COMMONWEALTH OF AUSTRALIA

Copyright Regulations 1969

WARNING

This material has been reproduced and communicated to you by and on behalf of The University of New South Wales pursuant to Part VB of the Copyright Act 1968 (the Act).

The material in this communication may be subject to copyright under this Act. Any further reproduction or communication of this material by you may be the subject of copyright protection under the Act.

Do not remove this notice.

Lecture 7 - Revision

https://kahoot.it/

✓ Trajectory generation

- Axis to axis movement
- Simultaneous movement
- Coordinated path
- Continuous straight path

✓ Point to point trajectory

- Quintic polynomial
- Trapezoidal (LSPB)
- Bang-bang trajectory
- Trajectory with via points

✓ Manipulability

- Near singularities $\det J(\mathbf{q}) = 0$: Poorly conditioned

❑ $m = \sqrt{\det(J(\mathbf{q})J(\mathbf{q})^T)}$

MTRN4230
Robotics

Lecture 8

Robotic Path Planning

Hoang-Phuong **Phan** – T2 2023

Learning Objectives

- Introduction to path planning – key definitions
  - Workspace
  - Configuration space

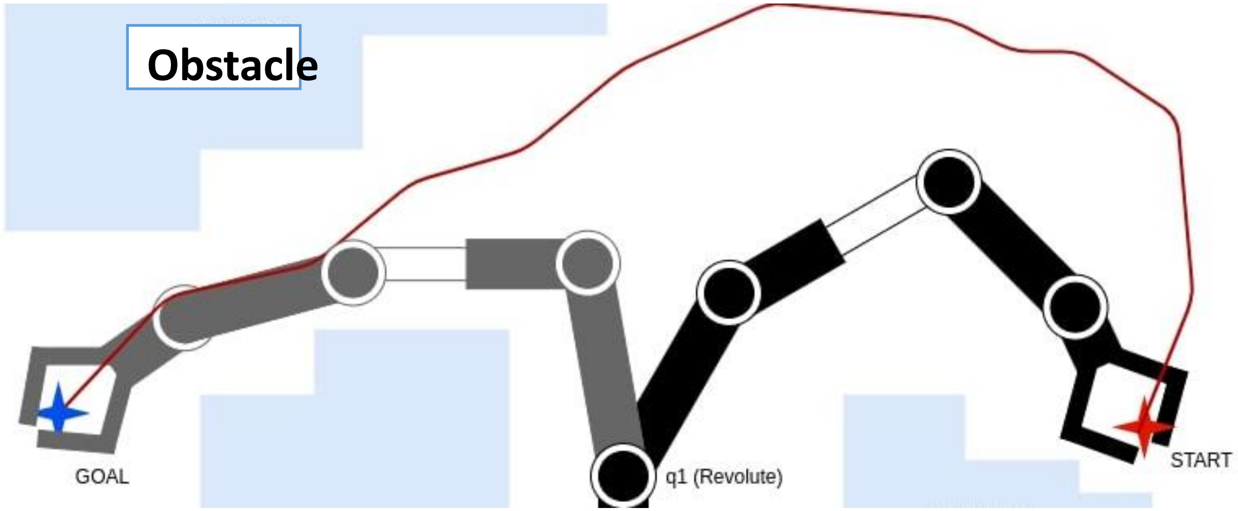
- Map based method
  - Reactive planning
  - D* Method
  - Probabilistic Roadmap method (PRM)
  - Others: Voronoi roadmap, Rapidly-exploring Random Tree (RRT)

- Artificial Potential Field (APF) method

I. Introduction to Path Planning

Introduction to Path Planning

- Last lecture: Trajectory Generation
  - How to control the end effector to pass through a sequence of configurations.
- Path Planning
  - How to generate that sequence of configurations.
  - Configuration space where obstacle exist



Configuration Space

- Configuration space is the set of positions reachable by an end-effector in 3D space, often denoted $SE(3)$.

- The set of joint parameter values is called the **joint space** (a convenient method to represent the Configuration Space - ***C***).

- The robot's forward & inverse kinematics equations define mappings between Workspace and Joint Space.

- Path planning uses these mappings to find a path in joint space that provides a desired path in the workspace of the end-effector.

Configuration Space

- If the robot is a fixed-base serial manipulator with N revolute joints, C is N-dimensional.

- The set of configurations that avoids collision with obstacles or itself is called the free space C_{free}.

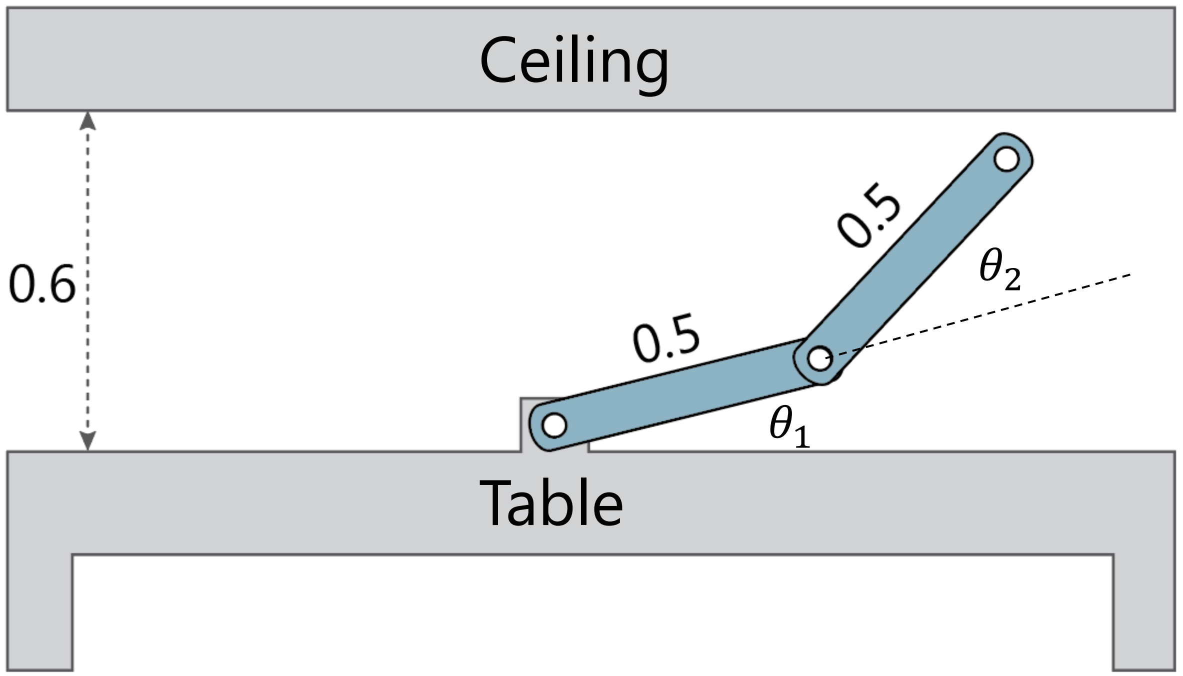
- The complement of C_{free} in C is called the obstacle region, C_{obst}.

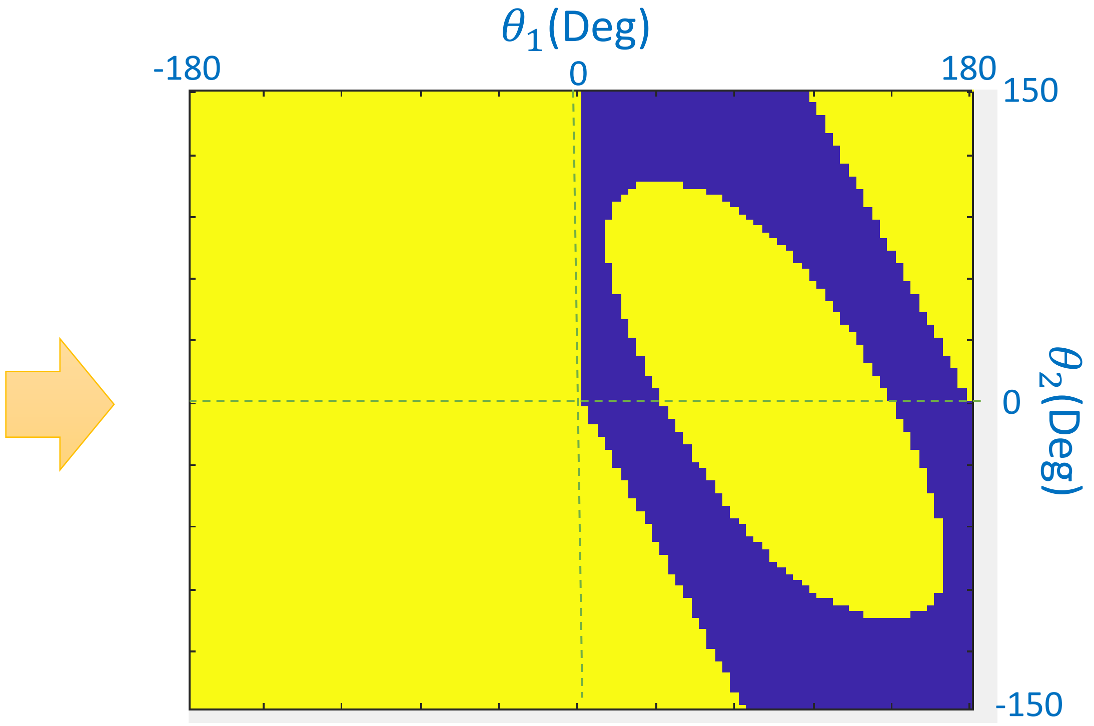
Configuration Space

- Often, it is prohibitively difficult to explicitly compute the shape of C_{free}.
  - However, testing whether a given configuration is in C_{free} is efficient.
  - First, forward kinematics determine the position of the robot's geometry. Then, collision detection tests if the robot's geometry collides with the environment's geometry.

Example: Configuration Space

- The robot is a two-link planar arm, and the workspace contains a table





<u>Workspace</u>

Workspace: the Cartesian space in which the robot moves

<u>Configuration space (See Lab 8)</u>

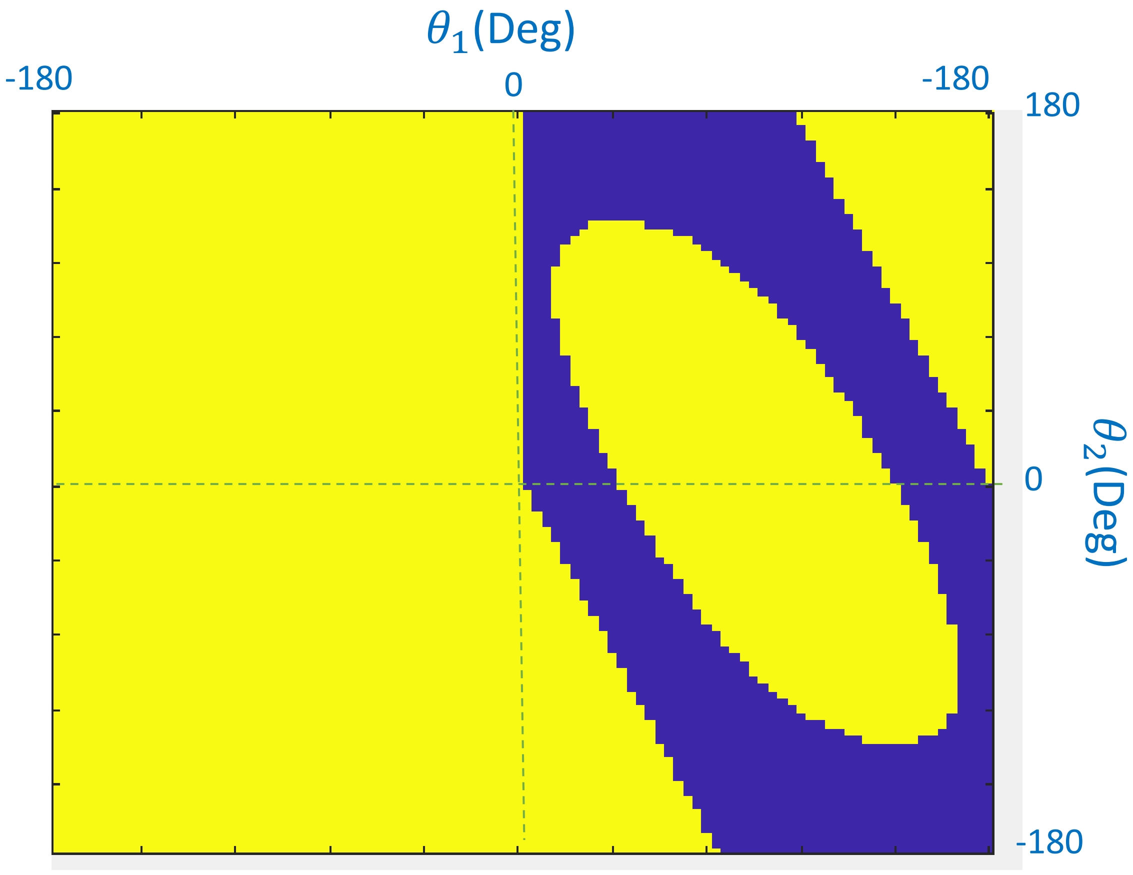
Vector of joint variables provides a convenient representation of a configuration

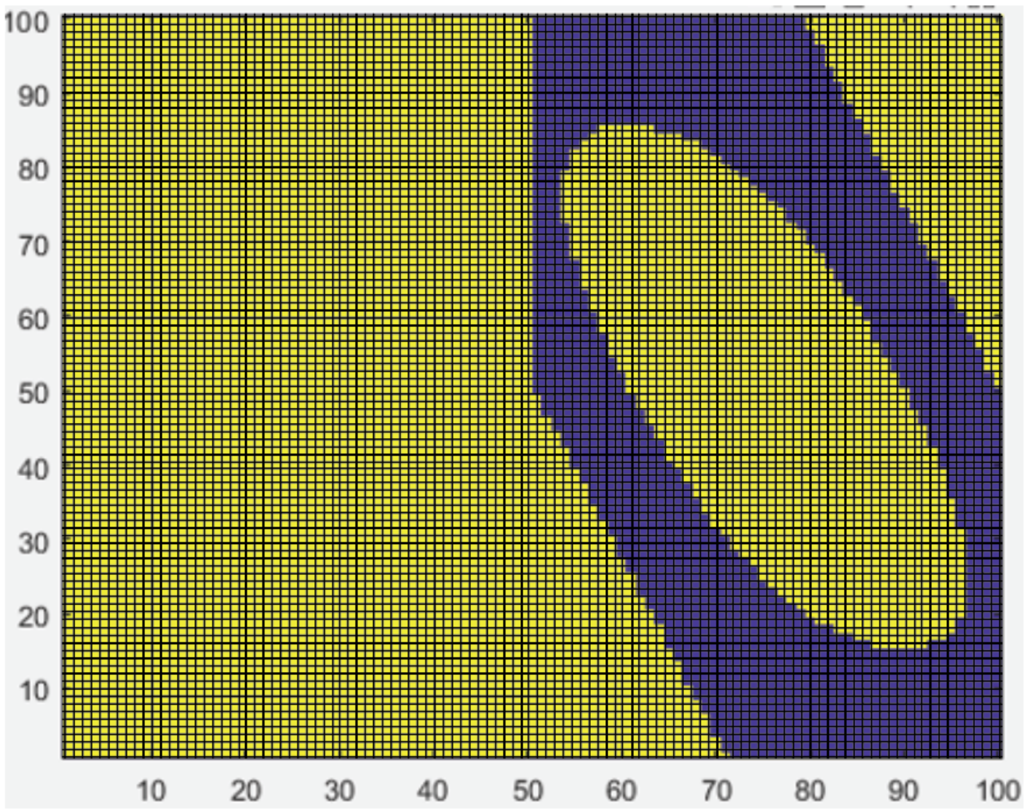
Source: Spong, Mark W., Seth Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. Wiley, 2005.

II. Grid Based Methods

Grid Based Methods

- Overlay a grid on configuration space, and assume each configuration is identified with a grid point.

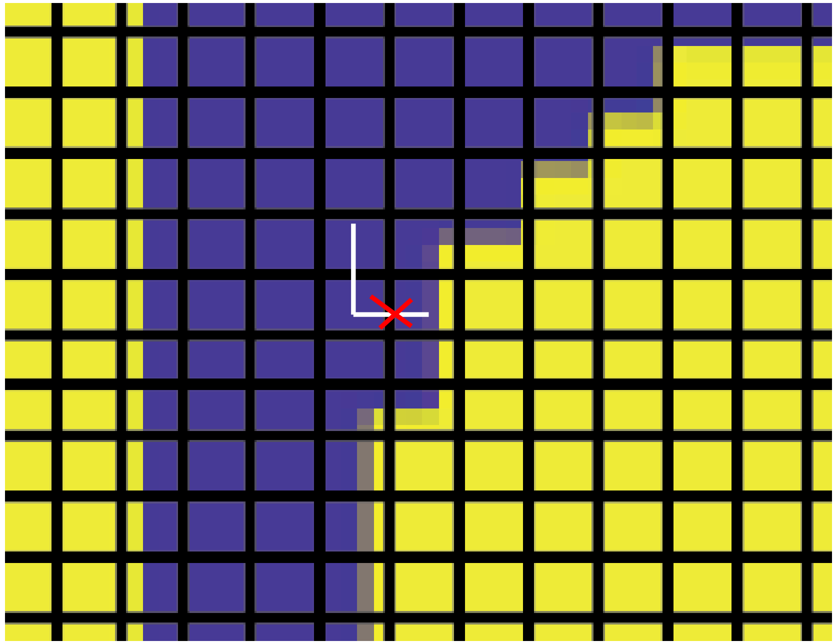


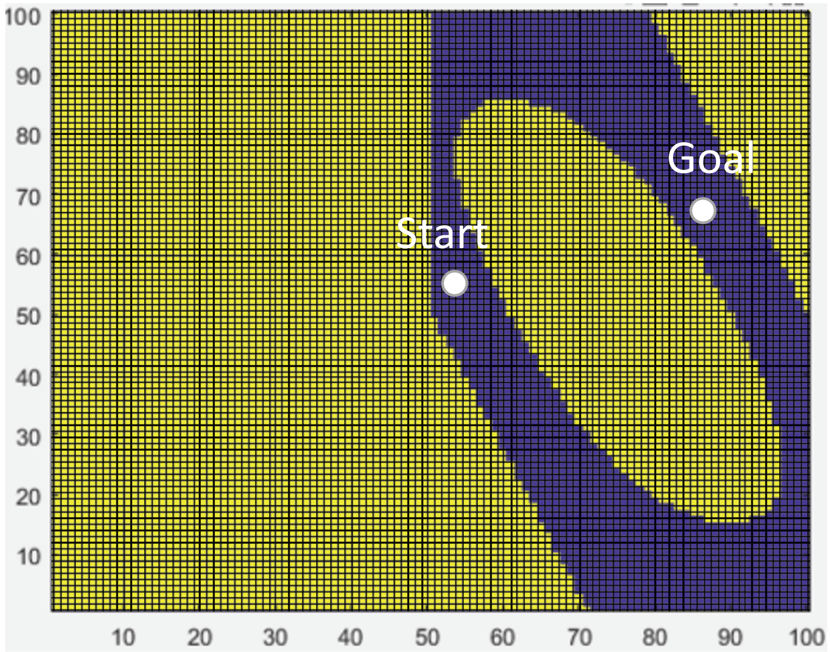


Grid Based Methods

- At each grid point, the robot is allowed to move to adjacent grid points as long as the line between them is completely contained within C_{free} (this is tested with collision detection).

- This discretizes the set of actions, and search algorithms (like D*) are used to find a path from the start to the goal.





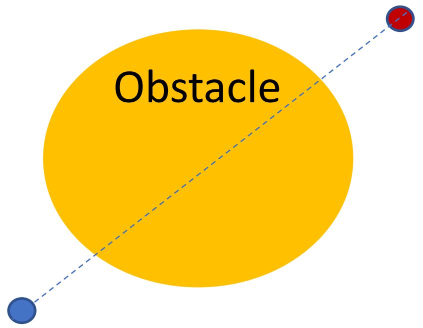
Grid Based Methods

- Reactive planning
- D* Method
- Probabilistic Roadmap method (PRM)

Reactive Planning

❑ Example: Bug2 Algorithm

- Moves in straight line towards GOAL
- Moves counter-clockwise around an obstacle
- Repeat until it encounter a point that lies along its original line that is closer to GOAL



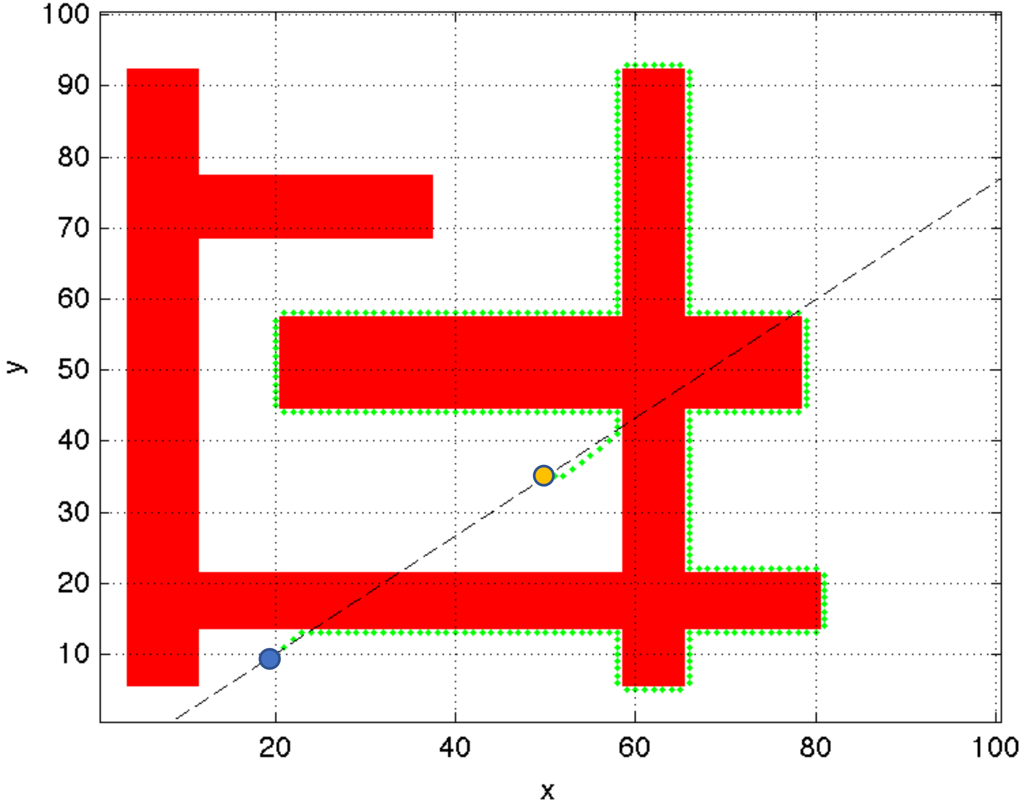
Reactive Planning – Bug2 Algorithm

- In the RVC toolbox

```
bug = Bug2(map) % planning phase

bug.goal = [x; y]; % set the GOAL

p = bug.query([START]); % query phase

dx.plot(p)  % Visualise the path
```

Use makemap(dimension) to create/edit the map

Reactive Planning – Bug2 Algorithm



Not globally optimal!

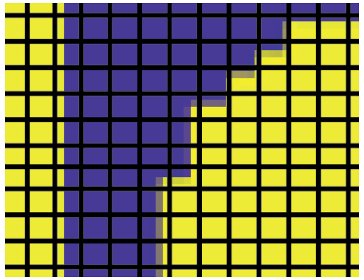
UNSW SYDNEY

D* Algorithm

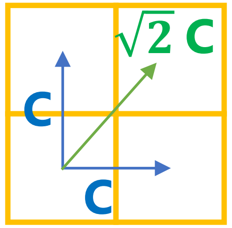
- ❑ A traversal cost is defined for each cell
  - In horizontal or vertical direction: C
  - In diagonal direction: $\sqrt{2}$ C
  - ∞ cost at obstacles

- ❑ This cost can be: distance (travel time or distance), the roughness of the terrain

- ❑ The graph search minimises the cost of traversal (optimal)



Grid



Cost

Search optimal path

Relatively analogous

Sources: P. Corke, *Robotics, Vision and Control: Fundamental Algorithms In MATLAB® Second, Completely Revised, Extended And Updated Edition*. Springer, 2017.
https://www.youtube.com/watch?v=-L-WgKMFuhE&ab_channel=SebastianLague

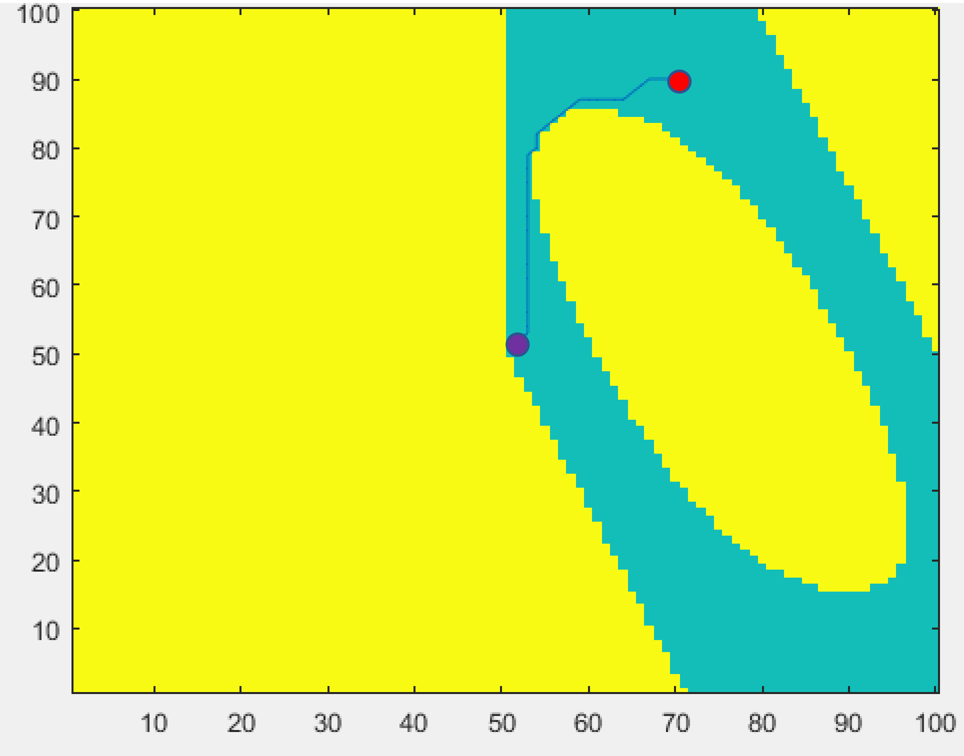
D* Algorithm

❑ In the toolbox

- ds = Dstar(map) % Create a navigation object
- c = ds.costmap() % Convert Grid map into a cost map
- ds.plan(goal) % Creates a very dense directed graph

Each graph vertex has a cost, a distance to GOAL, and a link to the neighbouring cell closest to GOAL

- ds.query(start) % Query phase



P. Corke, *Robotics, Vision and Control: Fundamental Algorithms In MATLAB® Second, Completely Revised, Extended And Updated Edition*. Springer, 2017.

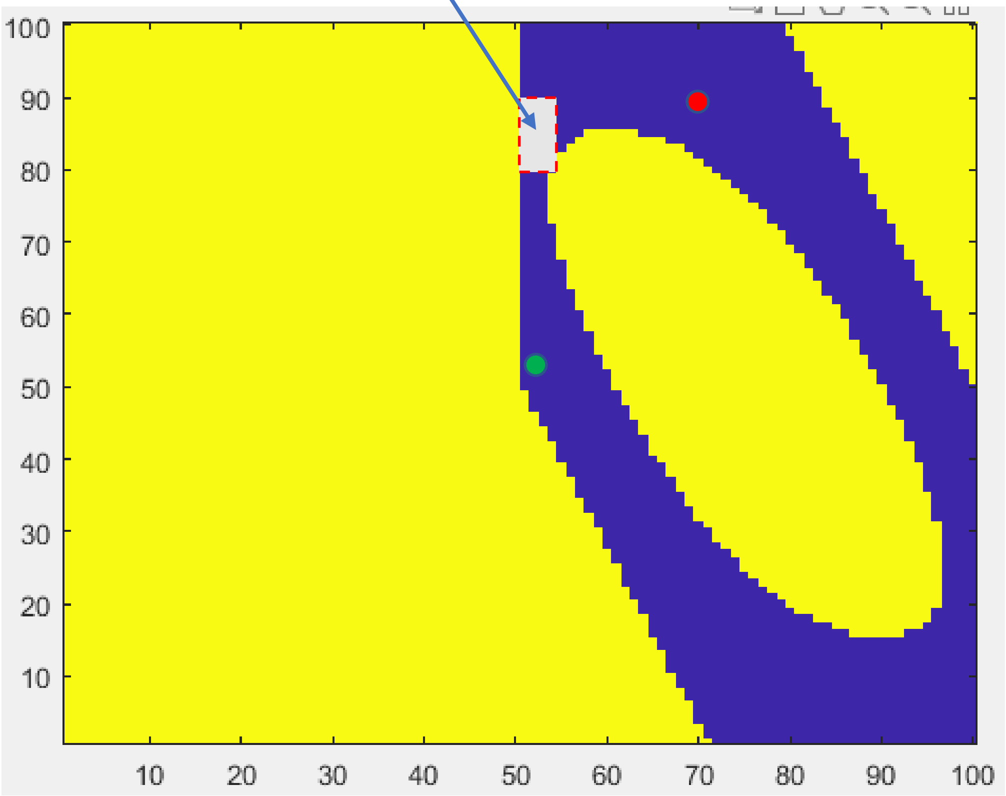
D* Algorithm

- The real power of D*: efficiently change the cost map during the mission.

```
ds.modify_cost( [50,55; 80,99], 100 );
% Modify cost
ds.plan();  % Replan
ds_path1 = ds.query(start_cell); % Search
```

- No need to change the whole graph (incremental replanning)

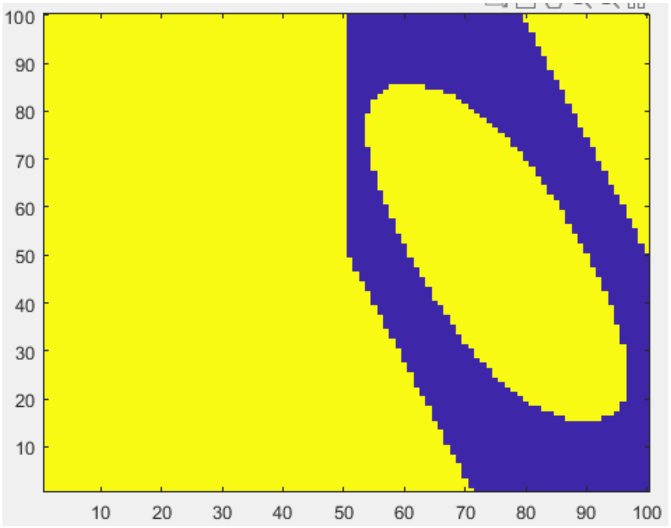
Modify cost in this area



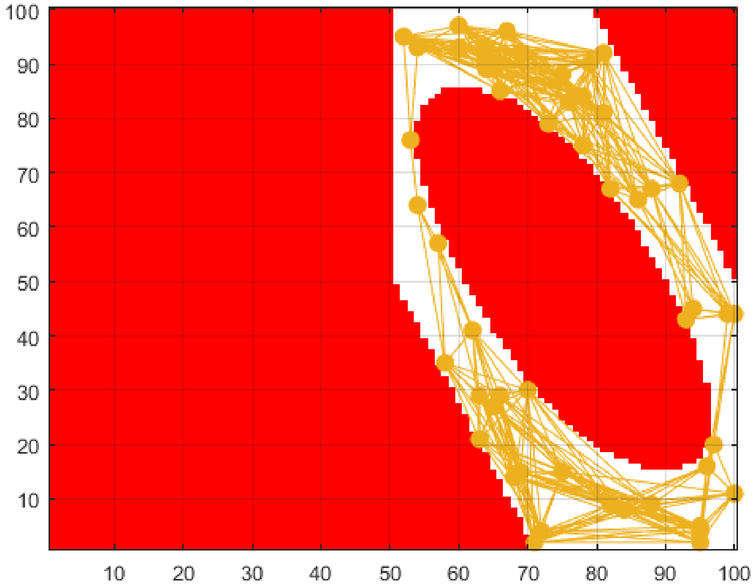
P. Corke, *Robotics, Vision and Control: Fundamental Algorithms In MATLAB® Second, Completely Revised, Extended And Updated Edition*. Springer, 2017.

Probabilistic Roadmap Method (PRM)

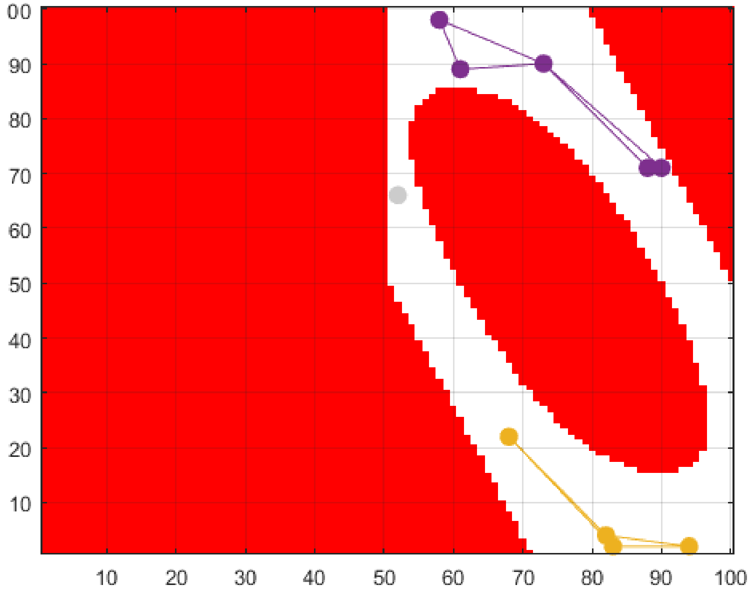
- ❑ The plan is independent of the goal
- ❑ Planning phase finds N random points
- ❑ Each point is connected to its nearest neighbours (distance threshold parameter of) by a straight-line path that does not cross any obstacles



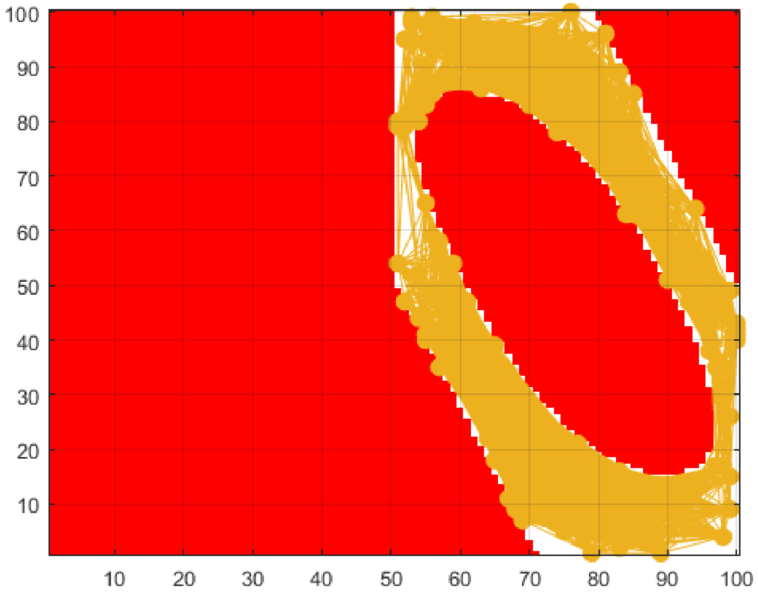
50 sampling points



10 sampling points



200 sampling points



Probabilistic Roadmap method (PRM)

- Toolbox PRM planner

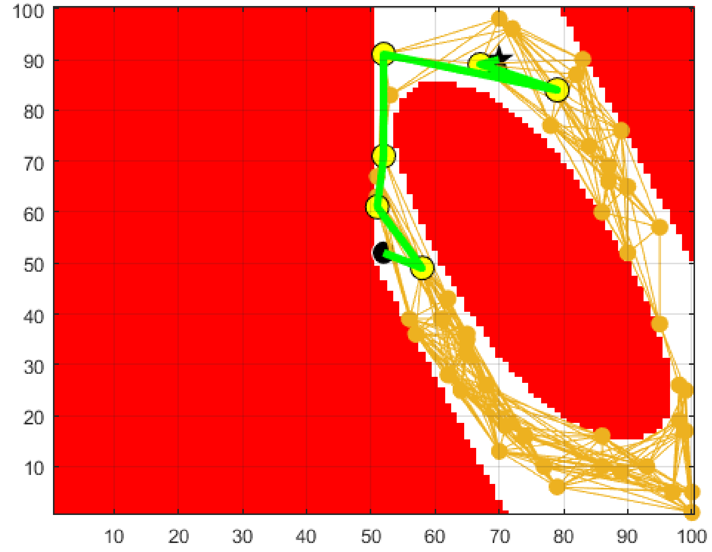
    prm = PRM(map) % Create PRM object

    prm.plan()   % Create Plan

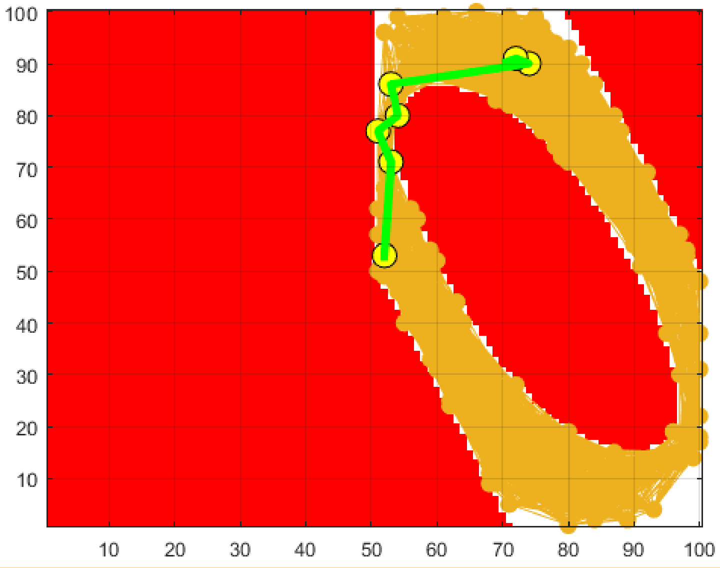
    prm.plot() % Visualize graph

    prm.query(start, goal) % Query phase:

Moving to the closest node in the network then follow the existing path



50 points



200 points

III. Artificial Potential Fields

Artificial Potential Fields (APFs)

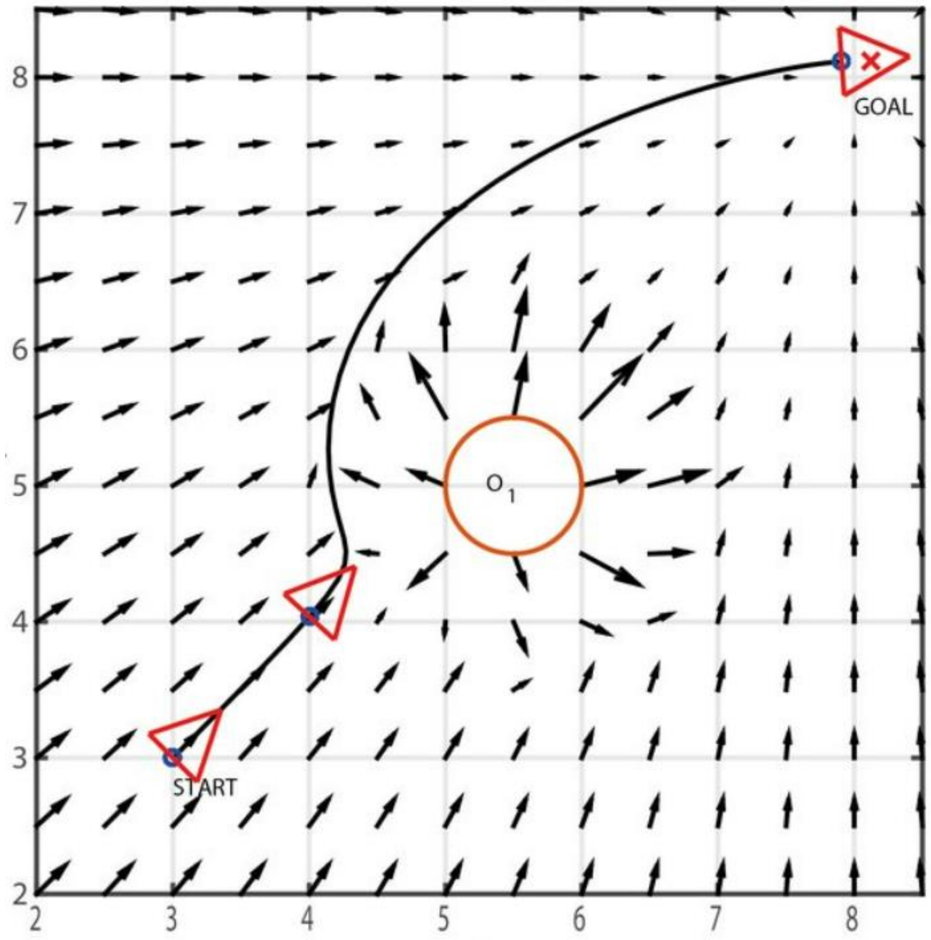
- Treat the robot's configuration as a point in a workspace potential field that combines attraction (to the goal) and repulsion (from obstacles) forces.

$$U(q) = U_{att}(q) + U_{rep}(q)$$

- Use **gradient descent** on the field to define a path (i.e., the negative gradient of *U* can be considered as a force acting on the robot)

$$F = -\nabla U(q) = -\nabla U_{att}(q) - \nabla U_{rep}(q)$$

Artificial Potential Fields (APFs)



Gradient descent

Source: Obstacles avoidance based on switching potential functions. Journal of Intelligent & Robotic Systems, 90(3), pp.387-405.

Attractive Potential

- $U_{att,i}(q)$ should be monotonically increasing with distance from q_f
- The simplest choice is a field that grows linearly with the distance from q_f(i.e., **Conic Well Potential**)

$$\boxed{U_{att,i}(q) = \zeta_i \parallel o_i(q) - o_i(q_f) \parallel}$$

where ζ_i is used as a scaling factor. Denote the position of the origin of the i^{th} DH frame by $o_i(q)$ and $o_i(q_f)$ as the goal configuration.

- The gradient has unit magnitude everywhere but the origin, where it is zero. This can lead to stability problems

$$\boxed{F_{att}(q) = -\nabla U_{att}(q) = -\zeta_i}$$

Attractive Potential

- A quadratic field (so as to be continuously differentiable), called the **Parabolic Well Potential** is defined as,

$$U_{att,i}(q) = \frac{1}{2}\zeta_i \parallel o_i(q) - o_i(q_f) \parallel^2$$

ζ_i : a scaling factor.

- The attractive force for o_i is equal to its negative gradient,

$$F_{att,i}(q) = -\nabla U_{att,i}(q) = -\zeta_i(o_i(q) - o_i(q_f))$$

- At long distances, a conic potential can be used instead of the quadratic field to avoid a too large attractive force

Source: Spong et al. Robot Modeling and Control, 2005

Attractive Potential

- A combination of **Parabolic Well Potential** and **Conic Well Potential**

$$U_{\mathrm{att},i}(q) = \begin{cases} \frac{1}{2}\zeta_i \|o_i(q) - o_i(q_{\mathrm{f}})\|^2 & : \quad \|o_i(q) - o_i(q_{\mathrm{f}})\| \le d \\ d\zeta_i \|o_i(q) - o_i(q_{\mathrm{f}})\| - \dfrac{1}{2}\zeta_i d^2 & : \quad \|o_i(q) - o_i(q_{\mathrm{f}})\| > d \end{cases}$$

- The attractive force is

$$F_{\mathrm{att},i}(q) \quad = \quad \begin{cases} -\zeta_i (o_i(q) - o_i(q_{\mathrm{f}})) & : \quad \|o_i(q) - o_i(q_{\mathrm{f}})\| \le d \\ -d\zeta_i \dfrac{(o_i(q) - o_i(q_{\mathrm{f}}))}{\|o_i(q) - o_i(q_{\mathrm{f}})\|} & : \quad \|o_i(q) - o_i(q_{\mathrm{f}})\| > d \end{cases}$$

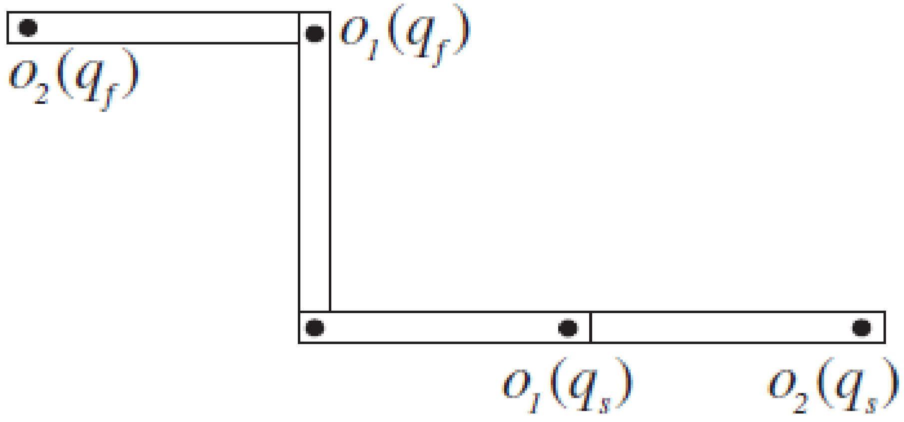
- The force is continuous at the boundary (*d*)

Source: Spong et al. Robot Modeling and Control, 2005

Example 1 - Attractive Potential for 2 Link Arm

Arm lengths are a1 = a2 = 1with initial and final configurations given by

$$q_s = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad q_f = \begin{bmatrix} \frac{\pi}{2} \\ \frac{\pi}{2} \end{bmatrix}$$



Calculate the attractive for if a parabolic well potential is used and the scaling factor ζ_1 and ζ_2 are 1 and 2 respectively.

Answer:

$$F_{att,i}(q) = -\nabla U_{att,i}(q) = -\zeta_i\big(o_i(q) - o_i(q_f)\big)$$

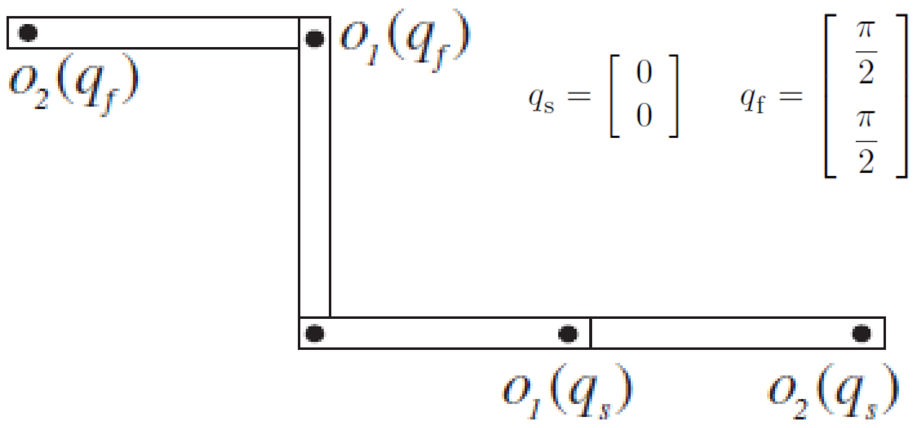
We need to find o_i(start) and o_i(final)

-> Use forward kinematics

Source: Spong et al. Robot Modeling and Control, 2005

Example 1 - Attractive Potential for 2 Link Arm

- Origin of the i^{th} DH frame by $o_i(q)$



$$o_1(q_s) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad o_1(q_f) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$o_2(q_s) = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \qquad o_2(q_f) = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

$$\boxed{F_{att,i}(q) = -\zeta_i\big(o_i(q) - o_i(q_f)\big)}$$

- Attractive force at O_1

$$F_{\text{att},1}(q_s) = -\zeta_1(o_1(q_s) - o_1(q_f)) = \zeta_1 \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

- Attractive force at O_2

$$F_{\text{att},2}(q_s) = -\zeta_2(o_2(q_s) - o_2(q_f)) = \zeta_2 \begin{bmatrix} -3 \\ 1 \end{bmatrix}$$

Repulsive Potential

- ❑ Repel robot from nearby obstacle
- ❑ ρ_0 is the distance of influence of an obstacle
- ❑ $\rho(o_i(q))$ is the shortest distance between o_i and any workspace obstacle

$$U_{rep,i}(q) = \begin{cases} \frac{1}{2}\eta_i\left(\frac{1}{\rho\left(o_i(q)\right)} - \frac{1}{\rho_0}\right)^2 & ; \quad \rho\left(o_i(q)\right) \leq \rho_0 \\ 0 & ; \quad \rho\left(o_i(q)\right) > \rho_0 \end{cases}$$

- η_i is a scaling factor

Source: Spong et al. Robot Modeling and Control, 2005

Repulsive Potential

- Hence the repulsive force is,

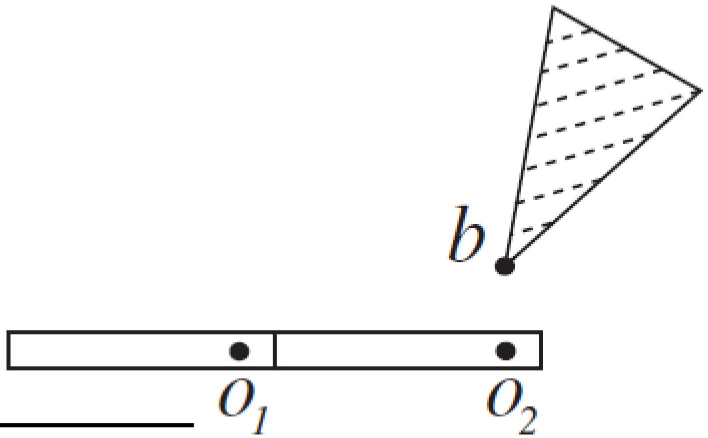
$$F_{rep,i}(q) = \eta_i \left(\frac{1}{\rho\big(o_i(q)\big)} - \frac{1}{\rho_0} \right) \frac{1}{\rho^2\big(o_i(q)\big)} \nabla \rho\big(o_i(q)\big)$$

- Note the gradient is not always continuous midway between obstacles

- Add floating repulsive control points on each link at closest position to nearest obstacle to avoid any part of the link colliding

Example 2 - Repulsive Force of Two-Link Arm

The vertex b of the triangle obstacle has the coordinates (2; 0.5). The coordinate of O_2(start) is (2; 0).
If $\rho_0 = 1$ and the scaling factor is $\eta_2 = 3$, calculate the repulsive force at O_2(start).



Answer:

Let x and y be the coordinate of O_2 $\quad \rho(O_2) = \sqrt{(x-2)^2+(y-0.5)^2}$

Distance from O_2(start) to b is $\rho(o_2(q_s)) = 0.5$

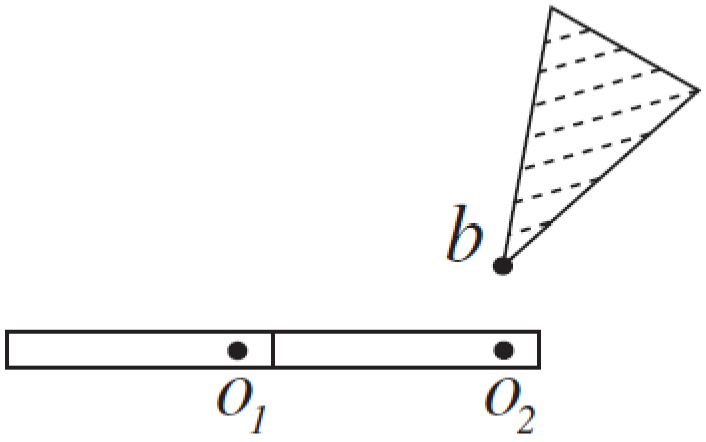
Gradient:

$$\nabla\rho(O_2) = \begin{pmatrix} \dfrac{x-2}{\sqrt{(x-2)^2+(y-0.5)^2}} \\ \dfrac{y-0.5}{\sqrt{(x-2)^2+(y-0.5)^2}} \end{pmatrix} \Rightarrow \nabla\rho(o_2(q_s)) = [0, -1]^T$$

Example 2 - Repulsive Force of Two-Link Arm

$$\rho(o_2(q_s)) = 0.5$$

$$\nabla\rho(o_2(q_s)) = [0, -1]^T$$



$$\boxed{F_{rep,i}(q) = \eta_i \left(\frac{1}{\rho(o_i(q))} - \frac{1}{\rho_0} \right) \frac{1}{\rho^2(o_i(q))} \nabla\rho(o_i(q))}$$

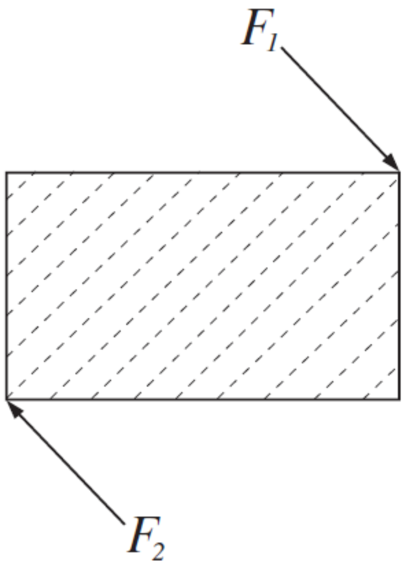
The repulsive force at $O_2(q_s)$ is

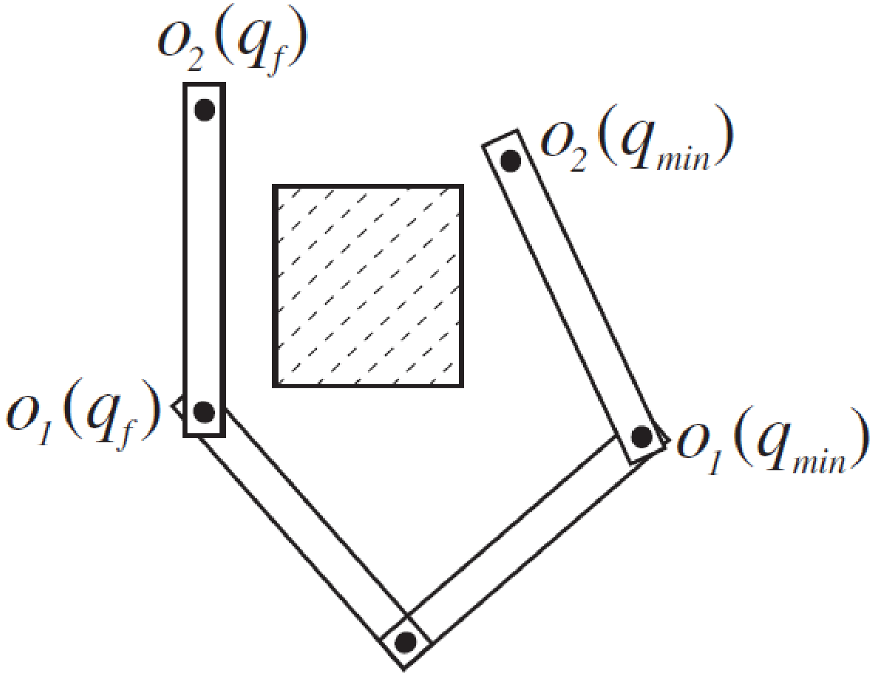
$$F_{\mathrm{rep},2}(q_s) = \eta_2 \left(\frac{1}{0.5} - 1 \right) \frac{1}{0.25} \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \eta_2 \begin{bmatrix} 0 \\ -4 \end{bmatrix}$$

Control with APFs – Using The Jacobian

- We **MUST use the joint torques** and not the workspace forces otherwise unexpected rigid body motion could occur, such as when two workspace forces induce a net moment

- Use the Jacobian to transform forces to joint torques





Local minima may also occur

Source: Spong, Mark W., Seth Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. Wiley, 2005.

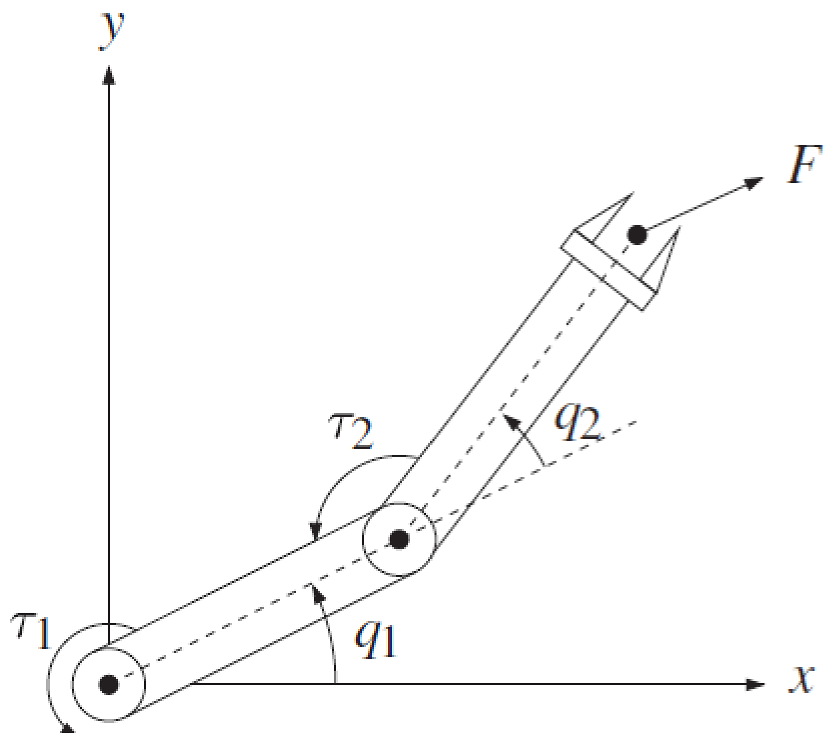
Control with APFs – Using The Jacobian

- The vector of forces and moments at the end effector

$$F = (F_x, F_y, F_z, n_x, n_y, n_z)$$

- The vector of joint torque

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



P. Corke, *Robotics, Vision and Control: Fundamental Algorithms In MATLAB® Second, Completely Revised, Extended And Updated Edition*. Springer, 2017.

Control with APFs – Using The Jacobian

- Can be applied directly to control.

$\boldsymbol{\tau}$: Vector of joint torques $$\boxed{\boldsymbol{\tau} = \boldsymbol{J}_v^T \boldsymbol{F}}$$ $\boldsymbol{F}$: workspace force at end effector

- **Total artificial joint torque** acting on the arm is the sum of the artificial joint torques that result from all attractive and repulsive potentials.

$$\boxed{\tau(q) = \sum_i J_{o_i}^T(q) F_{att,i}(q) + \sum_i J_{o_i}^T(q) F_{rep,i}(q)}$$

Source: Spong et al. Robot Modeling and Control, 2005

Example 3: Control with APFs

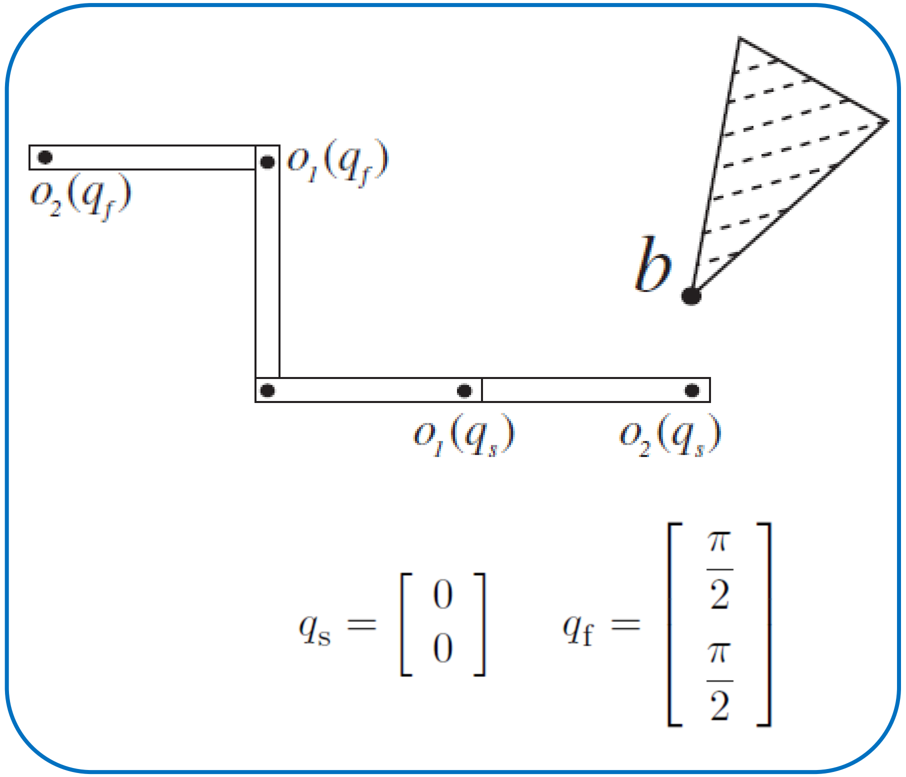
- A two-link arm with repulsive forces from an obstacle at (2, 0.5) and ρ_0 = 1 and q_f as per Example 1. Calculate the total artificial joint torque acting on the arm if $\eta_2 = \zeta_1 = \zeta_2 = 1$.

Answer

$$\tau(q) = \sum_i J_{o_i}^T(q) F_{att,i}(q) + \sum_i J_{o_i}^T(q) F_{rep,i}(q)$$

Step 1: Find the Jacobian for J_{o_1} and J_{o_2}

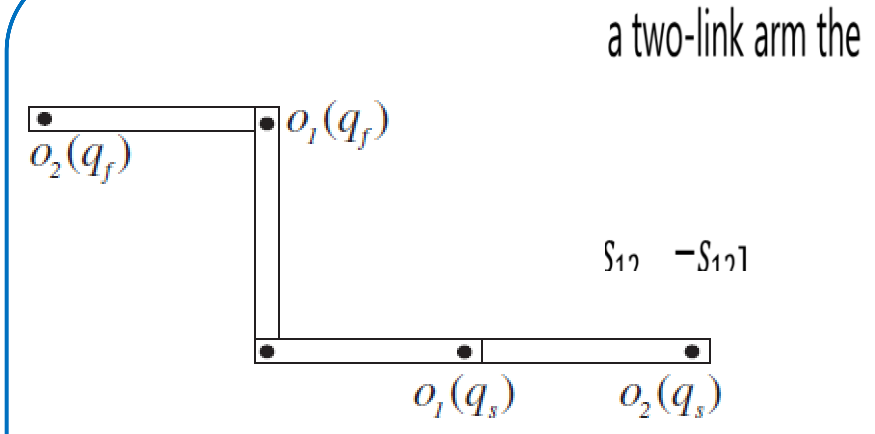
$$\dot{o}_i = J_{o_i}(q) \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}$$



Example 3: Control with APFs

- From lecture 5 (*Slide 36*), For a two-link arm the Jacobian matrix for o_2 is,

$$J_{o_2}(q_1, q_2) = \begin{bmatrix} -s_1 - s_{12} & -s_{12} \\ c_1 + c_{12} & c_{12} \end{bmatrix}$$



$$q_s = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad q_f = \begin{bmatrix} \frac{\pi}{2} \\ \frac{\pi}{2} \end{bmatrix}$$

Example: Control with APFs

- Using the similar method, the Jacobian matrix for o_1 is

$$J_{o_1}(q_1, q_2) = \begin{bmatrix} -s_1 & 0 \\ c_1 & 0 \end{bmatrix}$$

At $q_s = (0,0)$ we have,

$$J_{o_1}^T(q_s) = \begin{bmatrix} -s_1 & c_1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

And

$$J_{o_2}^T(q_s) = \begin{bmatrix} -s_1 - s_{12} & c_1 + c_{12} \\ -s_{12} & c_{12} \end{bmatrix} = \begin{bmatrix} 0 & 2 \\ 0 & 1 \end{bmatrix}$$

Example 3: Control with APFs

- The attractive potentials are (see example 1),

$$F_{att,1}(q_s) = -\zeta_1 \left(o_1(q_s) - o_1(q_f)\right) = \zeta_1 \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$
$$F_{att,2}(q_s) = -\zeta_2 \left(o_2(q_s) - o_2(q_f)\right) = \zeta_2 \begin{bmatrix} -3 \\ 1 \end{bmatrix}$$

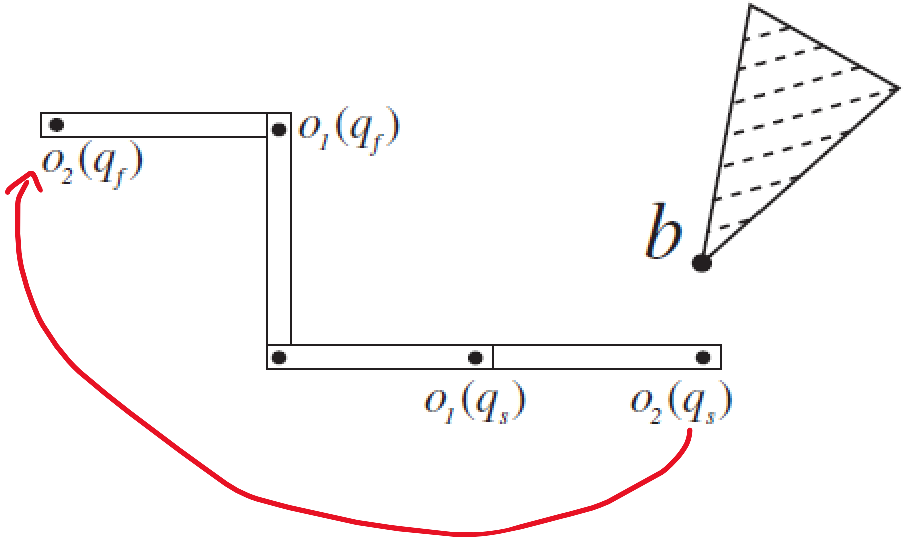
- The repulsive potential on link 2 is (see example 2),

$$F_{rep,2}(q_s) = \eta_2 \left(\frac{1}{0.5} - 1\right) \frac{1}{0.25} \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \eta_2 \begin{bmatrix} 0 \\ -4 \end{bmatrix}$$

Example 3: Control with APFs

- Substitute $\eta_2 = \zeta_1 = \zeta_2 = 1$, we can map the workspace attractive and repulsive forces to joint torques:

$$\tau_{att,1}(q_s) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$
$$\tau_{att,2}(q_s) = \begin{bmatrix} 0 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -3 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$
$$\tau_{rep,2}(q_s) = \begin{bmatrix} 0 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 4 \end{bmatrix} = \begin{bmatrix} -8 \\ -4 \end{bmatrix}$$



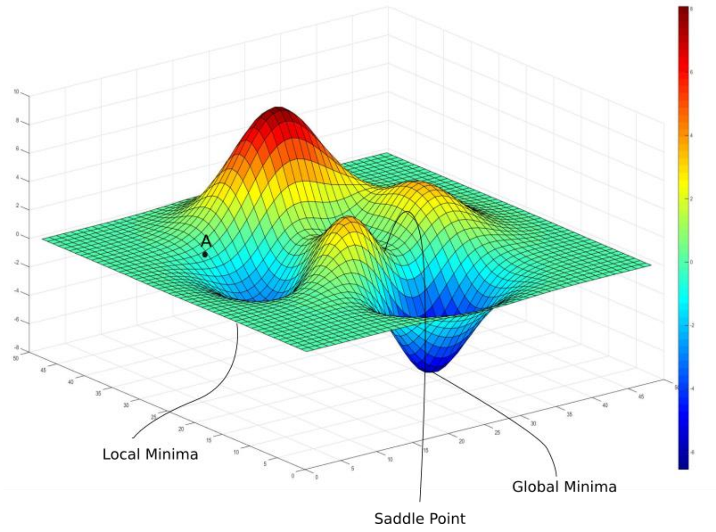
- The total joint torque induced given by,

$$\tau(q_s) = \tau_{att,1}(q_s) + \tau_{att,2}(q_s) + \tau_{rep,2}(q_s)$$

$$\tau(q_s) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix} + \begin{bmatrix} -8 \\ -4 \end{bmatrix} = \begin{bmatrix} -5 \\ -3 \end{bmatrix}$$

Hence both joints will rotate clockwise to avoid the obstacle

Critique of Artificial Potential Fields



- Efficient
- Easily adapted to the problem being solved
- Local minima
  - Harmonic Potential Fields can avoid this
  - Randomised Path Planner (RPP) includes random walks to avoid local minima (*Barraquand, 1991*)
  - See *Kavraki (1998)* and *Masoud (2013)* for a review of AFPs and Harmonic PFs respectively

https://bdtechtalks.com/2020/04/27/deep-learning-mode-connectivity-adversarial-attacks/gradient-descent-local-minima/

Next Week – Lecture 9 Robotics Dynamics

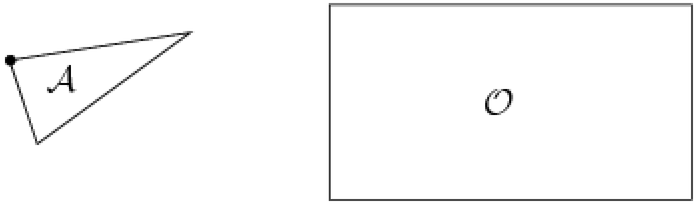
- The Lagrangian
- Euler-Lagrange Equations
- Applying Euler-Lagrange Method to an $\boldsymbol{n}$-link Manipulator

Appendix:
Configuration Space of non-point shaped robots

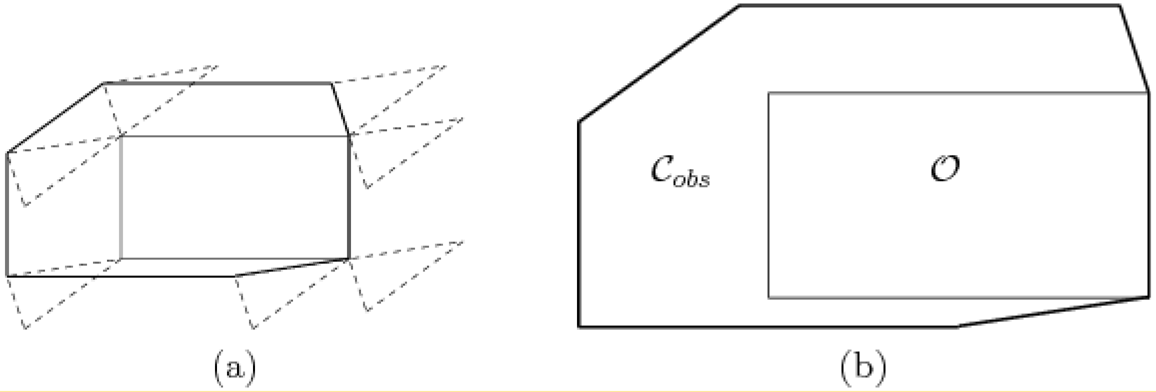
- Expanding $\mathcal{C}_{obst}$ for non-point shaped robots (translation only case shown below)
- This can be done in 3D for convex polygonal obstacles and robots but is cumbersome.

Consider a triangle-shaped mobile robot who possible motions include only translation in the plane

Robot: A
Obstacle: O



The boundary of $\mathcal{C}_{obst}$ can be obtained by computing the convex envelop of the configuration at which the robot makes vertex-to-vertex contact with obstacle



Source: Spong, Mark W., Seth Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. Wiley, 2005.