Note Title

Lecture - 12

2/21/2012

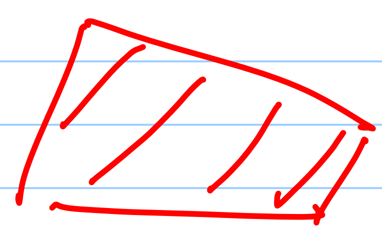
Path Planning

approaches: Roadmap based ✓

2) Cell decomposition ✓

3) Potential field based ⟵ today

Workspace "obstacles" as binary regions : free/occupied regions



In potential field based approach:

"obstacles" exert "repulsive" forces

goal " "attractive force"

In Electrical field analogy:

robot is a +ve charged particle. obs are +ve charged region

goal is a -ve charged region

or point.

"potential" field : robot moves under the influence of this field or forces.

$$U(q) : \mathbb{R}^N \longrightarrow \mathbb{R}$$

$$q : (x, y) \qquad \mathbb{R}^2 \longrightarrow \mathbb{R}$$

$$\vec{F}(q) : -\nabla U(q)$$

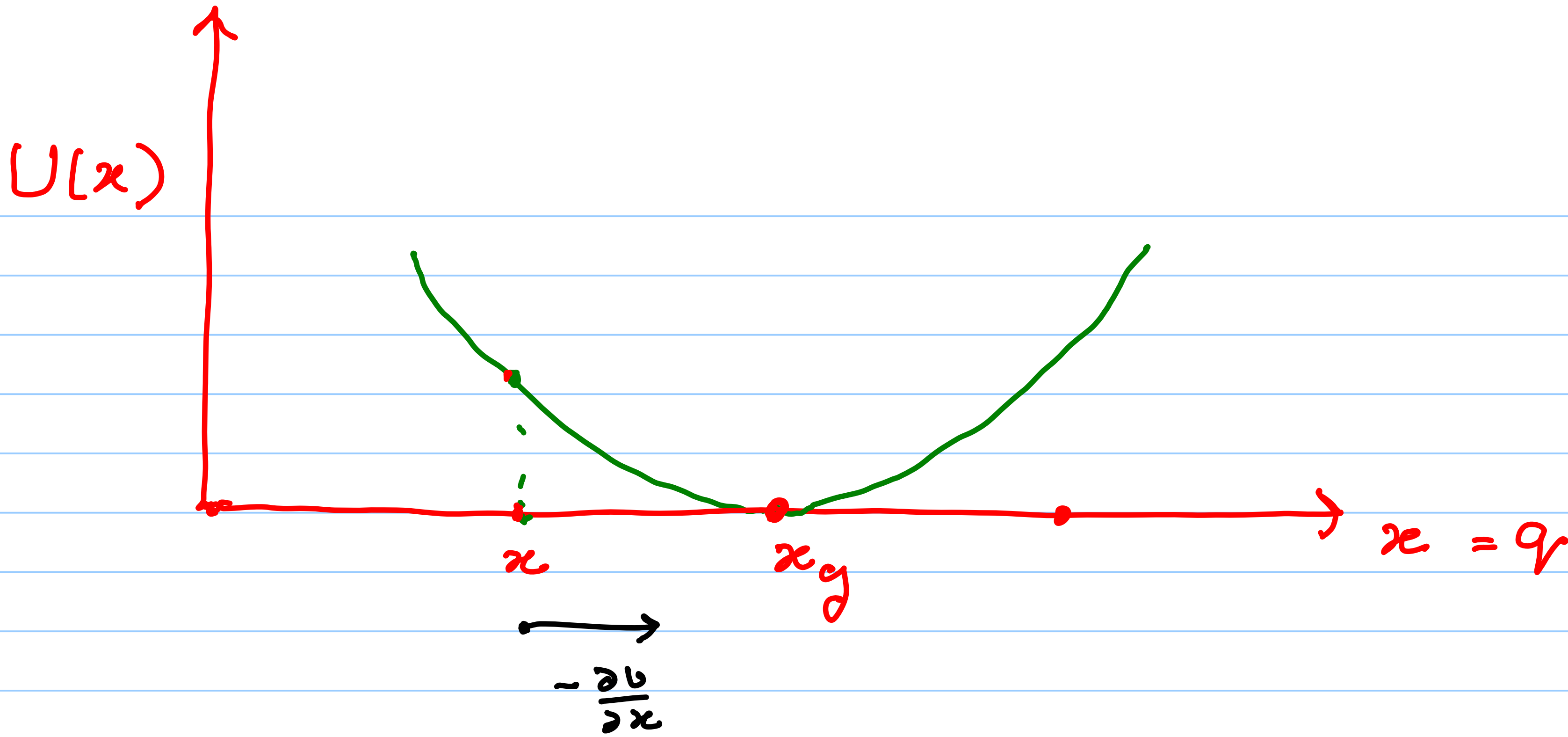
$$\nabla U(q) = \left[\frac{\partial U}{\partial q_1}, \frac{\partial U}{\partial q_2} \dots \frac{\partial U}{\partial q_N} \right)$$

robot is a particle moving under the influence of this potential field.

1) No physical laws that govern the pot. field in our case. we can therefore define pot. fields that suit our particular application.

2) Problem becomes that of finding the "minimum" of the pot. function (assume that goal is at a minimum)

Example: 1-D goal

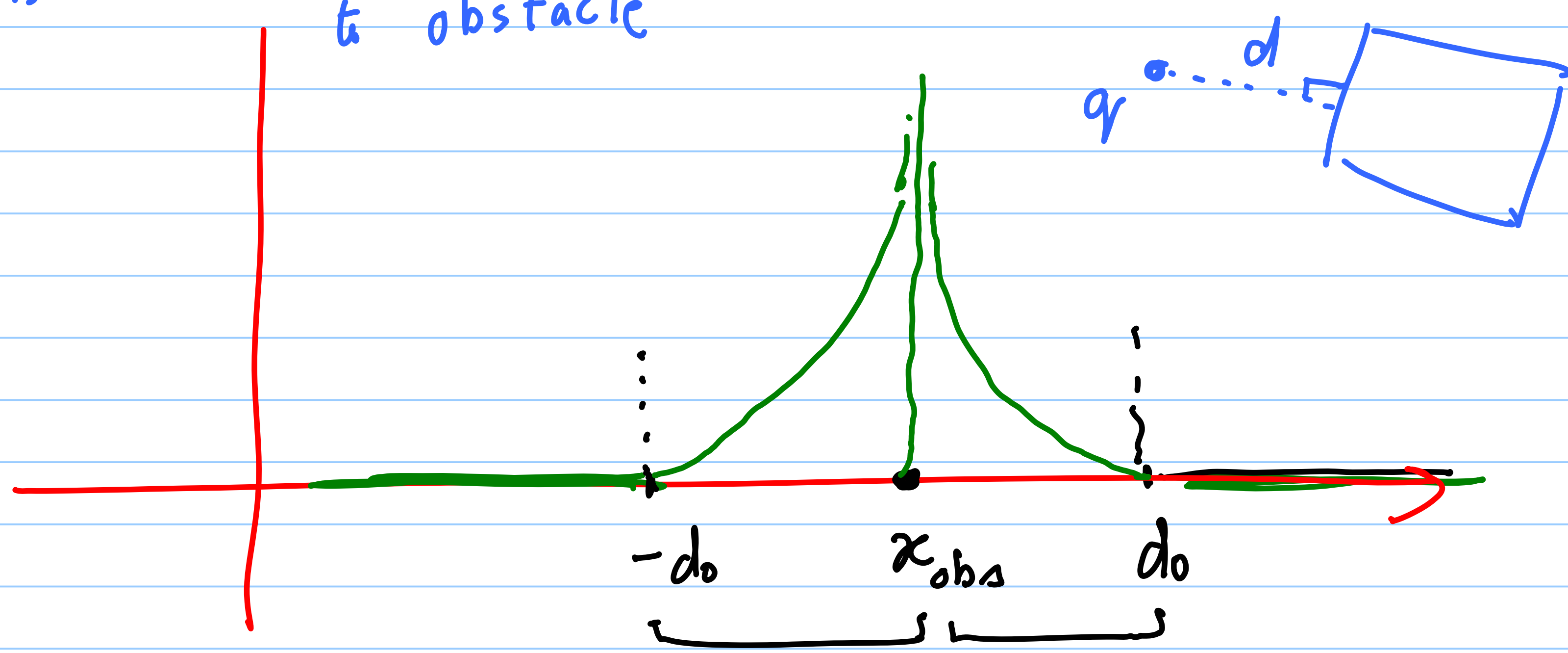


$$U_{att}(q) = \frac{1}{2}\,\zeta\, d^2(q, q_{goal})$$

$$\nabla U_{att}(q) = \zeta\,(q - q_{goal})$$ assuming Euclidean distance

$$U_{rep}(q) = \begin{cases} \frac{1}{2}\eta\left[\frac{1}{d(q,obs)} - \frac{1}{d_0}\right] & d(q) \le d_0 \\ 0 & \text{otherwise} \end{cases}$$

$d(q, obs)$ = shortest dist. to obstacle



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

Minimization :

Given $q(i)$

$$q(i+1) = q(i) + \alpha(i)\left[-\nabla U(q(i))\right]$$

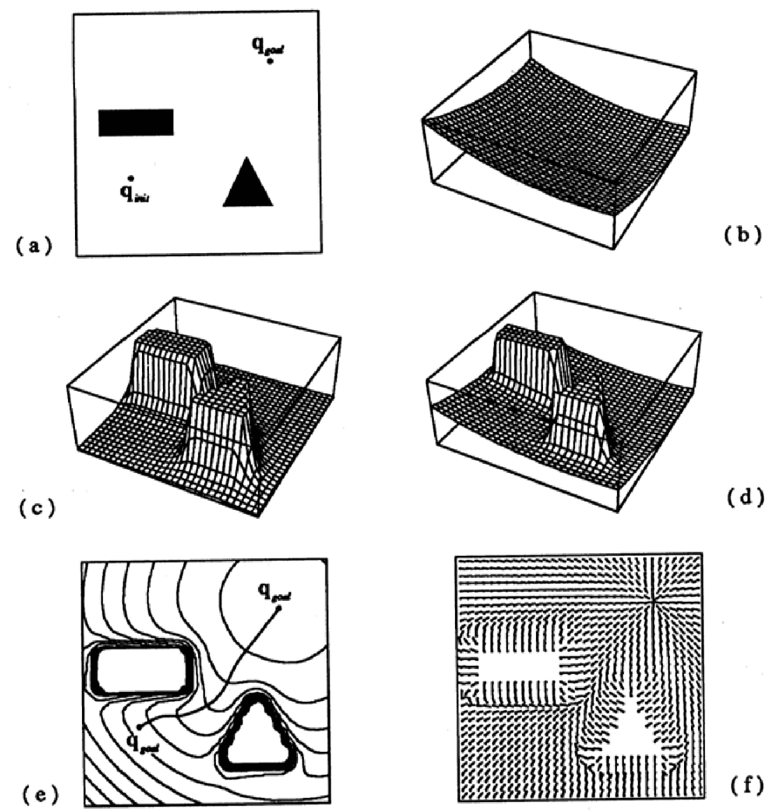


Figure 1. This figure shows an attractive potential field (Figure b), a repulsive potential field (Figure c) and the sum of the two (Figure d) in a two-dimensional

The re
easily
hedral
been
with
also e
In mo
rithm
ponen
$j = 1$
probl

Rem
tentia
2 of (
diagr
tic fu
poter
appr
path

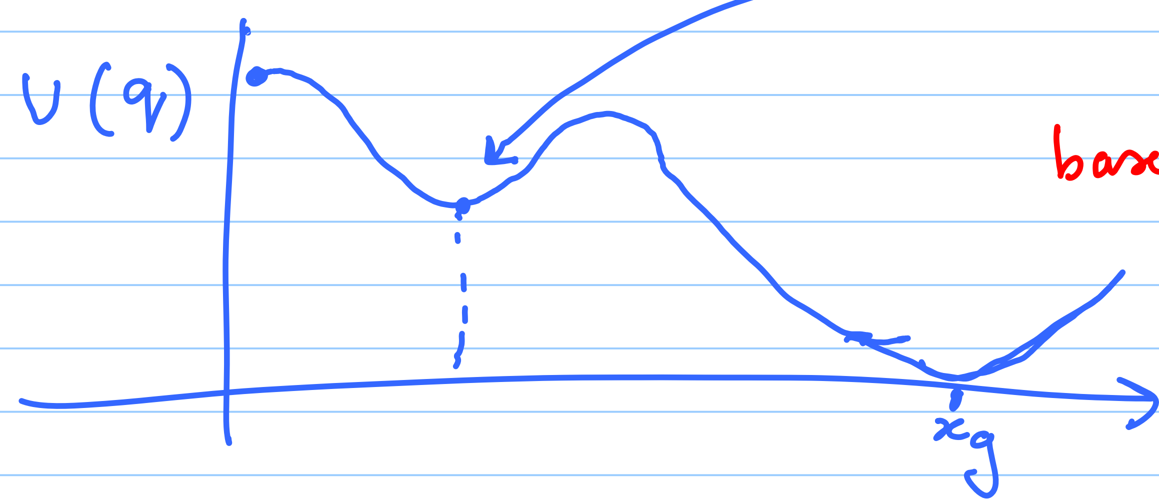
2

This
The
of th
read

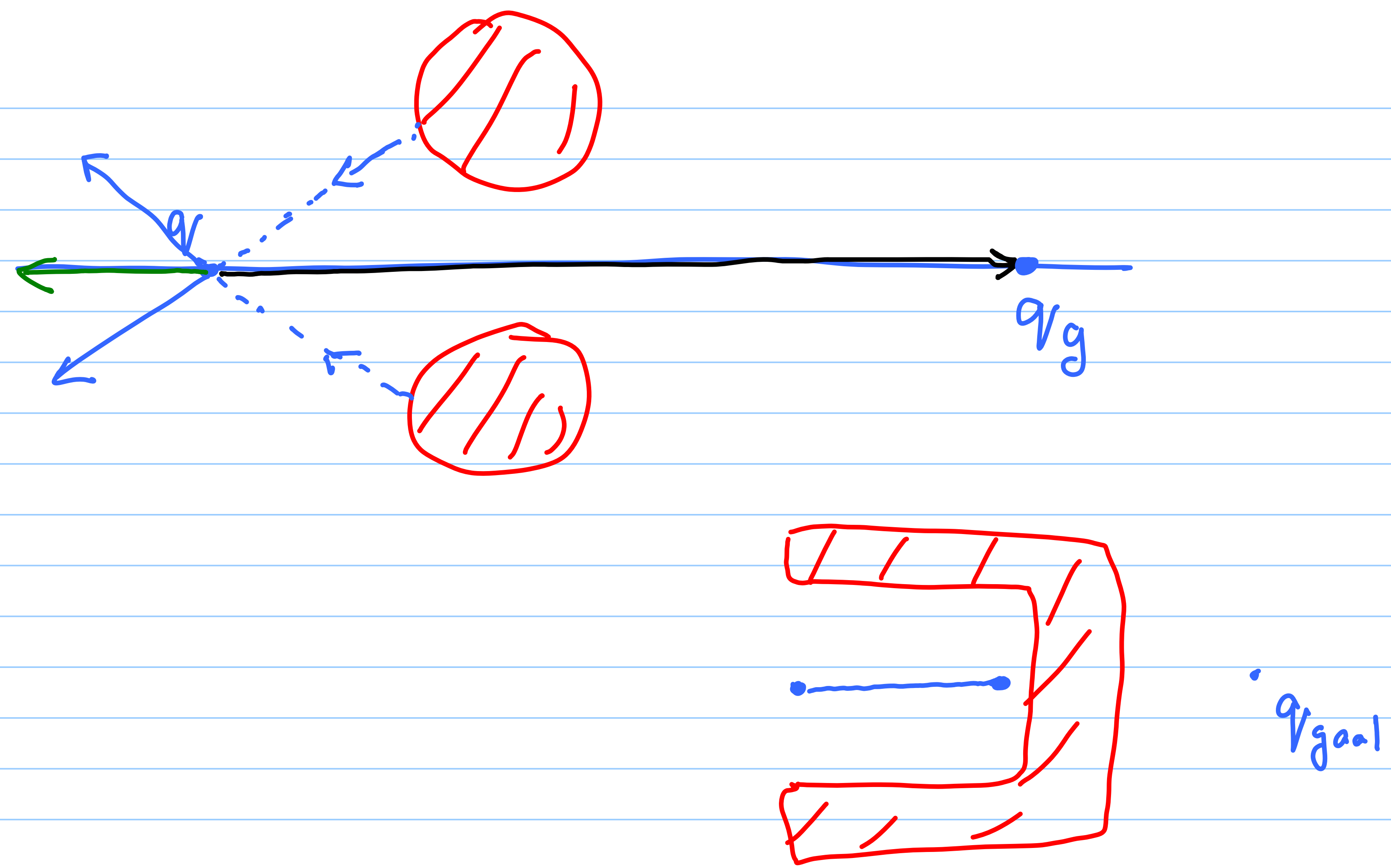
Issue 1:

Gradient search: "local minima"

a major issue in pot. field based approaches



Local minima examples:

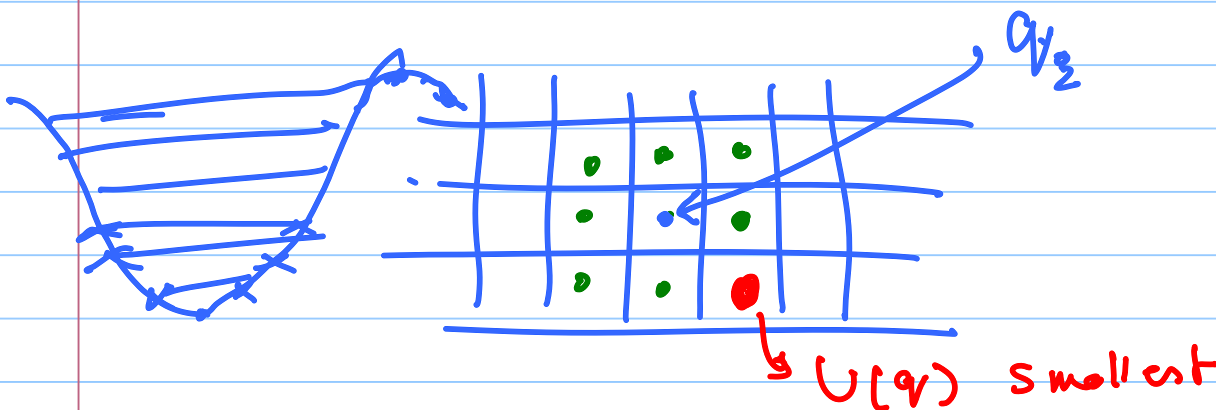


q
q_g
q_goal

Issue 2: scaling factors in potential field def: ζ, η : how to choose them? they govern the robot "velocity" — how fast/slow the robot moves

Alternative "Discretized" implementation of Gradient Search:

1) lay a grid over the C-space



"Best first search"

2) Evaluate $U(q)$ at all neighbor grid points of q_i (current config)

3) $q_i = q_{next}$: $U(q)\big|_{q_{next}}$ is lowest

to get out of local minima,

"fill the well" → computationally expensive

2^n : n = dim of C-space

Extension to non-point robots:

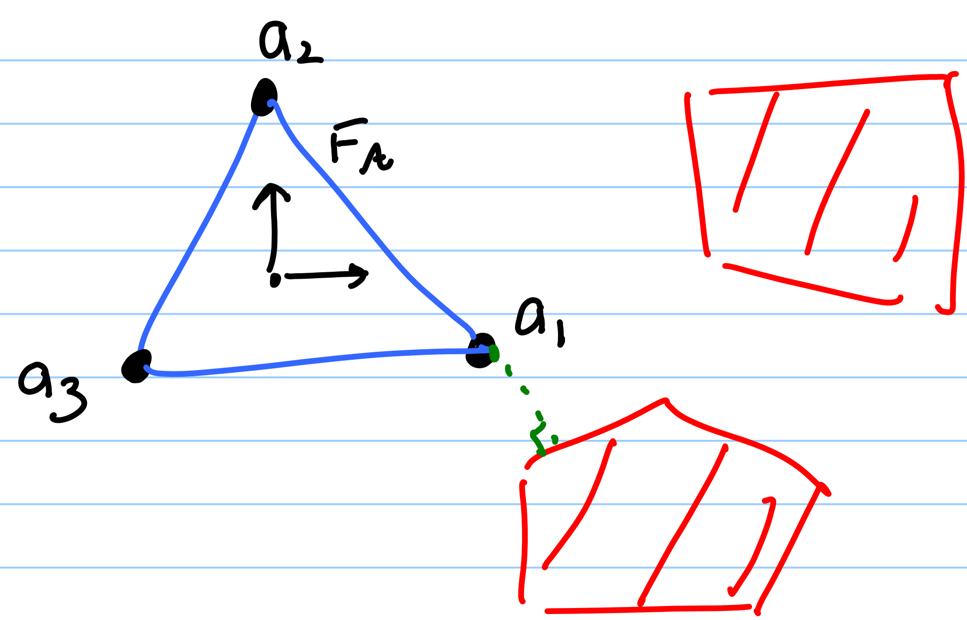
1) Translation only: polygon robot

C-obstacles (using minkowski sum)

$$CB = B \ominus A$$

use it to define $U(q)$

2) Trans. + Rotation:



$\mathbb{R}^N \times SO(N)$

$\downarrow$

gets computationally expensive to determin $d(q, obs)$

at each iteration in the minimization routine.

ANSWER:

1) define control pts on the robot

2) define potentials in workspace V (repulsive, att.)

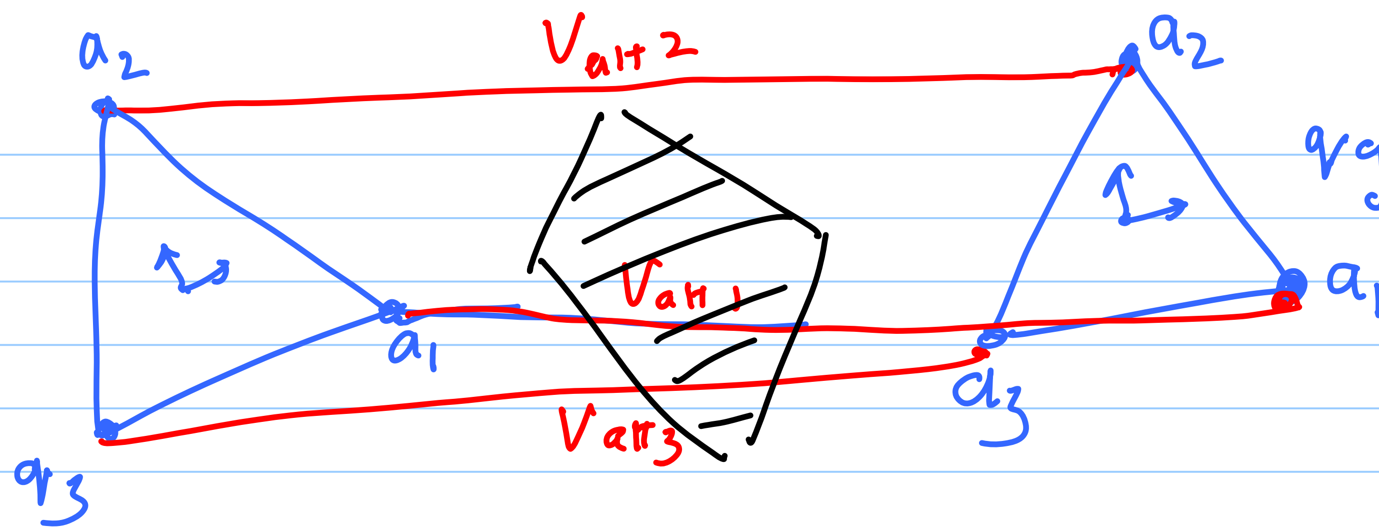
3) $V^j(q) = V(a_j(q))$

$U(q) = \sum_j V^j(q)$

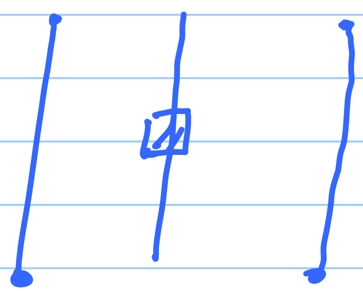
4) use best first search (BFS)

add. comp.

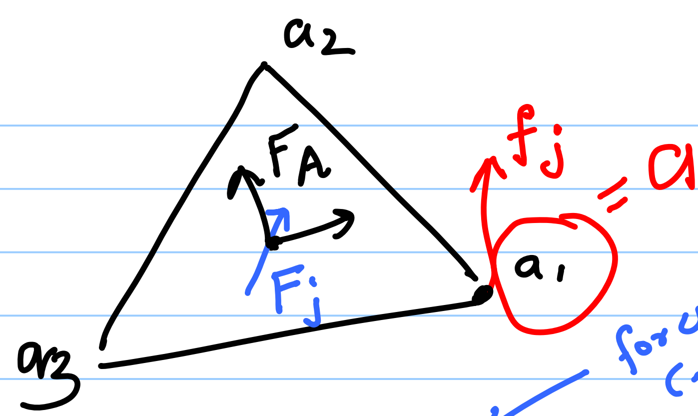
5) Collision check at each step in BFS



a2
qs
a1
Vatt 2
Vatt 1
Vatt 3
a2
qg
a1
a3



start
goal



$V^j = V(a_j(q))$

$\nabla V^j = \left(\frac{\partial V}{\partial x}, \frac{\partial V}{\partial y} \right)$

$\underline{x} = \begin{pmatrix} x \\ y \end{pmatrix}$

force in c-space

$$F(q) = -\sum_{j=1}^{3} J_j^T \nabla V^j$$

$f_j \cdot \delta \underline{x} = F_j \cdot \delta q$

in c-space

$f_j \cdot J \cdot \delta q = F_j \cdot \delta q$

work space

$f_j^T \delta x = F_j^T \delta q$

$$f_j^T \cancel{\delta} \delta q = F_j^T \delta q$$

$$\Rightarrow \boxed{F_j = J^T f_j}$$

<u>Navigation Function</u> : pot. func. that has a unique global minimum

No local minima.