The Fourteenth International Symposium on Artificial Life and Robotics 2009 (AROB 14th '09),
B-Con Plaza, Beppu, Oita, Japan, February 5 - 7, 2009

A unified motion planning method for a multifunctional underwater robot

Koichiro Shiraishi and Hajime Kimura
Dept. of Maritime Engineering
Graduate School of Engineering, Kyushu University
744 Motooka, Nishi-ku,Fukuoka, 819-0395 Japan
k-shiraishi@system.nams.kyushu-u.ac.jp and kimura@nams.kyushu-u.ac.jp

Abstract

This paper deals with motion planning for a multifunctional underwater robot which can accomplish various missions, e.g., swimming, walking and grasping objects. Authors have developed a unified motion planning method which can generate motion planning for a variety of task by a single algorithm. In this method, motion planning problems are modeled as finite horizon Markov decision processes. The optimum motion planning is obtained by Dynamic Programming, however Dynamic Programming is sometimes thought to be of limited applicability because of the curse of dimensionality. To avoid the curse of dimensionality, authors applied a random network as a state transition network. The explosion of the number of states can be suppressed by using the random network. The effectiveness of the proposed method is demonstrated through numerical simulations of two types of tasks for multifunctional robots. One is a reaching task, the other is a generating thrust force task.

Keywords: Mutifunctional underwater robot, Fluid Drag Force, Reaching task, Generating thrust force task, Dynamic Programming, Random network

1 Introduction

The field of underwater robotics is currently enjoying a period of high interest and growth. Applications for such robots include exploration of deep sea environment, cable maintenance, monitoring of subsurface structures, and biological surveys [1]. Authors have been developing a multifunctional underwater robot which can accomplish various missions, e.g., swimming, walking and grasping objects [2]. The robot is equipped with a redundant degree of freedom manipulator. It has two excellent features; One is robustness for troubles, the other is multifunction, that is, it can use not only as an arm, but also as a finger for gripping objects, or a fin for swimming.

It is necessary for a robot to generate motion planning according to various missions. The problem is that we must develop the corresponding motion planning algorithms for each individual task. Authors have developed a unified motion planning method that can generate motion planning for a variety of task by a single algorithm. In the method, motion planning problems are modeled as finite horizon Markov decision processes. The optimum motion planning can be acquired by using Dynamic Programming.

This paper presents a new approach which has ability to avoid curse of dimensionality. The number of state increase exponentially with the number of state variables is called the curse of dimensionality. The conventional Dynamic Programming is to discretize state space with full grid points. Due to the exponential growth in the full grid discretization as the number of state variables increases, Dynamic Programming is still commonly considered to be computationally intractable. The proposed method generates a random state transition network by discretizing the state space at random. The random network can prevent from exponentially increasing of the number of states. The optimum motion planning can be acquired by using the Dynamic Programming in the random network.

The effectiveness of the proposed method is demonstrated through numerical simulations of two types of tasks for multifunctional robots. One is a reaching task that is composed of a manipulator posture planning, so that the robot keeps to a minimum of energy consumption [3]. The other is a generating thrust force task that is driving the robot forward by a fluid drag force that acts on a manipulator. The experimental results show that the proposed method could generate reasonable motion planning at two problems of motion planning by a single algorithm. The proposed method used a random network could apply to a high dimension problem, and enabled the computation time to be shortened.

2 Motion Planning Problem

In this paper, authors dealt with two motion planning problems of a multifunctional underwater robot. The one is a reaching task which is a path planning task for a robotic manipulator. The other is a generating thrust force task for a multifunctional underwater robot. The reaching task is to obtain an optimum sequence of postures in fluid. In the reaching task, the optimum planning minimizes energy consumption caused by the fluid drag force. Authors consider a movable underwater robot equipped with a manipulator in the generating thrust force task. This robot generates fluid drag force to move in front by moving the manipulator. In this task, the objective function is to maximize the advancing distance.

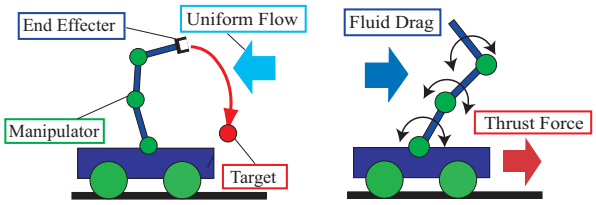

Figure 1: Reaching task and Generating thrust force task

2.1 Coordinate System of a Manipulator

Figure 2 shows the coordinate system and parameters in the manipulator. x and y are horizontal, vertical directions, respectively. l_i is the length of the ith link θ_i is the angle of the ith link. The shape of the links are assumed to be columns. Joints are assumed to be small enough compared with the links, so that fluid drag forces which act on joints are able to be omitted in the calculation. $\boldsymbol{S}$ is defined as a manipulator posture, which is vector composed of angles of the links, given by

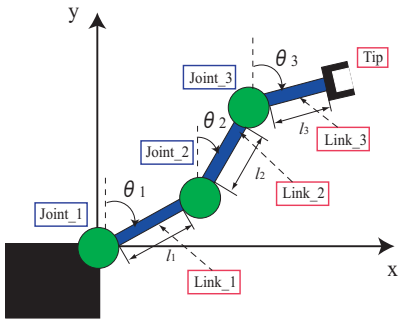

Figure 2: Configuration of three-link manipulator

$$\boldsymbol{S} = (\theta_1, \theta_2, \cdots, \theta_i, \cdots, \theta_n). \tag{1}$$

A point of the nth link's tip becomes the position of the end effecter. The point is given by

$$x_{Tip} = \sum_{i=1}^{n} l_i \sin\theta_i, \quad y_{Tip} = \sum_{i=1}^{n} l_i \cos\theta_i. \tag{2}$$

Coordinates of other links can be provided by using the equation (2). The notation $\boldsymbol{P}$ is a sequence of manipulator postures become target posture from initial posture, It is given by

$$\boldsymbol{P} = (\boldsymbol{S}_{init}, \boldsymbol{S}_2, \cdots, \boldsymbol{S}_j, \cdots, \boldsymbol{S}_{end}), \tag{3}$$

where $\boldsymbol{S}_{init}$ is the initial posture where the manipulator begins to move is an initial posture, $\boldsymbol{S}_{end}$ is the target posture which the end effecter reaches target coordinates, j is the number of posture changes, and $\boldsymbol{S}_j$ is jth manipulator posture.

2.2 The Fluid Drag Force

The Fluid drag force is given by

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

where D is a diameter of the column, C_d is a drag coefficient, C_m is an added mass coefficient, u is a velocity of a link through the fluid. ρ is a density of the fluid. Since the velocity of the link is small enough, the second term at the right of the equation (4) can be omitted. The fluid drag force acting on each link is shown as follows by the use of the equation (4)

$$F_{x,i} = F_{x,i+1} + f_i \cos\theta_i \tag{5}$$

$$F_{y,i} = F_{y,i+1} - f_i \sin\theta_i \tag{6}$$

$$M_i = M_{i+1} + T_i + F_{x,i+1} l_i \cos\theta_i - F_{y,i+1} l_i \sin\theta_i, \tag{7}$$

where f_i and T_i are given by

$$f_i = \int_0^{l_i} C_d \frac{1}{2}\rho D_i u_i |u_i| dr_i,$$
$$T_i = \int_0^{l_i} C_d \frac{1}{2}\rho D_i u_i |u_i| r_i dr_i.$$

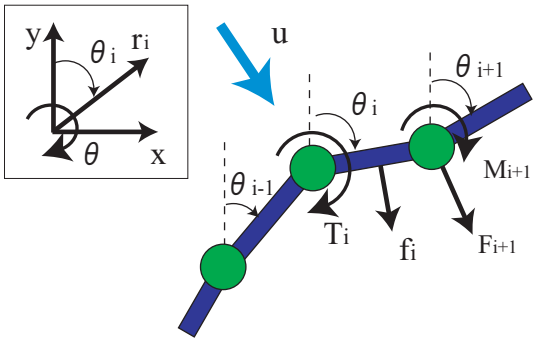

Figure 3: Forces acting on a link

2.3 A Reaching Task

Multifunctional underwater robot is operated under the subsea environment. It is necessary to consider an influence of fluid drag force caused by sea current. Thus, authors consider that a motion planning problem which minimizes energy consumption caused by the fluid drag force. The energy is determined by torque which acts on the joints and the angles of the links moved in a period. In the motion planning problem, the cost function is formulated by using the consumed energy. When a manipulator posture changes from $\boldsymbol{S}_A$ to $\boldsymbol{S}_B$, the cost function is shown approximately in the following,

$$Cost(\boldsymbol{S}_a, \boldsymbol{S}_b) = \sum_{i=1}^{n} \int_{\theta_{S_{a,i}}}^{\theta_{S_{b,i}}} M_i d\theta_i + \Delta E, \qquad (8)$$

where n is the number of the links, θ_{S_i} is the angle of the ith link in the manipulator posture S, ΔE denotes energy consumption caused by a mechanical friction. A total cost is provided by the use of the equation (8) as

$$Total\ Cost(\boldsymbol{P}) = \sum_{i=1}^{|\boldsymbol{P}|-1} Cost(\boldsymbol{S}_i, \boldsymbol{S}_{i+1}). \qquad (9)$$

In the reaching task, the motion planning problem is formulated as an optimization problem of minimizing a total cost of the equation (9).

$$Optimum\ Planning = \min_{\boldsymbol{P} \in \boldsymbol{P}_{All}} Total\ Cost(\boldsymbol{P}) \qquad (10)$$

where $\boldsymbol{P}_{All}$ denotes sets of all executable motion planning in the motion planning problem. In this paper, the positive energy obtained from the current is not considered.

2.4 A Generating Thrust Force Task

In this section, authors describe the motion planning problem of generating thrust force for a movable underwater robot. The robot is able to move in front by using the fluid drag force that acts on the manipulator. In this paper, it is assumed that the body of the robot is constrained by the rail. For a preliminary experiment, the one dimentional motion is considered. For simplicity, the effect of the added mass which is concerned with the fluid is assumed to be negligible. Consider that fluid drag forces that acts on main body and each links. The motion equation of the body is given by

$$M\ddot{x} = F_{body} + \sum_{i=1}^{n} F_{x,i}, \qquad (11)$$

where M is a mass of the movable underwater robot. F_{body} denotes the fluid drag force that acts on the body of the robot except for the force on the links. $F_{x,i}$ denotes the fluid drag force of x axial component that acts on the ith link. n is a number of links of the manipulator. In this problem, the optimum planning is a motion to maximize advancement distance of the robot.

3 A Unified Motion Planning Method

The motion planning problem of the multifunctional underwater robots is modeled as finite horizon Markov decision processes. Optimum motion planning for various tasks is obtained by using Dynamic Programming.

3.1 Markov Decision Processes

State transitions of the robot is represented as a network showing Fig.4.

In the network, a node denotes a manipulator posture. A direction of an arrow denotes a direction of the posture change. $f(s_i, s_{i+1})$ is an objective function, when the robot posture changes from s_i to s_{i+1}. In reaching task, an objective function is consumed energy caused by fluid drag force. On the other hand, an objective function is an advancement distance in generating thrust force task. Thus, the motion planning problem is considered as a graph search problem.

Assuming that Markov property is approved, the motion planning problem is considered as a Markov decision process. Markov Decision Processes framework for planning is rich in capturing the essence of purposeful activity in various situations. A Markov decision process is defined by its state and action sets

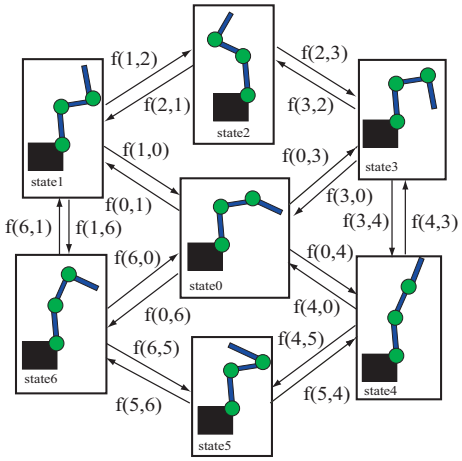

Figure 4: State transition network

and by the one-step dynamics of the environment [4] [5]. A state s is defined as a manipulator posture. An action a is motion of each link in the posture change. A reward $R(s'|s,a)$ is a value of the objective function in the state transition. A value function $V(s)$ denotes the expected total reward starting at state s according to the strategy. In Markov decision process, the Bellman equation is shown by

$$V^*(s) = \max_{a \in A} \sum_{s' \in S} p(s'|s,a) \left(R(s'|s,a) + \gamma V^*(s') \right), \quad (12)$$

where $\boldsymbol{S}$ is state sets, $\boldsymbol{A}$ is action sets, and γ is a discount factor. The optimum policy is obtained by solving the equation (12) by using the standard Dynamic Programming. The policy is the optimum motion planning for various tasks In this paper, a standard value iteration method is used for solving the Bellman equation.

3.2 Discretization of State Space

In this section, authors describe a random network used as a state transition network. State space should be discretized in standard Dynamic Programming. In conventional method, the state space is discretized like a lattice. However, this method is suffered from the curse of dimensionality. The curse of curse of dimensionality is a phenomenon that a number of states increases in exponential. A random network which discretizes the state space at random, is developed as a method to avoid the curse of dimensionality [6] [7]. In this paper, the state transition network is produced by the random network. The optimum motion planning can be obtained by using Dynamic Programming with this network.

4 Experiment and Results

In order to confirm the validity of proposed method, several experiments of two tasks for the multifunctional underwater robots were made. One is the reaching task, the other is the generating thrust force task. Figure 5 shows settings of the multifunctional underwater robot in two tasks.

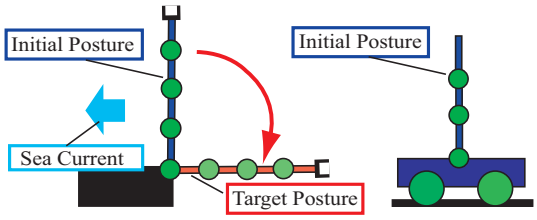

Figure 5: Settings of two Experiments

4.1 Experiment 1: Reaching Task

The aim of Experiment 1 is to verify whether the proposed method could obtain the motion planning minimizing the consumed energy by fluid drag. In the experiment, the 4 D.O.F. manipulator was used. The parameter of the experiment is set as follows: $l_i = 0.7[m], D_i = 0.2[m], C_d = 1.17, \rho = 1.023, |u| = 2.0[m/s]$, and $\theta_{limit} = \pm 1.571[rad]$, where θ_{limit} is the angle where the links of the manipulator can rotate.

In order to compare results, three motion planning were used: Planning 1 is to reach a target posture in the shortest time, Planning 2 is generated by the unified motion planning with the lattice network, and Planning 3 is generated by the unified motion planning with the random network. The lattice network was provided by discretization of the state space like a lattice. Each dimension of the state space is divided evenly into 11 parts. The number of nodes is $11^4 = 14641$, the number of links per a node is 80 in the lattice network. On the other hand, the random network was provided by discretization of the state space at random. The number of nodes is 8000, the number of links per a node is 50 in the random network. The simulation is executed 10 times and the performances are evaluated by the average of the results.

Figure 6, 7, and 8 show the motion planning obtained by simulation of the reaching task. All planning

could obtain the motion to reach from the initial posture to the target posture. Table 1 show the cost and the computation time on experiment 1. Planning 2 has the smallest cost among the three motions, although the computation time is longest. The computation time of Planning 3 is a half compared with Planning 2. Planning 3 is the most effective solution, considering two points of the computation time and the cost. Authors could notice that the motion of the best solution shown in Figure 7 or Figure 8 takes advantage of the sea current.

Table 1: Simulation results of experiment 1

	Cost[J]	Computation Time[s]
Planning 1	0.93044 ± 0	178.0 ± 0
Planning 2	0.00073 ± 0	661.3 ± 0
Planning 3	0.01326 ± 0.0088	344.5 ± 20.5

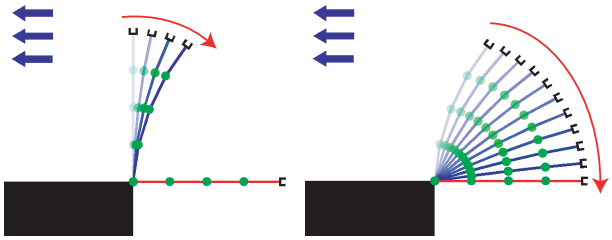

Figure 6: Motion to reach in the shortest time (Planning 1)

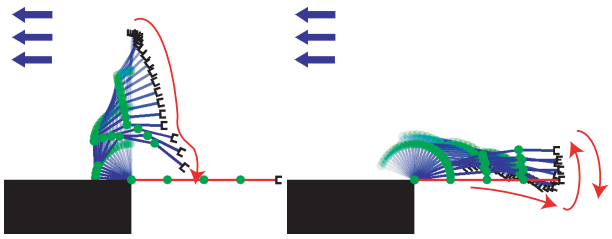

Figure 7: Motion obtained by proposed method with lattice network (Planning 2)

4.2 Experiment 2: Generating Thrust Force Task

The aim of the experiment is to verify whether the proposed method could obtain the generating thrust force motion. In the experiment, the movable underwater robot equipped with the 3 D.O.F manipulator is used. The parameter of the experiment is set as follows : $l_i = 0.5[m], D_i = 0.2[m], C_d = 1.17, M = 5.0[kg], \rho = 1.023, \theta_{limit} = \pm 1.047[rad]$, and $\dot{\theta} = 0.034$ $[rad/s]$, where $\dot{\theta}$ is the velocity of the links of the manipulator. The height and the width of the body is $0.2[m]$ and $0.3[m]$. In the lattice network, the number of nodes is $11^3 = 1331$, the number of links per a node is 26. In the random network, the number of nodes is 600, the number of links per a node is 50. In order to compare results, two motion planning were used : Planning 4 is the motion planning generated by the unified motion planning with a lattice network and Planning 5 is the motion planning generated by the unified motion planning with a random network.

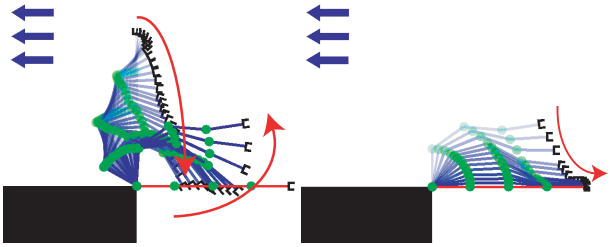

Figure 8: Motion obtained by proposed method with random network (Planning 3)

Figure 10 and 11 show the motion planning obtained by simulation of the generating thrust force task. In the both motion the motion of the manipulator is a flutter kick for swimming. Table 2 show the average velocity and the computation time on experiment 2. Planning 4 is superior to Planning 5 in the average velocity. The computation time of Planning 5 is a half compared with Planning 4. Planning 5 is the most effective solution, considering two points of the average velocity and the computation time.

Table 2: Simulation results of experiment 2

	Average velocity [m/s]	Computation Time [s]
Planning 4	0.0271 ± 0	510 ± 0
Planning 5	0.0235 ± 0.0052	248 ± 34.5

5 Summary and Conclusions

In this paper, authors developed the unified motion planning for multifunctional underwater robots. The motion planning problems are modeled as Markov decision processes and optimum motion planning is ob-

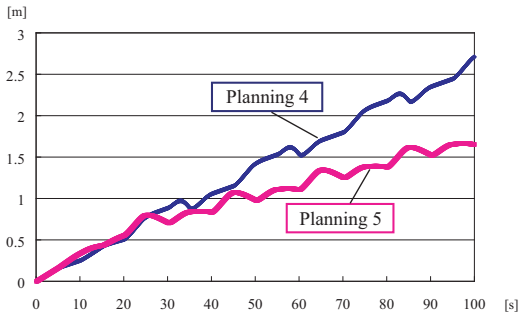

Figure 9: Time series graph for x coordinate value of a movable underwater robot

Figure 10: Motion obtained by proposed method with lattice network (Planning 4)

Figure 11: Motion obtained by proposed method with random network (Planning 5)

tained by standard Dynamic Programming. The proposed method can generate motion planning for a variety of tasks by a single algorithm. In addition, the proposed method used random networks enabled the computation time to be shortened. The method could avoid the curse of dimensionality, since the number of nodes is independent on the number of state variables. The validity of the method was confirmed through two experiments.

The drawback of the random network approach is that the quality of the best solution has large variance because of the random location of the state node. That is, the solution is depending on the initial design of the random network. It is necessary to develop the method to suppress the variance.

In the future work, authors will utilize the conjugate gradient method to solve the problem that the random networks approach has the variance of the solution. The method is an algorithm for finding the nearest local minimum of a function of variables which presupposes that the gradient of the function can be computed. The locations of the nodes in the solution are modified by the conjugate gradient method, so that the quality of the solution is improved. Therefore, this approach could reduce the variance of the solution in the random network.

Acknowledgements

This work partly was supported by the Sasakawa Scientific Research Grant from The Japan Science Society.

References

[1] G. Antonelli, *Underwater Robots: Motion and Force Control of Vehicle-Manipulator Systems*, Springer, 2003.

[2] K. Shiraishi and H. Kimura, "On a Unified Motion Planning for a Multifunctional Underwater Robot," *Proceedings of the 26th Annual Conference of the Robotics Society of Japan*, (in Japanese) 2008.

[3] S. Ishibashi, M. Ito, and E. Shimizu, "Autonomous Motion Planning for a Manipulator Equipped on AUV in a Workspace Divided into Cubes," *Proceedings of the 13th International Offshore and Polar Engineering Conference*, pp. 231-238, 2003.

[4] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*, Athena Scientific, 1996.

[5] Abhijit Gosavi, *Simulation-Based Optimization*, Kluwer Academic Publishers, 2003.

[6] C. G. Atkeson and B. J. Stephens, "Random Sampling of States in Dynamic Programming," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 38, pp. 924-929, 2008.

[7] S. M. LaValle, M. S. Branicky, and S. R. Lindemann, "On the Relationship Between Classical Grid Search and Probabilistic Roadmaps," *International Journal of Robotics Research*, Vol. 23, pp. 673-692, 2004.