August 18-21th, 2009
ICCAS-SICE 2009 Fukuoka, Japan

Unified motion planning method using random network and gradient method for multifunctional underwater robots

Koichiro Shiraishi and Hajime Kimura

Department of Maritime Engineering
Graduate School of Engineering
Kyushu University
Fukuoka, Japan

Outline of Presentation

1. Introduction

   Background and objectives

2. Formulation

   Motion planning problems

   for multifunctional underwater robots

3. Approach

   Unified motion planning method

   using random network and gradient method

4. Experiments

   Reaching task and thrust force generation task

5. Conclusions

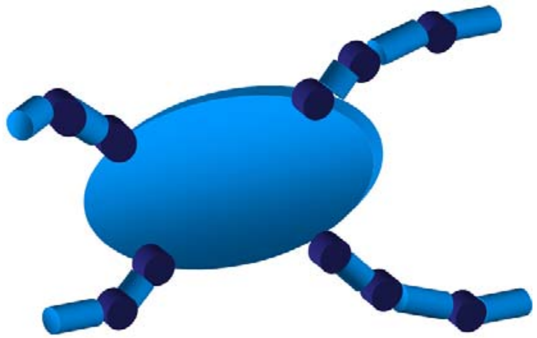
Outline of Presentation

1. Introduction
   Background and objectives
2. Formulation
   Motion planning problems
   for multifunctional underwater robots
3. Approach
   Unified motion planning method
   using random network and gradient method
4. Experiments
   Reaching task and thrust force generation task
5. Conclusions

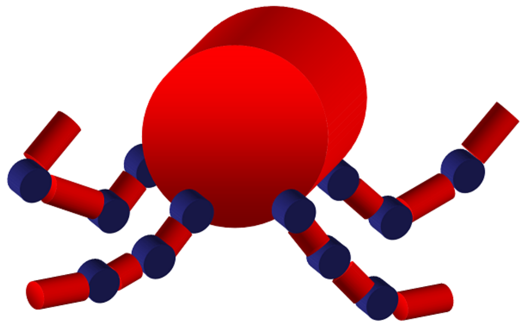
Introduction

Development of robots

Multifunctional underwater robot equipped manipulators



Driving beetle robot



Octopus robot

Features of these robots

- Having the ability of swimming, walking and grasping objects
- Using the manipulator as an arm, an leg, a finger and a fin

Necessary for the robot to generate
motion planning according to the various tasks

Problem in the motion planning

Conventional method

Necessary for the corresponding algorithms for individual tasks

➡ The motion planning require a large amount of time

Swimming task ← Motion planning algorithm for swimming task

Walking task ← Motion planning algorithm for walking task

Grasping task ← Motion planning algorithm for grasping task

➡ **Motion planning for each task is annoying**

Objective

Development of a unified motion planning method

To generate various motion planning using a single algorithm

Swimming task
Walking task
Grasping task
← Unified motion planning method

Outline of Presentation

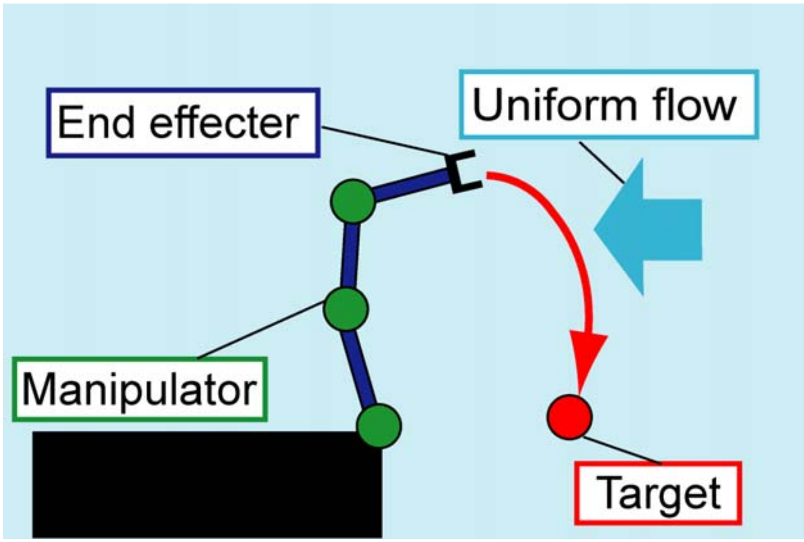
1. Introduction
   Background and objectives
2. Formulation
   Motion planning problems
   for multifunctional underwater robots
3. Approach
   Unified motion planning method
   using random network and gradient method
4. Experiments
   Reaching task and thrust force generation task
5. Conclusions

Motion Planning Problem

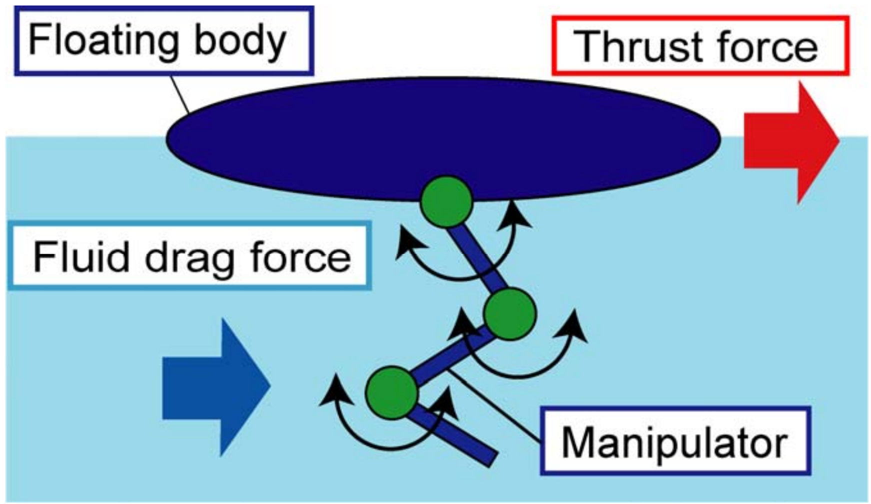
Target robot

Multifunctional underwater robot equipped the manipulator

Reaching Task



Thrust force generation task



Feature of Problems

Difference of cost functions according to target tasks

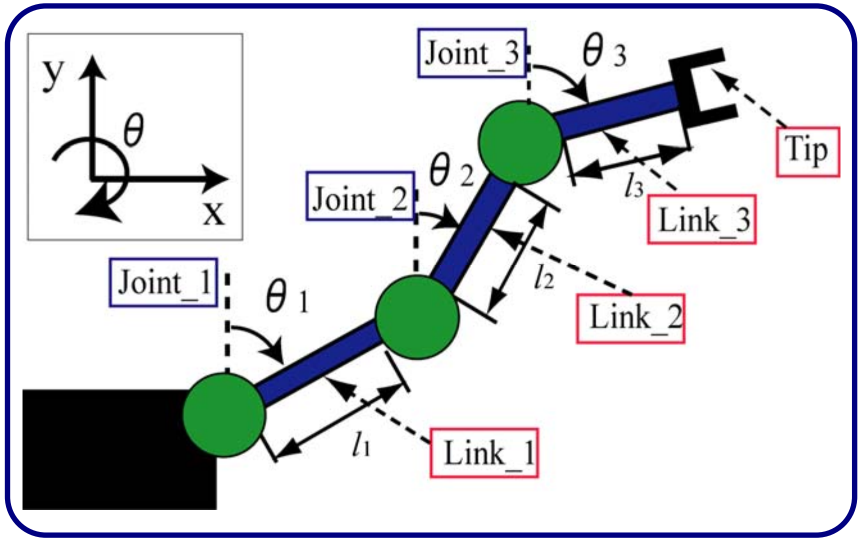
- Reaching task : Minimization of an energy consumption
- Thrust force generation task : Maximization of an advancing distance

Manipulator Coordinate System

- The coordinates of the i th link's tip : (x_i, y_i)

$$x_i = \sum_{j=1}^{i} l_j \sin\theta_j, \quad y_i = \sum_{j=1}^{i} l_j \cos\theta_j$$

$\left(\begin{array}{l} \theta_j \text{ : The angle of the } j \text{ th link} \\ l_j \text{ : the length of the } j \text{ th link} \end{array} \right)$



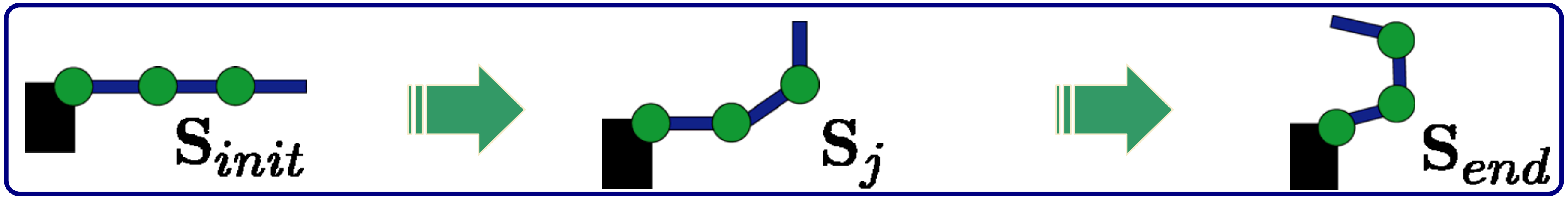
- Manipulator posture : $\mathbf{S}$

$$\mathbf{S} = (\theta_1, \theta_2, \cdots, \theta_i, \cdots, \theta_n)$$

- Motion planning : $\mathbf{P}$

$$\mathbf{P} = (\mathbf{S}_{init}, \mathbf{S}_1, \cdots, \mathbf{S}_j, \cdots, \mathbf{S}_{k-1}, \mathbf{S}_{end})$$

$\left(\begin{array}{l} \mathbf{S}_{init} \text{:the initial posture} \\ \mathbf{S}_{end} \text{:the target posture} \end{array} \right)$



Fluid Drag Force

Morrison's Equation

$$f = C_d \frac{1}{2} \rho D u|u| + C_m \rho \frac{\pi}{4} D^2 \frac{du}{dt}$$

(ρ: The fluid density
C_d: The drag coefficient
C_m: The added mass coefficient
D: The diameter of the column
u: The velocity of a link moving through the fluid)

The fluid drag force acting on each link

x direction

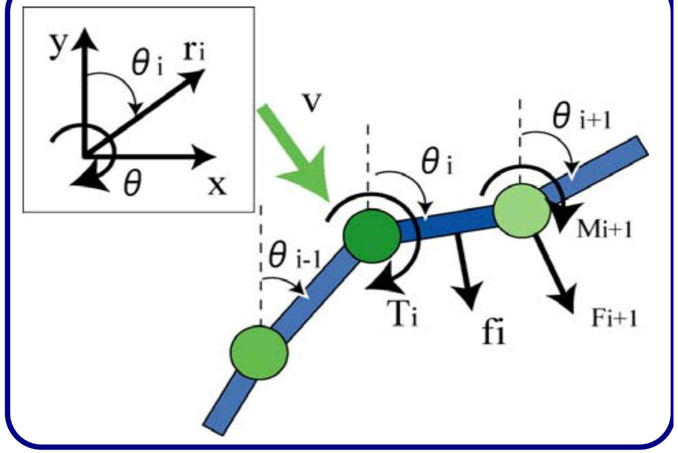
$$F_{x,i} = F_{x,i+1} + f_i \cos\theta_i$$

y direction

$$F_{y,i} = F_{y,i+1} - f_i \sin\theta_i$$

θ direction

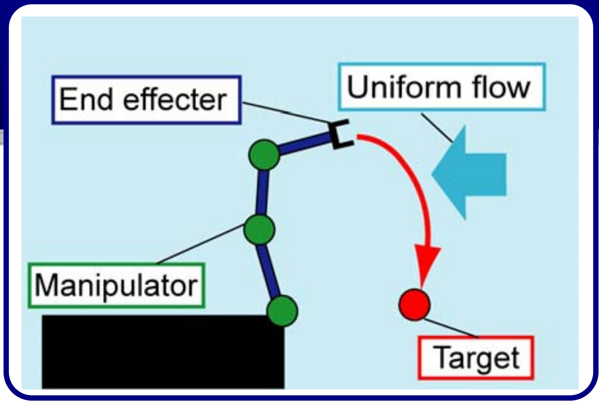
$$M_i = M_{i+1} + T_i + F_{x,i+1} l_i \cos\theta_i - F_{y,i+1} l_i \sin\theta_i$$



$$\left(f_i = \int_0^{l_i} C_d \frac{1}{2} \rho D u_i |u_i| dr_i, \quad T_i = \int_0^{l_i} C_d \frac{1}{2} \rho D u_i |u_i| r dr_i \right)$$

In our study, effects of vortices can be neglected.

Reaching Task



Cost function

The energy consumption
caused by the fluid drag force

$$Cost(\mathbf{S}_a, \mathbf{S}_b) = \sum_{i=1}^{N} \int_{\theta_{\mathbf{S}_{a,i}}}^{\theta_{\mathbf{S}_{b,i}}} M_i d\theta_i + \Delta E$$

$$\left(Total\_Cost(\mathbf{P}) = \sum_{i=1}^{|\mathbf{P}|-1} Cost(\mathbf{S}_i, \mathbf{S}_{i+1}) \right)$$

(ΔE: The energy consumption caused by mechanical friction)

Optimization problem

Formulating as an optimization problem to minimize the total cost

$$Optimal\ Planning = \min_{\mathbf{P} \in \mathbf{P}_{All}} Total\_Cost(\mathbf{P})$$

($\mathbf{P}_{All}$: The set of all executable motion planning in this problem)

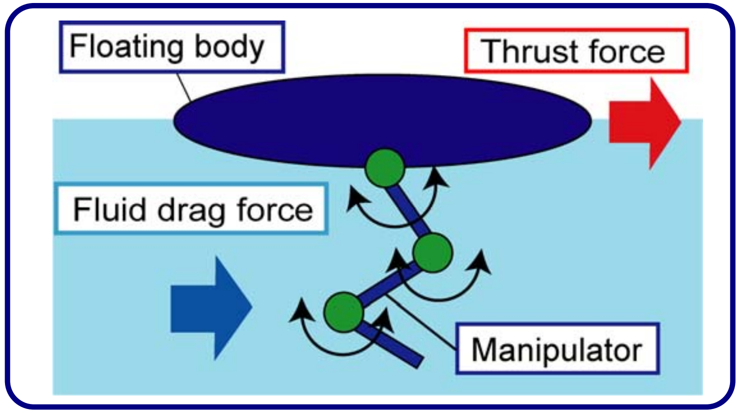
➡ The positive energy obtained from the current is not considered

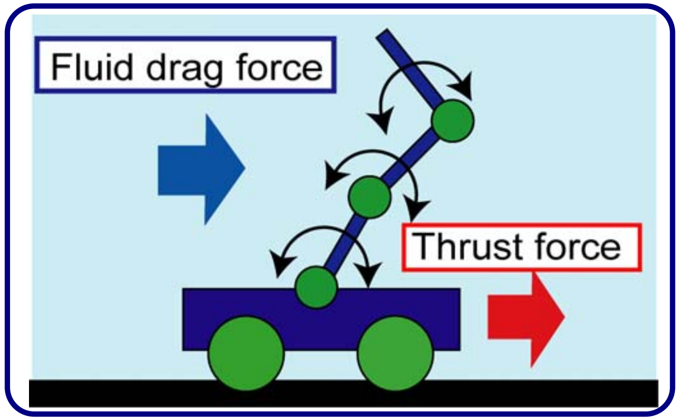
Consider as shortest path problems

Obtaining a path that minimizes a sum of a cost function

Thrust Force Generation Task

Modeling of the robot





Optimization problem

Formulating as an optimization problem to maximize the advanced distance

$$Optimal\ Planning = \max_{\mathbf{P} \in \mathbf{P}_{All}} \sum_{k=1}^{|\mathbf{P}|-1} L(S_k, S_{k+1})$$

($L(S_k, S_{k+1})$: a distance that the robot advances when the posture changes from S_k to S_{k+1})

Consider as a infinite horizon problem

Obtaining a policy to maximize an average reward in an infinite horizon

Outline of Presentation

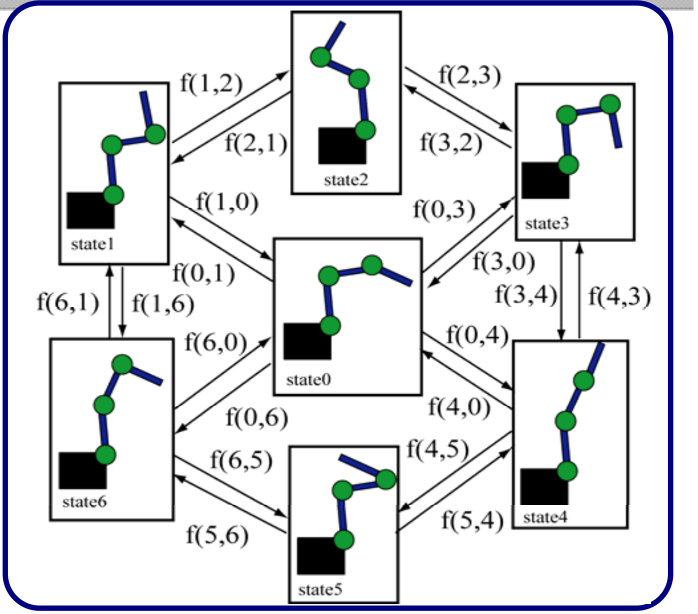
1. Introduction
   Background and objectives
2. Formulation
   Motion planning problems
   for multifunctional underwater robots
3. Approach
   Unified motion planning method
   using random network and gradient method
4. Experiments
   Reaching task and thrust force generation task
5. Conclusions

Modeling of the Motion Planning Problems

State transition network

Manipulator posture changes are represented as the network

- Node : A manipulator posture
- Arrow : A direction of the posture change
- $f(s_i, s_{i+1})$: A cost (or reward)



State transition network

This problem is considered as a graph search problem

Assuming that Markov property is confirmed

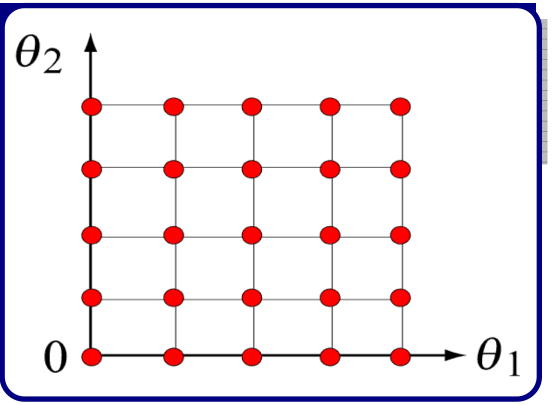
This problem is also modeled as a Markov decision process

The optimal policy is obtained using Dynamic programming

Discretization of State Space

Conventional technique

The state space is
discretized in a form of a lattice



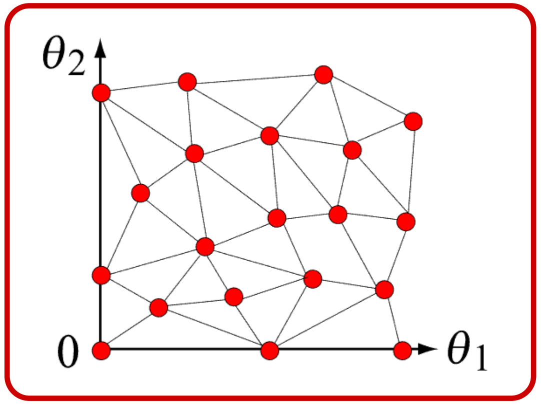
Lattice network

Suffering from the "Curse of the dimensionality"

- The number of states increases in exponentially
- The calculation using dynamic programming is intractable

Proposed technique

To avoid the curse of dimensionality,
the state space is discretized at random



Random network

Prevent the exponential increase in the number of states

- Adjusting the number of states is possible
- Well-balanced discretization of the state space is produced

⇒ **Degradation of the solution caused by a sparse network**

Modification Method

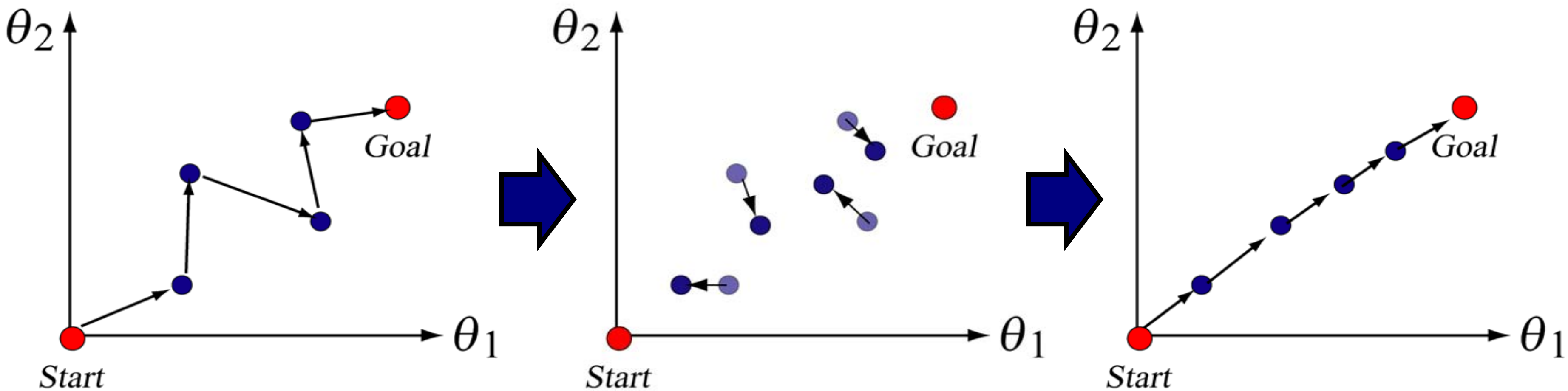
Objective

Modification of the obtained solution
with the sparse random network using a gradient method

➡ The locations of the nodes in the solution are modified using the gradient method to improve the quality of the solution

We can obtain good solutions in high dimensional problems

Illustration of the modification method



① Initial solution　② Modified solution using the gradient method　③ Improved solution

Outline of the Unified Motion Planning Method

1. Discretize the state space using the random network
2. Formulate the cost function according to the target task
3. Model the motion planning problem as the Markov decision process
4. Obtain the optimal value function using Dynamic programming
5. Determine the optimum policy according to the optimum value function
6. Modify the obtained solution using the gradient method

Obtaining the optimum motion planning according to the tasks using the single algorithm

Outline of Presentation

1. Introduction
   Background and objectives
2. Formulation
   Motion planning problems
   for multifunctional underwater robots
3. Approach
   Unified motion planning method
   using random network and gradient method
4. Experiments
   Reaching task and thrust force generation task
5. Conclusions

Experiment 1: Reaching Task

Setting for Experiment 1

- A 4-DOF manipulator was used.
- Parameters of the manipulator were as follows.

$l_i = 0.8[\mathrm{m}],\ D_i = 0.2[\mathrm{m}], C_d = 1.17, \theta_{limit} = \pm 1.571[\mathrm{rad}]$

(θ_{limit}:is the angle within which the links can rotate.)

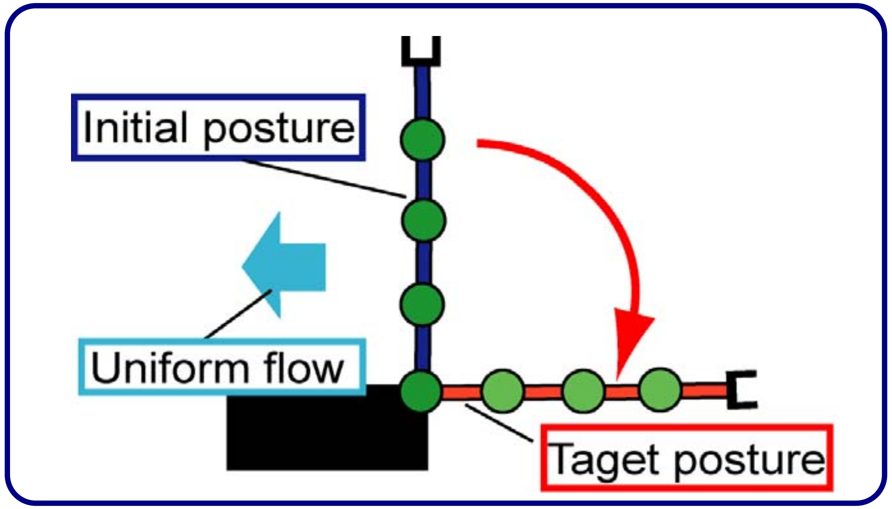
- Parameters of the underwater environment were as follows.

$\rho = 1.023, |u| = 2.0[\mathrm{m/sec}]$

- The direction of the uniform flow was the left .

Calculation Environment

CPU : Core2Duo E4500 2.2GHz
Memory : 2GB
OS : Windows XP
Program language： Java



Experiment 1: Reaching Task

Discretization of state space

	The number of nodes	The number of links per a node
Lattice network	28,561(=13^4)	80
Random network	10,000	40

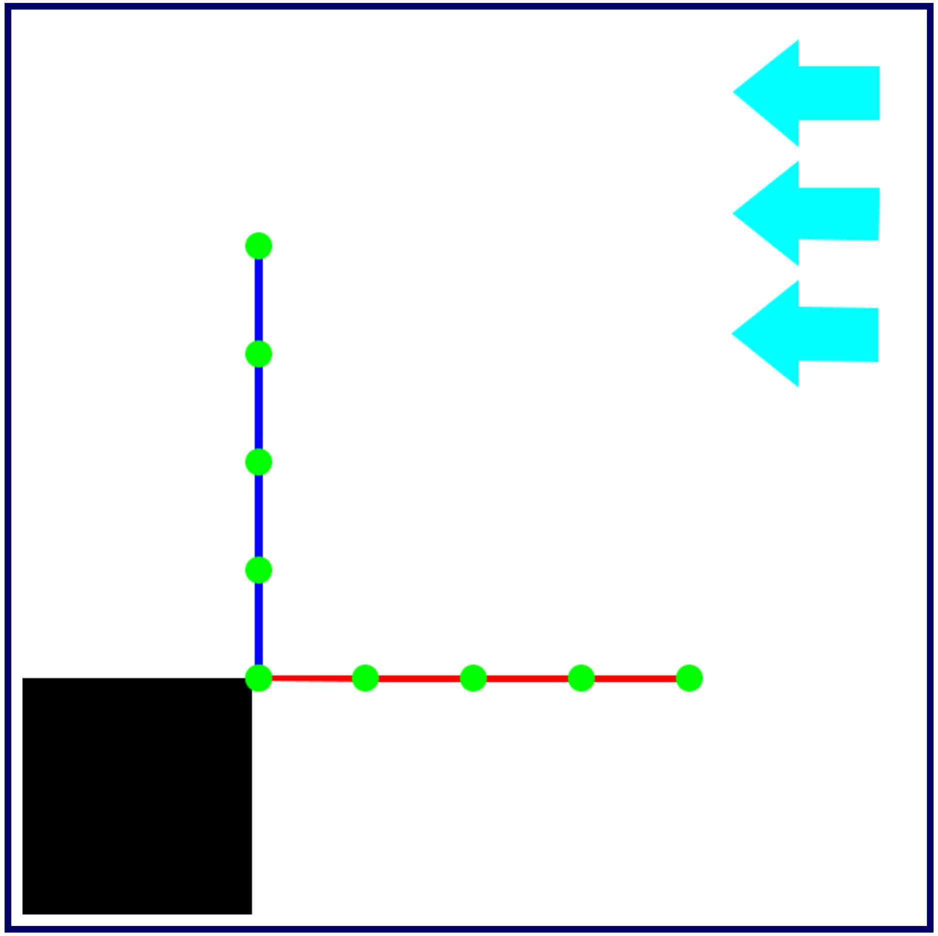
Motion planning sets

To enable comparison of the results,
three motion planning sets were used.

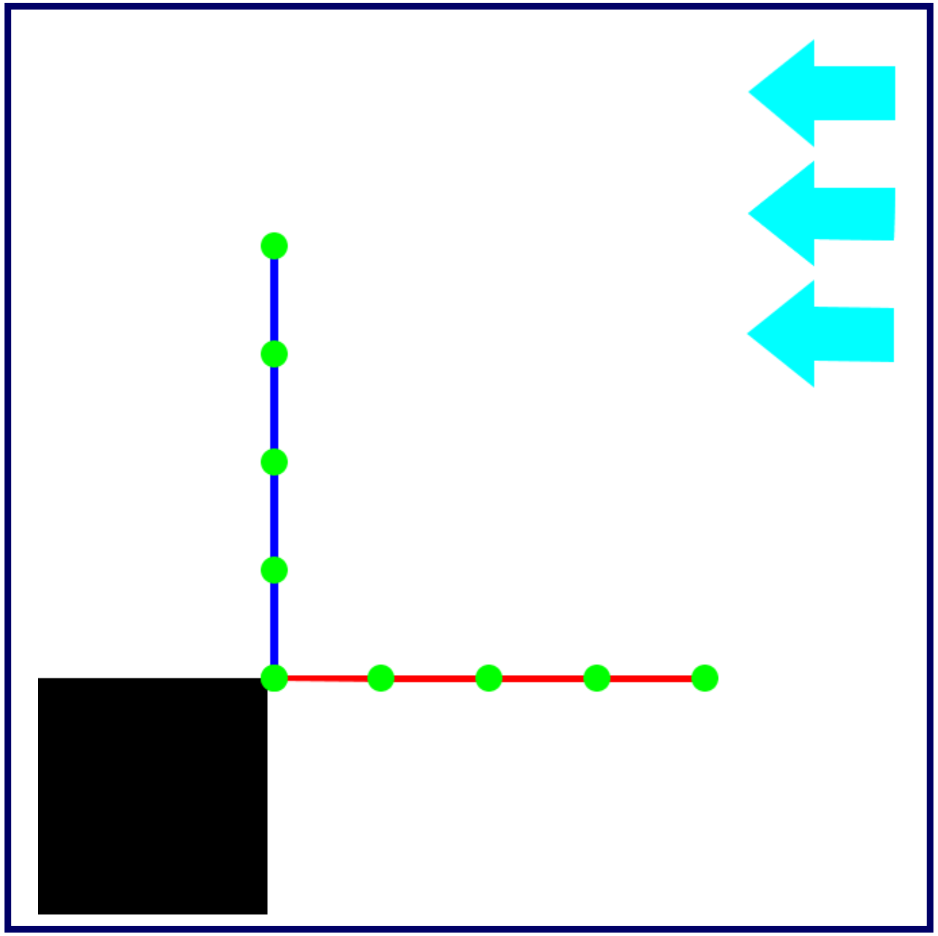
	Minimization	Network
Planning 0	Sum of rotation angle	—
Planning 1	Energy consumption	Lattice network
Planning 2	Energy consumption	Random network

Simulation was executed 10 times and the performance was evaluated from an average of the results

Results of Planning 0 (Conventional Method)



The motion of Planning 0

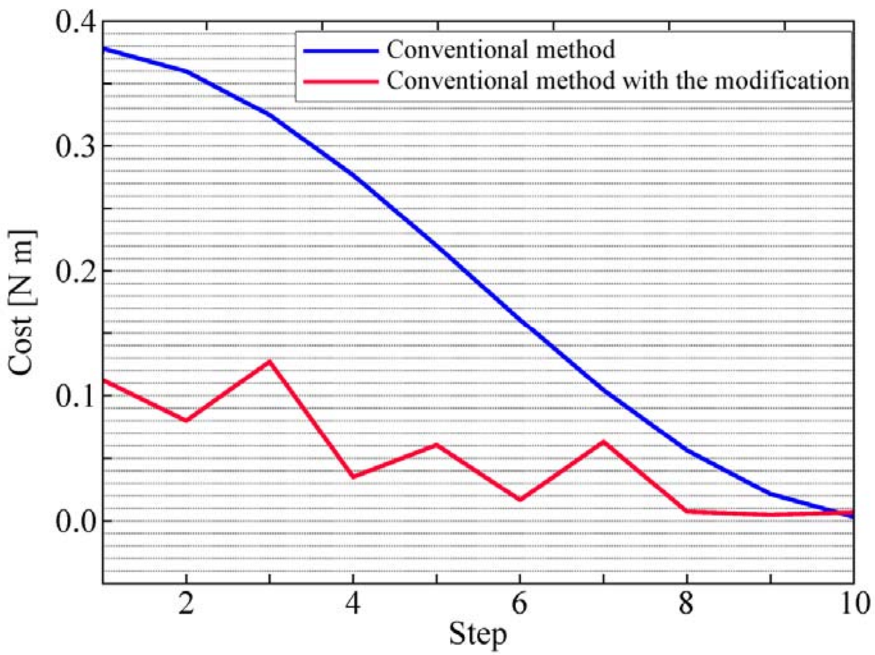


The motion of Planning 0 applied the modification

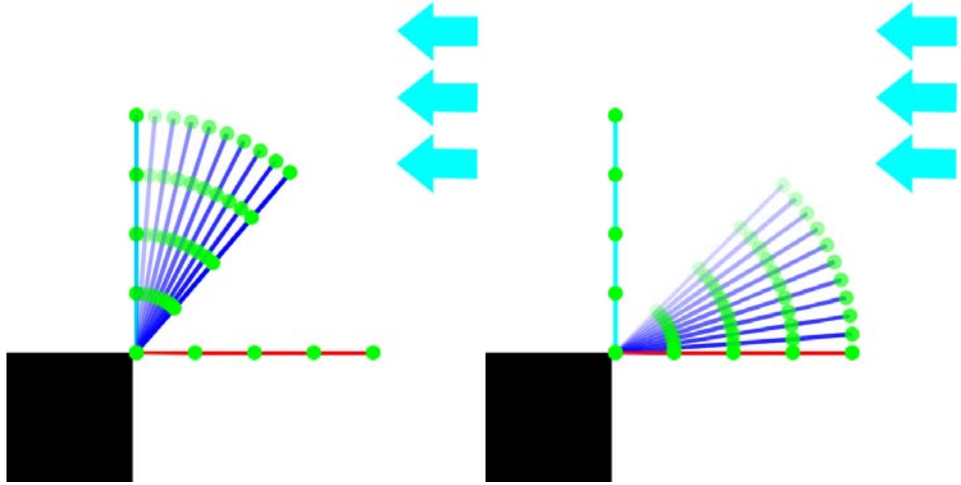
Results of Planning 0 (Conventional Method)

Method

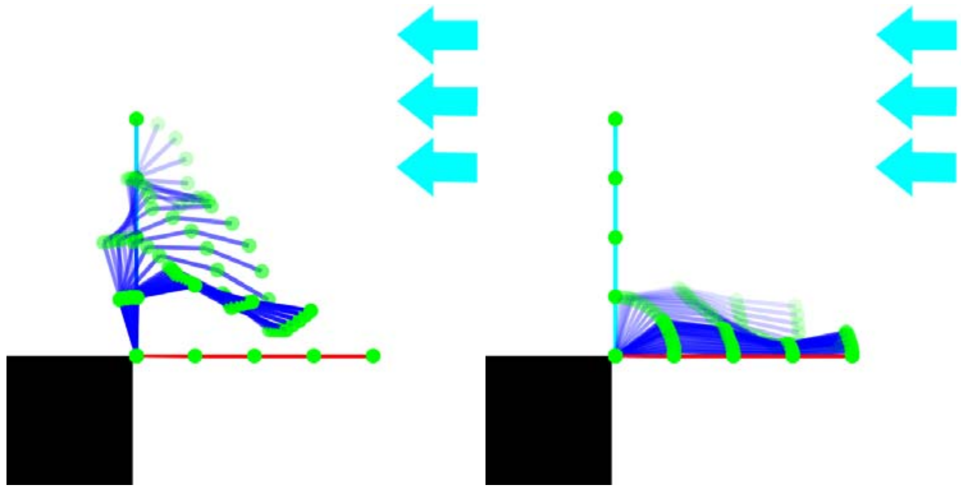
- Conventional method
  Minimization the sum of the rotation angle



Time series graph for the cost

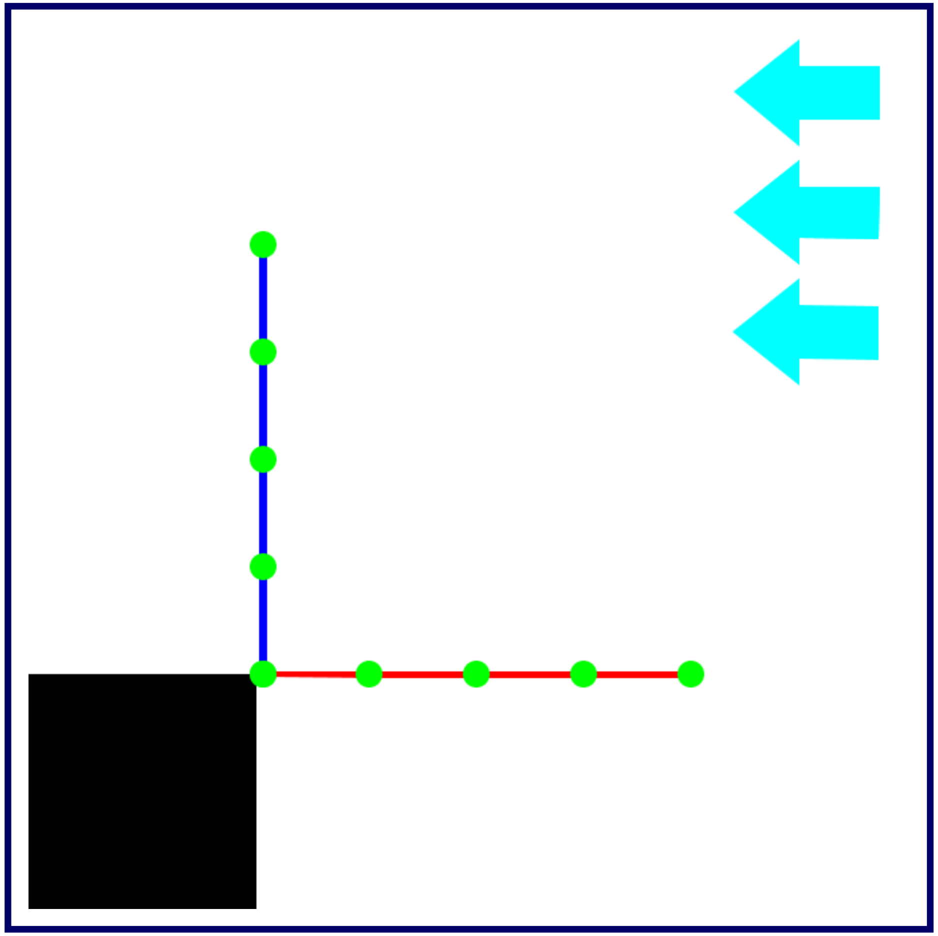


Trajectory of Planning 0

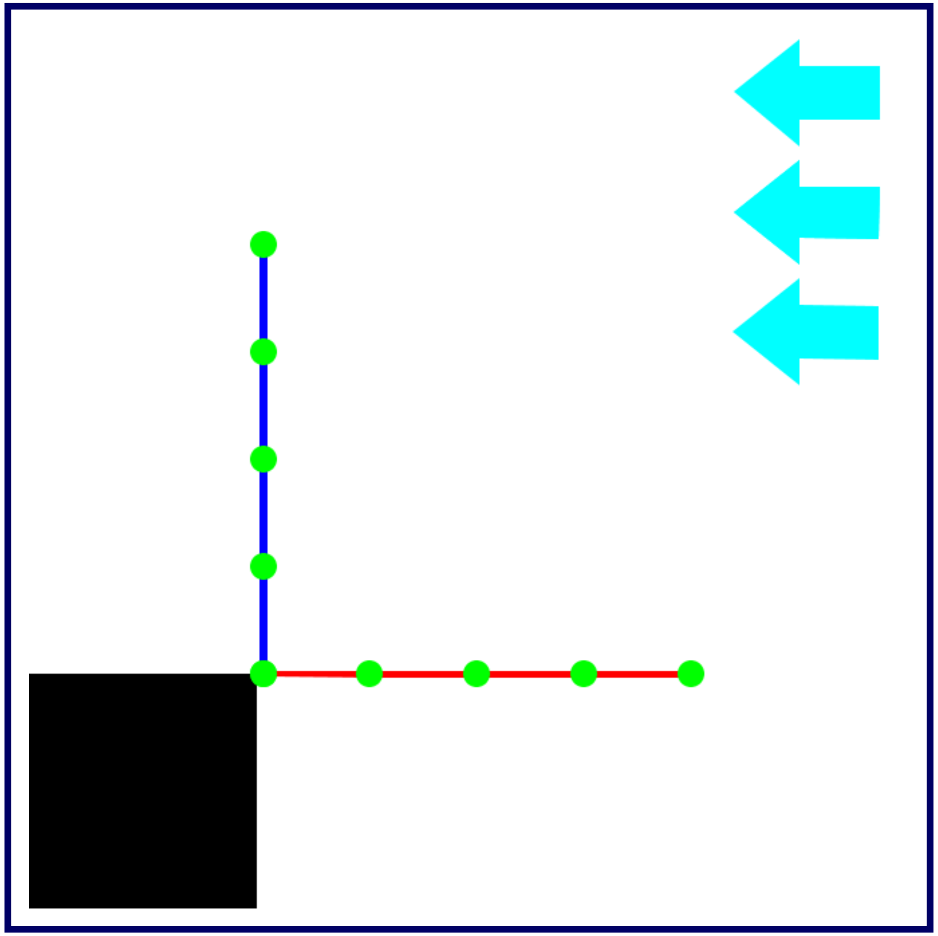


Trajectory of Planning 0 applied the modification

Results of Planning 1 (Energy, Lattice)



The motion of Planning 1

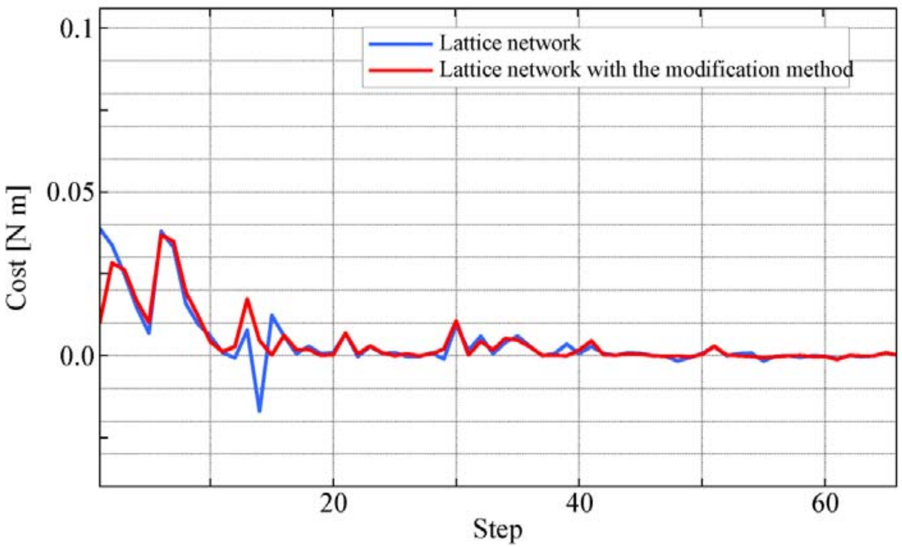


The motion of Planning 1 applied the modification

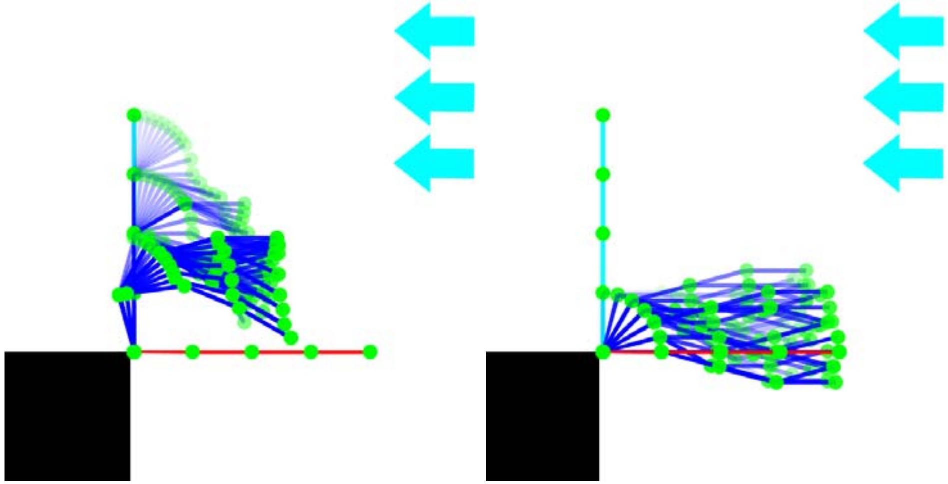
Results of Planning 1 (Energy, Lattice)

Method

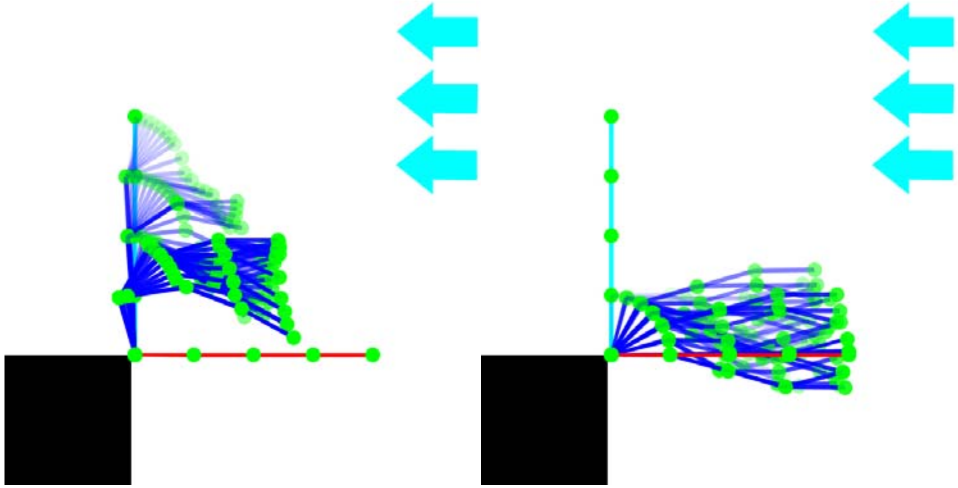
- Minimization:
  The energy consumption
- Discretization:
  The lattice network



Time series graph for the cost

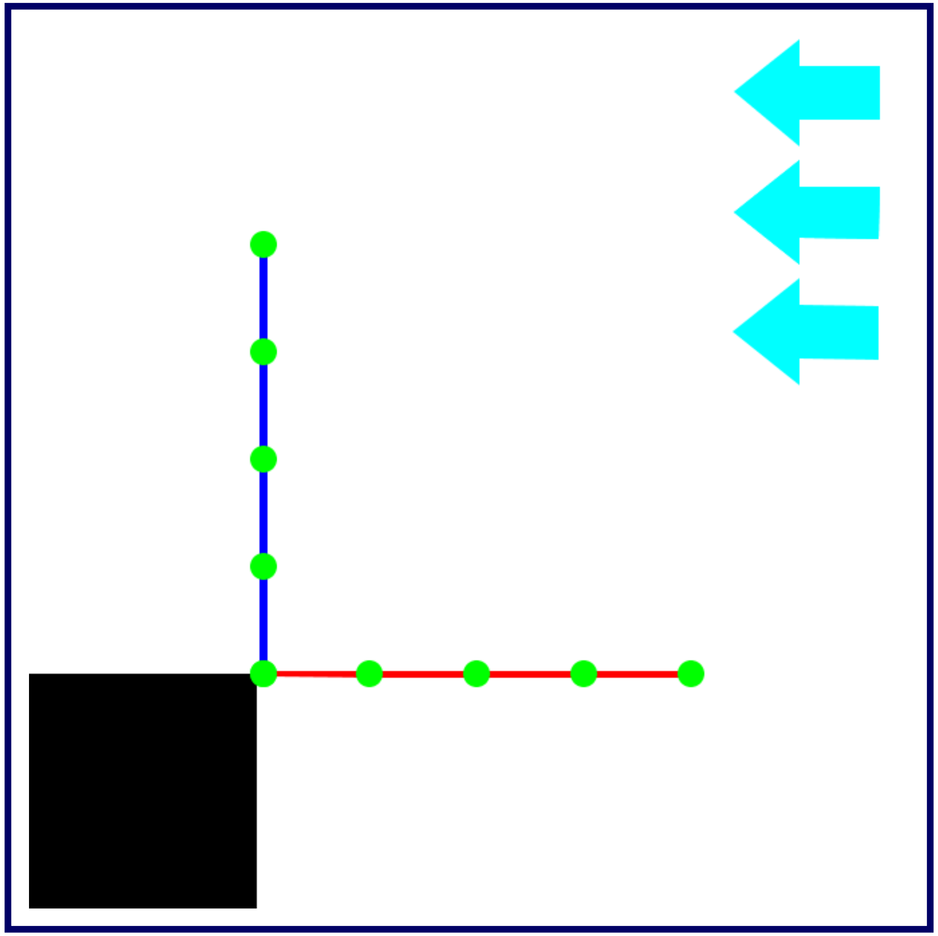


Trajectory of Planning 1

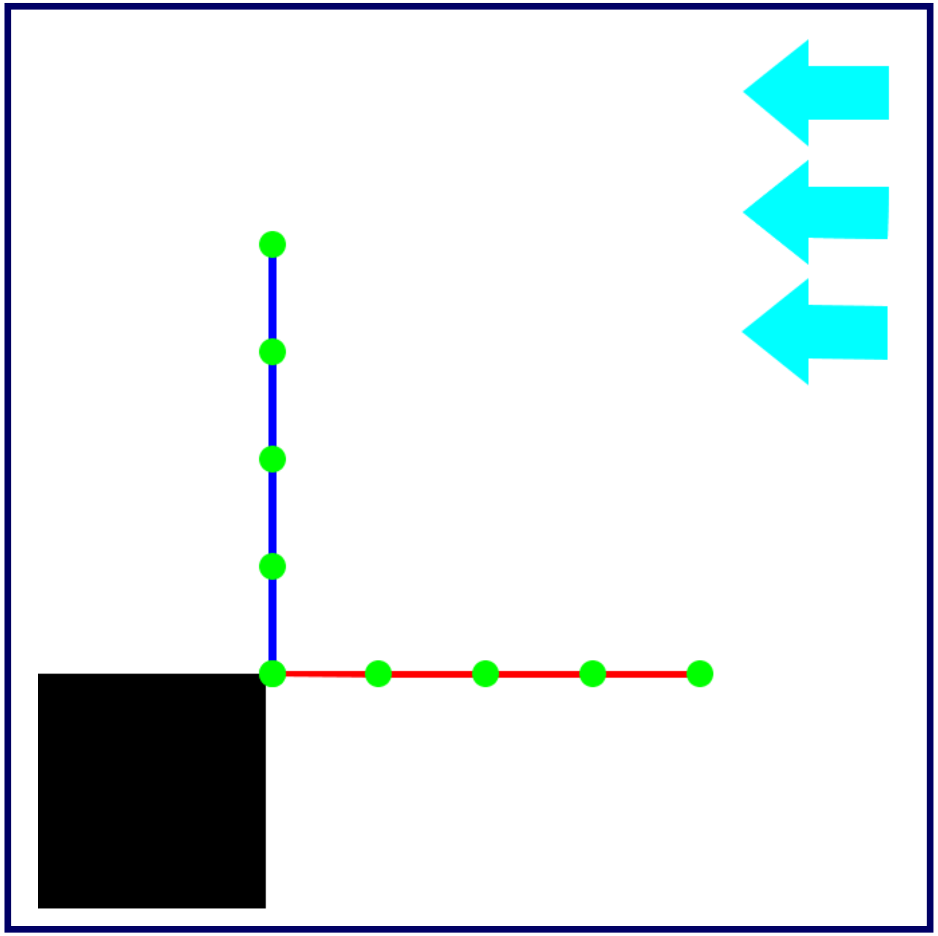


Trajectory of Planning 1 applied the modification

Results of Planning 2 (Energy, Random)



The motion of Planning 2

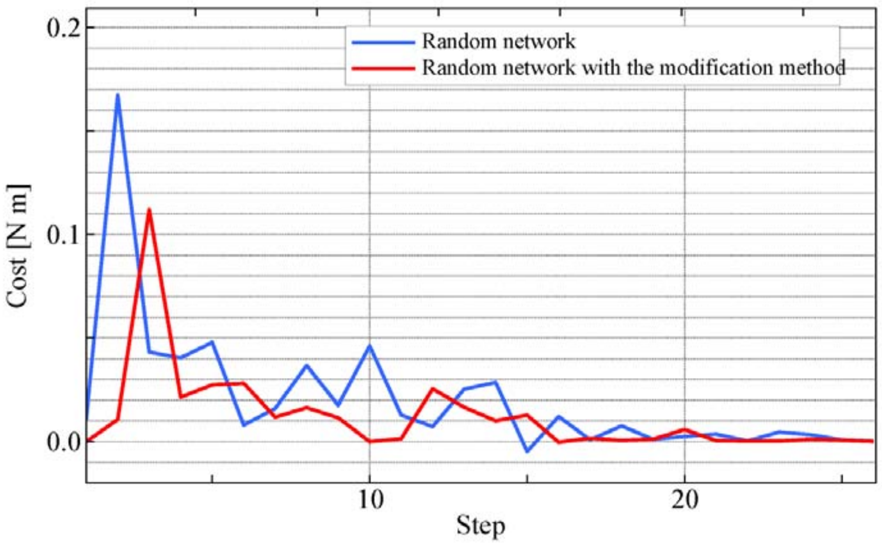


The motion of Planning 2
applied the modification

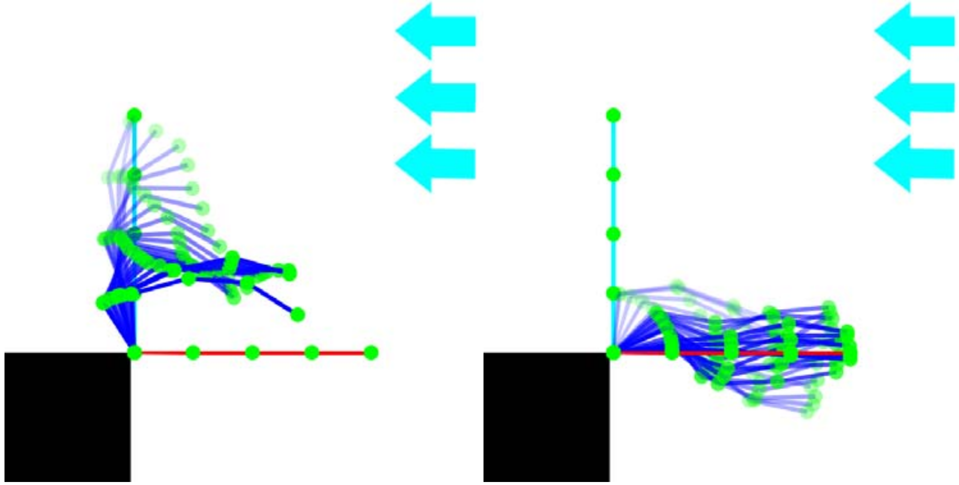
Results of Planning 2 (Energy, Random)

Method

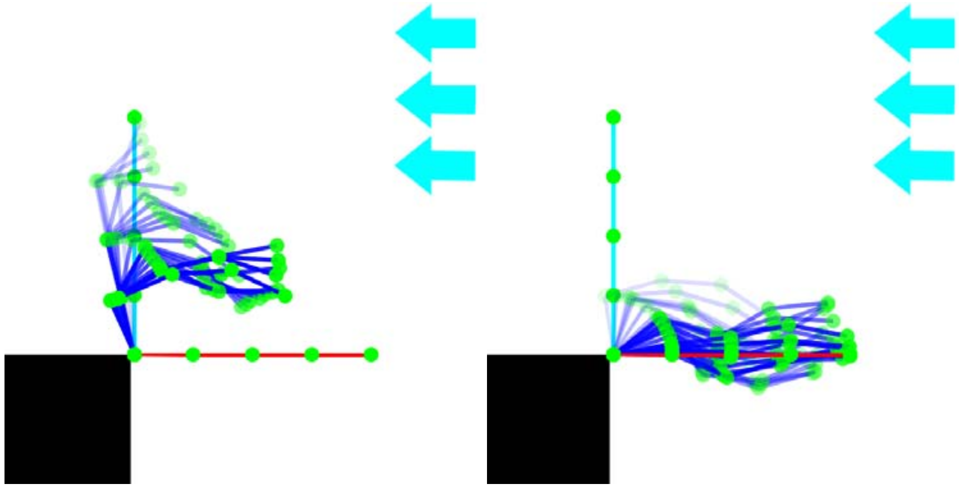
- Minimization:
  The energy consumption
- Discretization:
  The random network



Time series graph for the cost



Trajectory of Planning 2



Trajectory of Planning 2 applied the modification

Results of Experiment 1

Results of calculation for the motion planning

Planning (Cost, Network)	Cost [N·m]	Calculation time [s]
Planning 0 (Angle, —)	1.906±0.0	—
Planning 1 (Energy, Lattice)	0.311±0.0	744.6±0.0
Planning 2 (Energy, Random)	0.536±0.031	150.7±14.1

In Planning 0, the cost was recalculated using the obtained motion.

Results of calculation for the modification

	Cost [N·m]	Calculation time [s]
Modified Planning 0	0.514±0.0	25.9±0.0
Modified Planning 1	0.291±0.0	1532.5±0.0
Modified Planning 2	0.338±0.018	274.2±76.4

We found the motion which have little consumed energy when the manipulator moved against the uniform flow.

Experiment 2:Thrust Force Generating Task

Setting for Experiment 2

- A 3-DOF manipulator was used.
- Parameters of the manipulator were as follows.

$l_i = 0.7[\mathrm{m}], D_i = 0.2[\mathrm{m}], C_d = 1.17,\ \theta_{limit} = \pm 1.047[\mathrm{rad}]$

$\dot{\theta} = 0.034[\mathrm{rad/s}]$ ($\dot{\theta}$:is the velocity of the links)

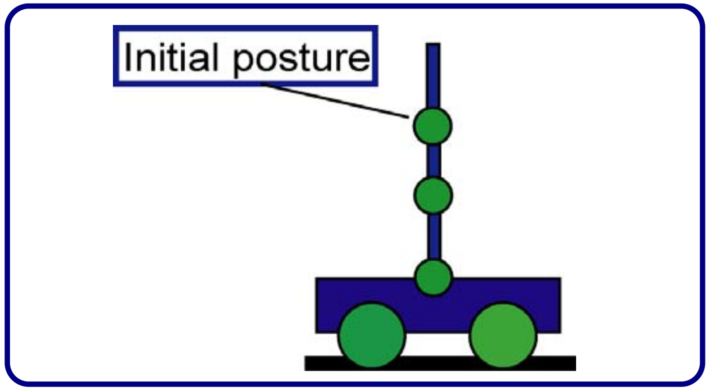
- Parameters of the robot were as follows.

The height and width of the body were 0.2 [m] and 0.3 [m],respectively.

- Parameters of the underwater environment was $\rho = 1.023$.

Calculation Environment

CPU : Core2Duo E4500 2.2GHz
Memory : 2GB
OS : Windows XP
Program language : Java



Experiment 2:Generating Thrust Force Task

Discretization of State Space

	The number of nodes	The number of links per a node
Lattice network	1,331(=11^3)	26
Random network	600	15

Motion planning sets

To enable comparison of the results,
two motion planning sets were used

	Reward function	Network
Planning 3	Advanced distance	Lattice network
Planning 4	Advanced distance	Random network

Simulation was executed 10 times and the performance was evaluated from an average of the results

Results of Planning 3 (Distance,Lattice)

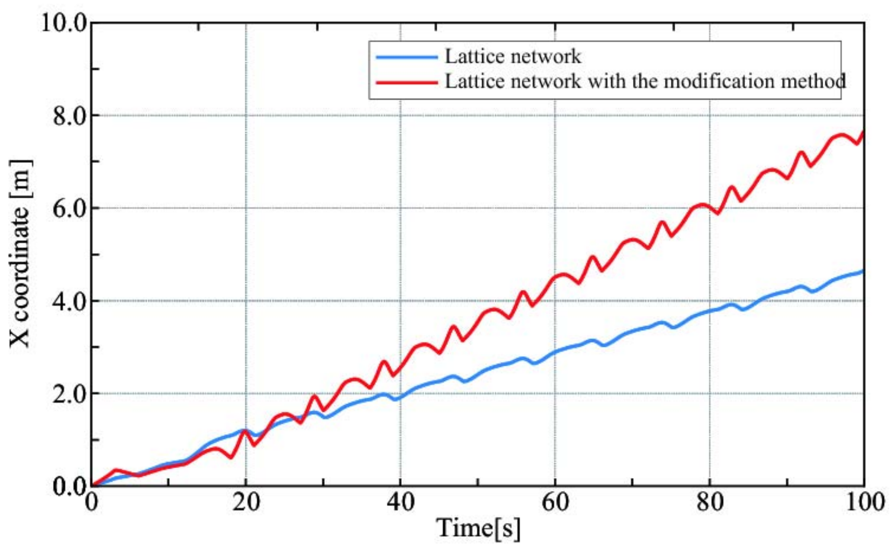
The motion of Planning 3

The motion of Planning 3 applied the modification

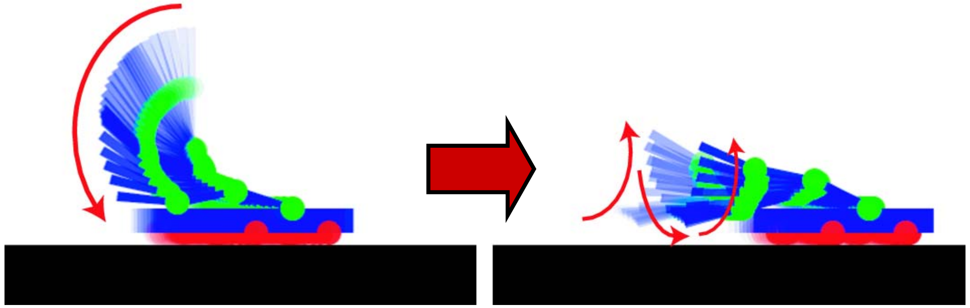
Results of Planning 3 (Distance,Lattice)

Method

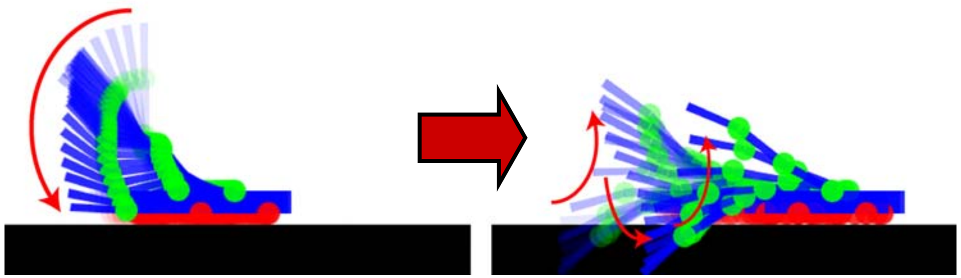
- Maximization:
  The advanced distance
- Discretization:
  The lattice network



Time series graph for the x-coordinate values



Trajectory of Planning 3



Trajectory of Planning 3 applied the modification

Results of Planning 4 (Distance,Lattice)

The motion of Planning 4

The motion of Planning 4 applied the modification

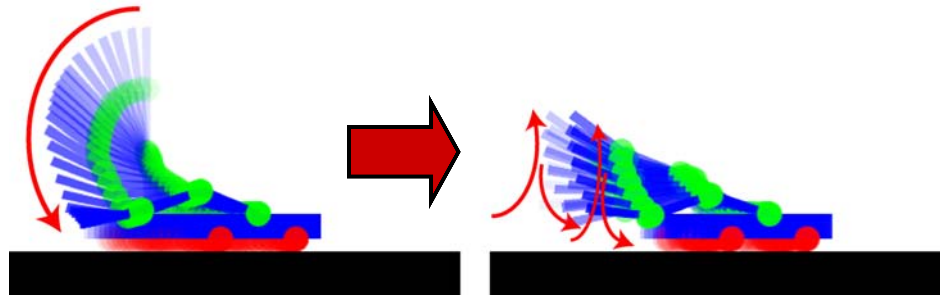
Results of Planning 4 (Distance,Random)

Method

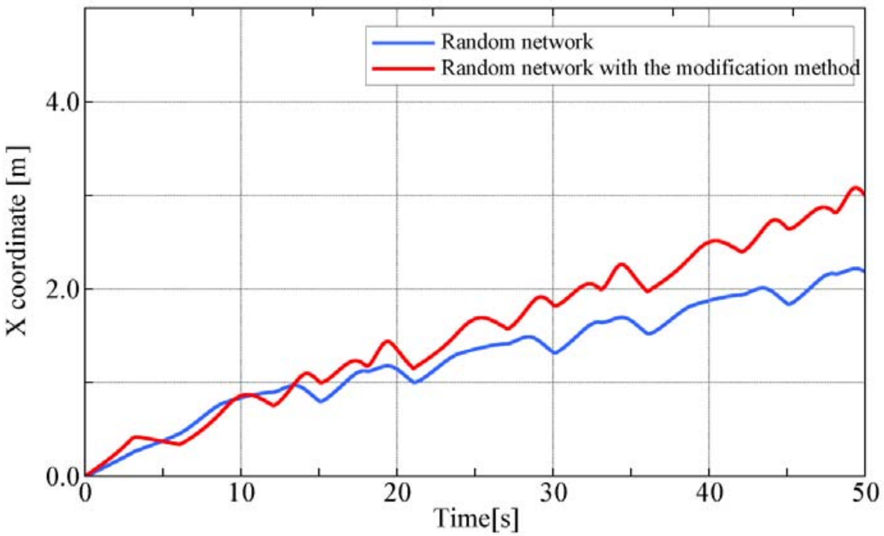
- Maximization:

  The advanced distance
- Discretization:

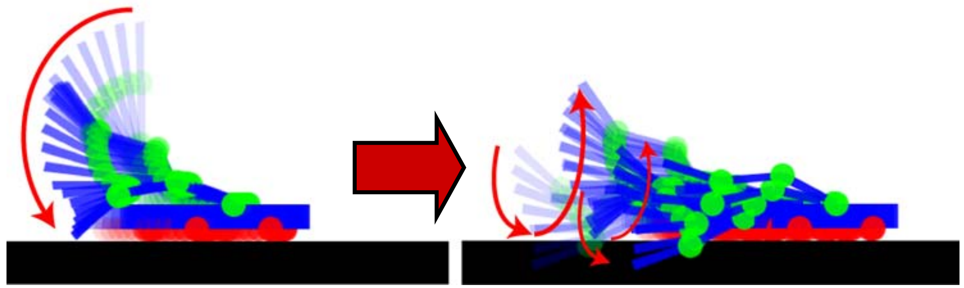
  The random network



Trajectory of Planning 4



Time series graph for the x-coordinate values



Trajectory of Planning 4 applied the modification

Results of Experiment 2

Results of calculation for the motion planning

Name of Planning (Reward, Network)	Velocity [m / s]	Calculation time [s]
Planning 3 (Distance, Lattice)	0.0465±0.0	509.7±0.0
Planning 4 (Distance, Random)	0.0417±0.0065	247.8±3.45

Results of calculation for the modification

	Velocity [m / s]	Calculation time [s]
Modified Planning 3	0.0724±0.0	5.74±0.0
Modified Planning 4	0.0571±0.0153	4.16±2.291

We obtained the repetitive motion like the flutter kick for swimming

Outline of Presentation

1. Introduction
   Background and objectives
2. Formulation
   Motion planning problems
   for multifunctional underwater robots
3. Approach
   Unified motion planning method
   using random network and gradient method
4. Experiments
   Reaching task and thrust force generation task
5. Conclusions

Conclusions

- The unified motion planning method using Dynamic programming was proposed.
- The proposed method can generate motion planning for very different tasks using the same algorithm.
- To avoid the curse of dimensionality, we applied a random network as a state transition network.
- We developed the modification method using the gradient method to improve the depleted solution.
- The effectiveness of the proposed method was verified through some numerical experiments.

Thank you very much for your attention !!