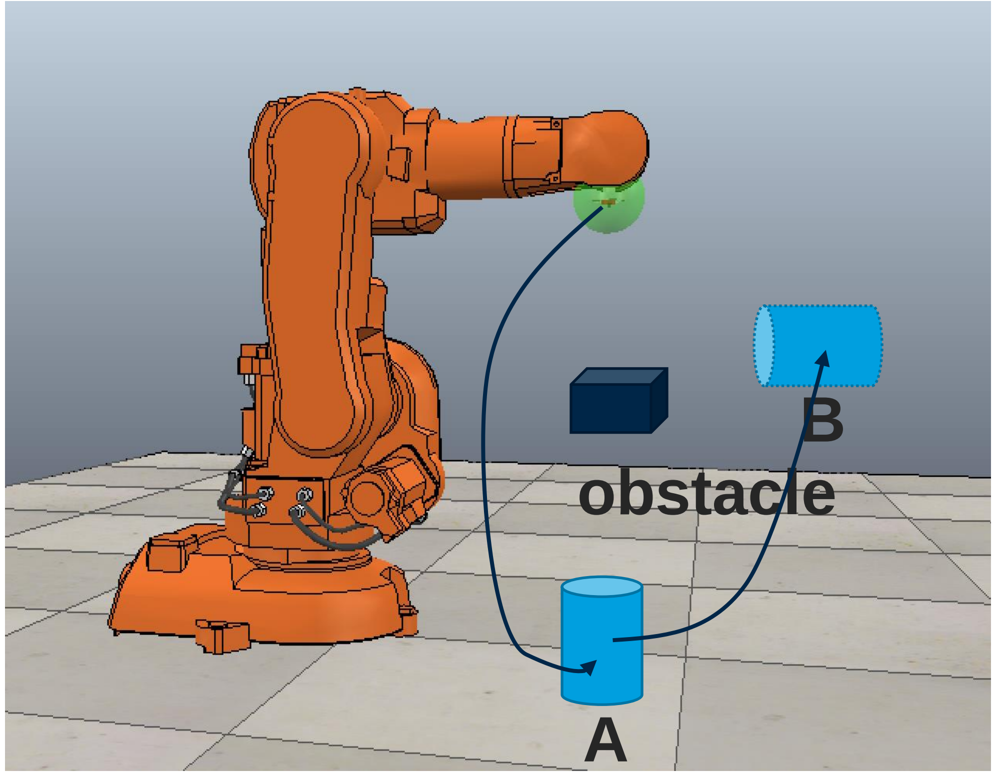
Centre for Robotics

Summary

- Path planning is to find a collision free path in the joint space to connect the starting configuration to the goal configuration.
- Artificial potential field method can formulate the problem in the task space then convert it into the joint space with the Jacobian matrix.
- Trajectory is path associated with time.
- Cubic polynomial trajectory or even higher order trajectories can be used.

Motivating Problem

- Imagine one of your arms is replaced by a robotic arm. You are supposed to move an object from A to B.
- Now you know where the object is in front of you (homogeneous transformation).
- You know where your “hand” is with respect to your “body” (forward kinematics).
- You know how to move your “hand” to reach the object (inverse kinematics) at a certain speed (velocity kinematics).
- Can you find a path to move the object while avoiding the obstacle?
  - Path and trajectory planning



Final Remarks

- Acknowledgements
  - Some material of the slides was developed by the previous lecturers of EGB339 - Introduction to Robotics (Michael Milford, Peter Corke, and Leo Wu)