Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Coordinated Box-pushing of Multiple Biomimetic Robotic Fish

Dandan Zhang, Yimin Fang, Long Wang, and Guangming Xie

***Abstract*— This paper proposes a coordination method for multiple biomimetic robotic fish in box-pushing task. Having successfully developed a robotic fish prototype, and built a platform of Multiple Robotic Fish Coordinated Control System to study coordinated control strategies for multiple robotic fish, we step further to study coordination problems of multiple robotic fish in unstructured and dynamic environments. To simplify the difficulty of the path planning and action decision, we employ a *situated-behavior* design method which is based on situations, and for each situation a specific behavior is elaborately designed. On dealing with the synchronization and coordinated pushing problem, fuzzy logic method is adopted for motion planning of the fish. Experimental results of box-pushing performed by two robotic fish validate the effectiveness of the proposed method.**

I. INTRODUCTION

As is known, robotics inspired by biomimetics is being developed in recent years to investigate biological systems. In particular, research on robotic fish has become a hot topic and received much attention[1]-[7]. It is well-known that a fish in nature propels itself by coordinated motion of the body, fins, and tail, achieving tremendous propulsive efficiency and excellent maneuverability. From the perspective of engineering science, fish is a prototype of a distinguished autonomous underwater vehicle (AUV). Taking advantage of progress in robotics, control technology, artificial intelligence, hydrodynamics of fish-like swimming, new materials, fabrication technologies, sensors, and actuators, emerging research has been focused on developing novel fish-like vehicles, to imitate the locomotion mechanism of fish in nature to get favorable efficiency, maneuverability and low noise performance.

We define a robotic fish as a fish-like aquatic vehicle driven by the undulatory motion, with anatomic structure of a real fish: primarily the undulatory body, the highly controllable fins and the large aspect ratio lunate tail[1]. Since most of the previous research on robotic fish was mainly concerned with the hydrodynamic mechanism of fishlike swimming, coordinated control of multiple robotic fish is definitely an important research topic for future engineering applications. However, few research results are available in the literature. Although fruitful cooperation methods and valuable experimental results for ground robots have appeared, it's very difficult to apply these methods directly to robotic fish for the complexity and particularity of the underwater working environment and the particular movement modes of robotic fish (see section III-B for the difficulties in robotic fish control). Coordinated control of multiple robotic fish remains certainly a challenging research topic.

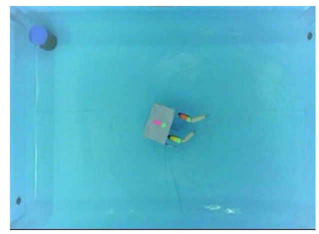


Fig. 1. Environment of The Box-pushing Task.

Previously we have developed several radio-controlled, multi-linked robotic fish of different forms and functionalities, each propelled by the flexible posterior body and the oscillatory tail[3]-[5]. In this paper, we study multiple robotic fish coordination problem in context of the box-pushing task, since box-pushing has long been one of the canonical task domains for cooperative robotics[8]. In the box-pushing task, several robotic fish are required to cooperatively move a rectangular box, which is large relative to the size of the fish, from an initial location to a designated goal location in the water environment (depicted in Fig. 1). Because of the complex hydrodynamics of the fluid environment and dynamics of the fish when it swims, it is difficult to establish precise mathematical model by purely analytical methods[1], and we can only predict approximately the response of the fish on the control commands. Under these limitations, to simplify the path planning and action decision for the individual fish when it is in different positions relative to the box, we utilize the *situated-behavior* method to divide the environment into a set of complete and exclusive situations, and for each situation, a specific behavior is elaborately designed, which may include several actions. To implement the method, a geometry-based approach called Comfortable Circle Approach (CCA) is proposed. In real time, sensory information from an overhead camera is used to identify one situation, then the fish can perform the behavior associated with the identified situation by executing relative actions. Since the robotic fish can not stop immediately on a "stop" command, fuzzy logic method is employed to facilitate the synchronous arrival of the two fish at the Attacking Points. The orientation of the box is controlled by the coordinated motion of the fish, obtained through a set of fuzzy logic rules. Experiments on two robotic fish validate the effectiveness of our proposed method.

This work is supported by the National Natural Science Foundation of China #10372002, #60274001 and the National Key Basic Research and Development Program #2002CB312200

The authors are with Intelligent Control Laboratory, Center for Systems and Control, Department of Mechanics and Engineering Science, Peking University, Beijing 100871, P. R. China twinkleice@pku.edu.cn

The paper is organized as follows. In section II, a review of the robotic fish design and Multiple Robotic Fish Coordinated Control System (MRFCS) is presented. In section III, we describe our box-pushing task and the coordination method in detail. Experimental results are given in section IV. Section V concludes the paper.

II. ROBOTIC FISH PROTOTYPE AND MRFCS

A. Simplified Propulsive Model of The Robotic Fish

Body or caudal fin (BCF) swimming movements of natural fish are usually categorized into anguilliform, carangiform, subcarangiform, and thunniform mode primarily according to the wavelength and the amplitude envelope of the propulsive wave underlying fish′s behavior[1][9]. Our designed robotic fish takes carangiform movement. Barrett et al. has presented a relative swimming model for RoboTuna (carangiform) in [10], and the undulatory motion is assumed to take the form of a propulsive travelling wave which is described by:

$$y_{body}(x,t) = [(c_1 x + c_2 x^2)][\sin(kx + \omega t)]. \qquad (1)$$

In (1), y_{body} is the transverse displacement of the fish body, x the displacement along the main axis, k the body wave number ($k = 2\pi/\lambda$), λ the body wave length, c_1 the linear wave amplitude envelope, c_2 the quadratic wave amplitude envelope, and ω the body wave frequency (ω=$2\pi f$).

For simplification, we consider the discrete form of travelling wave (1), which is described by:

$$y_{body}(x,i) = [(c_1 x + c_2 x^2)][\sin(kx \pm \frac{2\pi}{M} i)], \qquad (2)$$

where i denotes the index of the sequences, M is the body wave resolution, representing the discrete degree of the overall travelling wave.

The designed oscillatory part of a robotic fish consists of several rotating hinge joints, as shown in Fig. 2. It is modelled as a serial of links along the main body axis, and the positions of the links in the chain is achieved by numerical fitting. See [2] for details of determination and optimization of the ratio $l_1 : l_2 : ... : l_n$.

B. Multiple Robotic Fish Coordinated Control System (MRFCS)

We have built a platform of MRFCS for developing and evaluating coordination mechanisms for the robotic fish. As depicted in Fig. 3, MRFCS is a system consisting of three hierarchies: the decision-making hierarchy, including a host pc with a friendly man-machine interface, and several control algorithms embedded in; the information exchange hierarchy, in which the control commands are sent to fish through a radio transmitter, and also the environment information is fed back to the decision-making hierarchy by the overhead camera; the executive hierarchy, in which the robotic fish execute the commands from the decision-making hierarchy.

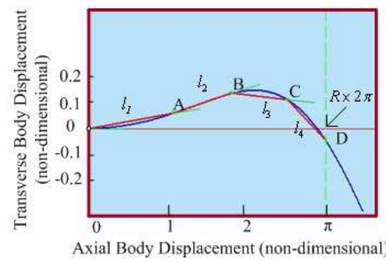


Fig. 2. Links Based Body-wave Fitting.

III. MULTIPLE ROBOTIC FISH BOX-PUSHING

A. Task Description

The box-pushing task is stated as follows: Two robotic fish are required to coordinately move a rectangular box 365 mm $\times$260 mm $\times$95 mm (length $\times$ width $\times$ height) from an initial location to a goal location in a swim tank with dimension of 3200 mm $\times$2200 mm $\times$700 mm (length $\times$ width $\times$ depth). The box has an orientation and the fish are only allowed to push the box behind the box with their heads, one on the left side, the other on the right side. The orientations of the box and the fish can be recognized by the overhead camera.

B. Difficulties in Multiple Robotic Fish Box-pushing

Difficulties of robotic fish control in the box-pushing task are briefly summarized as follows:

⋆ It is difficult for a robotic fish to trace a linear path. In addition, the fish can not move reversely like a wheeled ground robot.

⋆ The robotic fish can not stop immediately on receiving a ”stop” command because the resistance in water is much less than that on the ground. So the two fish shall synchronize the arrival at the Attacking Points of the box, but it is hard to control.

⋆ Waves occur when a robotic fish moves, which affect the movement of the robotic fish and others even in static state. This leads to great difficulty in precise point-to-point (PTP) control.

⋆ Since it is very difficult to establish the hydrodynamic model of fishlike swimming using analytical method, we can only predict approximately the response of the robotic fish on a control command.

C. Comfortable Circle Approach for Box-pushing

Considering the difficulties in precise locomotion control mentioned above, we employ a methodology similar to the *situated-activity* paradigm(see [11]) to reduce the difficulties of path planning and action casting in the box-pushing task. The *situated-activity* paradigm is a design methodology based on defining a set of situations which describe the relative states of the problem entities and applying the associated action to each situation. The design based on this paradigm shall satisfy the following requirements[12]:

⋆ The situations shall be identifiable, exclusive and complete to represent the relative states of the problem entities,

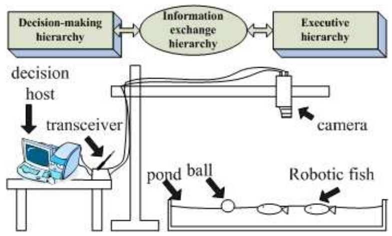


Fig. 3. Multiple Robotic Fish Coordinated Control System.

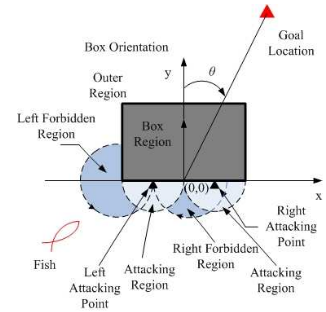


Fig. 4. Problem Entities in Box-pushing.

and in defining situations, an explosion in the number of the situations shall be avoided.

$\star$ Each action designed for the associated situation is required to solve the associated problem individually.

The advantages of employing the *situated-activity* paradigm is that the paradigm is a "divide and conquer" strategy, which reduces the task difficulty; in addition, the real-time *action coordination problem* need not to be taken into consideration, since the situations are complete and exclusive.

Our design methodology is similar to the *situated-activity* methodology except that a corresponding "behavior", instead of action, is designed for each situation. These behaviors can also include low-level behaviors. We call this methodology the *situated-behavior* methodology. This methodology reduces the number of the situations, and some basic low-level behaviors, for example, obstacle avoidance behavior, can be shared by the situation associated behaviors directly.

We propose a geometry-based implementation of the *situated-behavior* methodology called Comfortable Circle Approach (CCA). Next we will describe CCA in detail.

1) The Situations: Firstly we give some definitions.

Attacking Points: Points at which the fish push the box.

Attacking Regions: Semicircular regions centered at the Attacking Points with radius of $1/4$ length of the box. In these regions, the fish can push the box directly.

Comfortable Radius and *Comfortable Circle*: The Comfortable Radius is 1.2 length of the minimum turn radius of the fish, which means that the fish can turn comfortably with this radius. A directed circle with Comfortable Radius is called a *Comfortable Circle*.

Forbidden Regions: The boundaries of the Forbidden Regions are *Comfortable Circles*, passing the Attacking Points with directions the box orientation.

The situations are obtained according to the relative states of the problem entities, including the robotic fish, the box, the Attacking Points and the goal location. Shown in Fig. 4, the environment is divided into Box Region, Attacking Regions, Left Forbidden Region, Right Forbidden Region, and Outer Region. Here we discuss only the situations for the Left Attacking Point and the fish whose goal Attacking Point is the left one. The situations for the Right Attacking Point and the fish whose Attacking Point is the right one are similar. Our objective is to plan appropriate paths which lead the robotic fish to arrive at the Attacking Points with direction the box orientation, in order to prepare for the next step – box-pushing. Considering that the robotic fish can not stop immediately on a "stop" command when arriving at the Attacking Point, the motion planning shall lead the two fish to reach the Attacking Points synchronously. We use a decision tree to define the set of situations according to the relative states of the problem entities. As shown in Fig. 5, the inputs of the decision tree are the goal location information and sensory information from the overhead camera, including the locations and orientations of the fish and the box. The current situation is identified according to the input information. The decision tree is traversed through binary decision rules according to the following five criteria.

Criterion 1: Attacking Region criterion. This criterion classifies the situations into the following two categories according to the relative location between the robotic fish and the Attacking Region:

1) **IAR** (in Attacking Region) situation: The fish is in the Attacking Region;

2) **NIAR** (not in Attacking Region) situation: otherwise.

Criterion 2: Feasible attacking direction criterion. This criterion classifies **IAR** situation into the following two categories:

1) **FAD** (feasible attacking direction) situation: The robotic fish is in the Attacking Region and the orientation of the fish is consistent with the box orientation within a constant limit (in our experiment, we select 45°), as well intersects with the boundary of the box (see Fig. 6 (a));

2) **NFAD** (not feasible attacking direction) situation: The fish is in the Attacking Region but not with a feasible orientation like in FAD (see Fig. 6 (c));

Criterion 3: Synchronization criterion. According to this criterion, FAD situation is divided into the following two situations:

1) **SYN** (synchronizing) situation: The other fish is also in FAD situation (see Fig. 6 (a)), so the two fish can push the box synchronously.

2) **NSYN** (not synchronizing) situation: The other fish is not in FAD situation (see Fig. 6 (b)).

Criterion 4: *Feasible Path* criterion. First we give the following definitions.

Feasible Comfortable Circle (shown in Fig. 7):

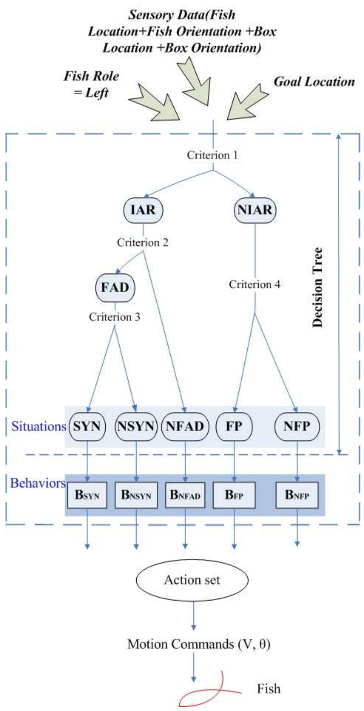
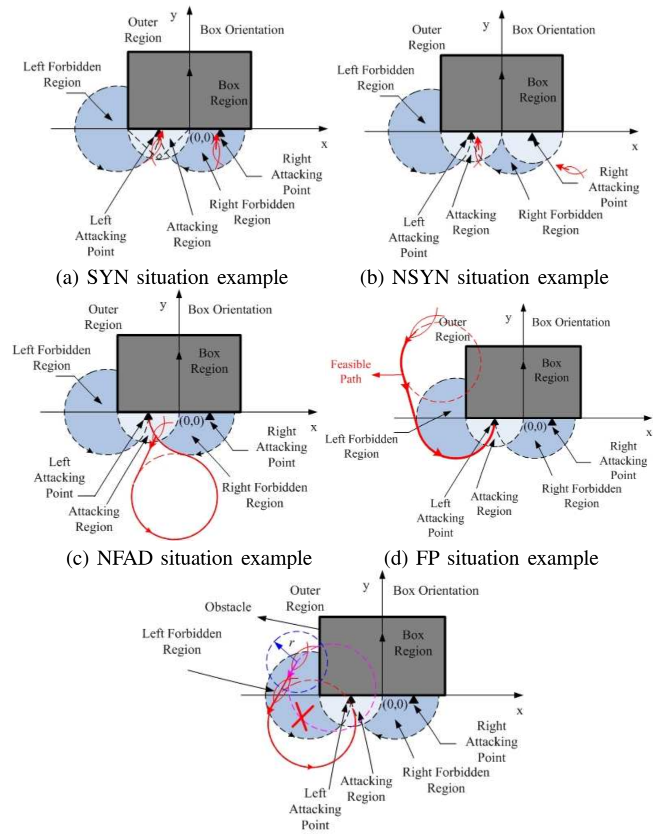


Fig. 5. Decision Tree of CCA. Derived from a binary decision tree, the leaf situations are exclusive and complete. Each situation corresponds to a specific behavior, which can include several actions. Finally, the low level control commands are computed for servicing the selected behavior.



(a) SYN situation example
(b) NSYN situation example
(c) NFAD situation example
(d) FP situation example
(e) NFP situation example

Fig. 6. Situations and Associated Behaviors Design.

Let $P(x,y)$ denote the current position of the robotic fish, ϕ the current orientation, and R_c the Comfortable Radius. The *Feasible Comfortable Circle* is defined as:

if $\exists$ *a Comfortable Circle passing* $P(x,y)$ *with direction* ϕ $\bigwedge$ $\exists$ *a common tangent between the comfortable circle and either boundary of the forbidden regions* $\bigwedge$ *the direction of the tangent is consistent with the directions of the comfortable circle and the boundary of the forbidden region, the comfortable circle is called a Feasible Comfortable Circle associated with the current posture of the fish.*

free path:

A path which is not obstructed by obstacles is called a *free path*.

Semi-Feasible Path:

A *Semi-Feasible Path* is *a path from the current position of the fish to the Left Attacking Point, which is composed of an arc of the Feasible Comfortable Circle, an arc of the Forbidden Region Circle and the directed tangent line of these two circles* (shown in Fig. 7).

Feasible Path:

A *Feasible Path* is a *free Semi-Feasible Path* (shown in Fig. 7).

NIAR situation is divided into two situations depending on whether a *Feasible Path* exists:

1) **FP** (Feasible Path) situation: there exits at least one *Feasible Path* from the current position of the fish to the Left Attacking Point (see Fig. 6 (d)).

2) **NFP** (No Feasible Path) situation: there exits no *Feasible Path* from the current position of the fish (see Fig. 6(e)) .

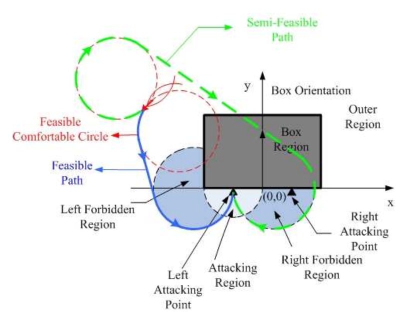


Fig. 7. Feasible Comfortable Circle, Semi-Feasible Path, and Feasible Path.

We only care the leaf nodes of the decision tree: **SYN**, **NSYN**, **NFAD**, **FP**, **NFP**. Obtained through a binary decision tree, these five situations are exclusive and complete.

2) Associated Behavior Design: Next we describe the associated behavior with each situation. The design of the behaviors is required to lead the robotic fish to the destination posture with position the Left Attacking Point and orientation the box orientation, along a fine path under the limit of the minimal turn radius of the fish. In addition, the behaviors shall facilitate the synchronization of both fish's arrival.

1) $\mathbf{B_{SYN}}$: The two fish push the box coordinately since

they have synchronized, while adjusting their heads toward the box orientation (see Fig. 6 (a)).

2) $\mathbf{B_{NSYN}}$: The **NSYN** situation shall be avoided with our motion planning which will be described later. Once this situation appears, the fish in position has to stop to wait for synchronizing with the other fish (see Fig. 6 (b)).

3) $\mathbf{B_{NFAD}}$: In **NFAD** situation, although the fish is in the Attacking Region, it can not push the box because it has not a feasible orientation. In this situation, the fish moves toward outside of the Attacking Region (see Fig. 6 (c)).

4) $\mathbf{B_{FP}}$: The fish approaches the Left Attacking Point along the shortest *Feasible Path* (see Fig. 6 (d)).

5) $\mathbf{B_{NFP}}$: The fish approaches the Left Attacking Point along the shortest *Semi-Feasible Path* while avoiding obstacle if the obstacle is near (within a specific distance r to the fish) (see Fig. 6 (e)).

3) Role Assignment Mechanism: We define two roles in the box-pushing task: role *Left* and role *Right*. The fish assigned with *Left* role is responsible for pushing the box on the left side of the box, and the fish assigned with *Right* role shall push the box on the right side. Let $len(A,B)$ denote the length of the path planned according to CCA from B to A. We use the following cost function to evaluate one assignment:

$$\begin{aligned} &F(A) = |WL(i) - WR(j)| + k(WL(i) + WR(j)),\\ &WL(i) = len(LP, P_i) + CNL_{obj},\\ &WR(j) = len(RP, P_j) + CNR_{obj}, \end{aligned} \tag{3}$$

in which LP (respectively RP) denotes the Left (respectively Right) Attacking Point; P_i (respectively P_j) the current position of fish i (respectively fish j, which is the other fish); NL_{obj} (respectively NR_{obj}) is 1 if there is an obstacle on the planned path from P_i (respectively P_j) to LP (respectively RP), and 0 otherwise; k, C are two positive constants, and $WL(i)$ (respectively $WR(j)$) is the cost when fish i (respectively fish j) is assigned with role *Left* (respectively role *Right*), which represents the approximate time consuming of the fish when approaching the Left (respectively Right) Attacking Point. Either fish can be assigned with role *Left*, or role *Right*. Thus totally there are two possible assignments: fish 1 is assigned with role *Left*, fish 2 is assigned with role *Right*; or fish 1 is assigned with role *Right*, fish 2 is assigned with role *Left*. Optimizing role assignment is to select the assignment which minimizes $F(A)$, the former part of which is to lead the two fish to arrive at the Attacking Points synchronously, while the latter part is to minimize the total time consuming.

D. Fuzzy Logic Based Motion Planning for Box-pushing Task

As mentioned above, it is difficult to facilitate precise control for the robot fish. To synchronize the two robotic fish, and to move the box successfully to the goal location, we employ fuzzy logic control method to plan the motion of the fish, because rule-based fuzzy logic provides a scientific mechanism for reasoning and decision making with uncertain and imprecise information.

1) Motion Planning in Synchronizing Procedure: Aiming to direct the two fish to reach the Attacking Points synchronously, we control the speeds of the fish when approaching the Attacking Points with several fuzzy rules. The inputs are $WL(i)$ and $WR(j)$, representing the approximate time consuming of the fish when approaching the Attacking Points. The outputs are the speeds of the fish. Let VL (respectively VR) denote the speed of the fish with role *Left* (respectively role *Right*). Firstly $WL(i)$ and $WR(j)$ are represented by the linguistic fuzzy sets $\{L, M, S\}$, abbreviated from LARGE, MEDIUM, SMALL, with the membership functions shown in Fig. 8. VL, VR are represented by $\{F, M, S\}$, abbreviated from FAST, MEDIUM, SLOW, with the membership functions shown in Fig. 9. VL and VR are derived using the following intuitive rules:

1) If $WL(i)$ is $L \wedge WR(j)$ is L, then VL is F, VR is F;
2) If $WL(i)$ is $L \wedge WR(j)$ is M, then VL is F, VR is M;
3) If $WL(i)$ is $L \wedge WR(j)$ is S, then VL is F, VR is S;
4) If $WL(i)$ is $M \wedge WR(j)$ is L, then VL is M, VR is F;
5) If $WL(i)$ is $M \wedge WR(j)$ is M, then VL is F, VR is F;
6) If $WL(i)$ is $M \wedge WR(j)$ is S, then VL is F, VR is S;
7) If $WL(i)$ is $S \wedge WR(j)$ is L, then VL is S, VR is F;
8) If $WL(i)$ is $S \wedge WR(j)$ is M, then VL is S, VR is F;
9) If $WL(i)$ is $S \wedge WR(j)$ is S, then VL is M, VR is M.

The final speeds $\overline{VL}$ and $\overline{VR}$ are computed using the Center-of-Gravity (Centroid) defuzzification method as:

$$\overline{VL} = \frac{\sum_{k=1}^{9} VL_k \mu_k}{\sum_{k=1}^{9} \mu_k}, \qquad \overline{VR} = \frac{\sum_{k=1}^{9} VR_k \mu_k}{\sum_{k=1}^{9} \mu_k}, \tag{4}$$

in which μ_k is the conjunction degree of the If part of the kth rule, $(k = 1, \ldots, 9)$, VL_k and VR_k the estimated outputs derived from the kth rule, related to the center of the membership functions of the output viables.

2) Motion Planning in Coordinated Box-pushing Procedure: When the two fish are both in SYN situation, they shall push the box coordinately toward the goal location with their heads. Let θ denote the relative angle of the box orientation to the goal direction. The positive direction of θ is shown in Fig. 4. To move the box toward the goal location, we devise that the orientation of the box is updated through controlling the speeds of the two fish, which are derived through several fuzzy rules. Here θ is the input, and VL, VR are the outputs. We represent θ by the linguistic fuzzy sets $\{PB, PM, PS, Z, NS, NM, NB\}$, abbreviated from positive big, positive medium, positive small, zero, negative small, negative medium, negative big, respectively. The membership functions of θ are shown in Fig. 10. VL, VR are derived through the following reasoning rules:

1) If θ is PB, then VL is F, VR is S;
2) If θ is PM, then VL is M, VR is S;
3) If θ is PS, then VL is M, VR is S;
4) If θ is Z, then VL is M, VR is M;
5) If θ is NS, then VL is S, VR is M;
6) If θ is NM, then VL is S, VR is M;

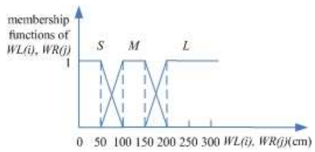


Fig. 8. Membership Functions of WL, WR.

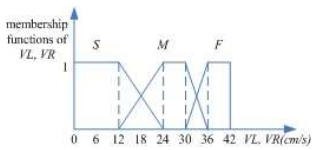


Fig. 9. Membership Functions of VL, VR.

7) If θ is NB, then VL is S, VR is F;

The final speeds $\overline{VL}$ and $\overline{VR}$ are also computed using Centroid defuzzification method, similar to (4).

IV. EXPERIMENTAL RESULTS

Typical experiment scenarios are depicted in Fig. 11. Due to the small contact area of the fish head and the box, it is easy for the fish to overshoot. When overshoot occurs, the fish has to replan the path to return to the Attacking Point. This is very time consuming and inefficient for evaluating our proposed coordination method. To avoid overshoot of the fish, in the experiments, three baffles are added on the lateral of the box to help the fish remain in the Attacking Regions when they reach these regions. Also mechanical devices can be fixed on the fish head to avoid overshoot situation. The goal position is at the top left corner of the swim tank, marked by a blue pole. Fig. 11 (a) shows the scenario at 2.0 s, in which the two fish have started out along the path planned by CCA. The yellow fish is assigned with role *Left*, while the orange fish is assigned with role *Right*. The two fish are trying to synchronize with different swimming speeds. The orange fish is swimming with a lower speed than the yellow fish, since the planned path of the orange fish is shorter than that of the yellow fish. In Fig. 11 (b), the two fish have synchronized successfully, and begin to push the box coordinately toward the goal direction. The speeds of the fish are controlled through fuzzy rules. Fig. 11 (c) shows the scenario at 23.0 s. The box orientation has been changed almost the same to the goal direction. At 30.0 s, the box has been pushed to the goal position successfully, shown in Fig. 11 (d).

V. CONCLUSION

We have presented a coordination method for box-pushing of multiple robotic fish. Considering the complexity of the underwater working environment and the particularity of the movement modes of robotic fish, CCA is developed to simplify the path planning and action decision, with which the environment is divided into several complete, exclusive situations, and for each situation, an appropriate behavior is designed. In addition, fuzzy logic method is employed to synchronize the motion of the two fish, and control the orientation of the box. Experimental results demonstrate that the proposed method is effective.

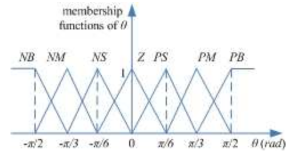
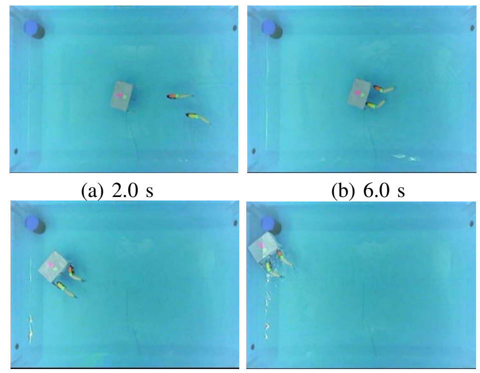


Fig. 10. Membership Functions of The Relative Angle.



(a) 2.0 s (b) 6.0 s

(c) 23.0 s (d) 30.0 s

Fig. 11. Experiment Scenarios.

REFERENCES

[1] J. Yu, M. Tan, S. Wang and E. Chen, "Development of a biomimetic robotic fish and its control algorithm", *IEEE Trans. System, Man and Cybernetics - Part B: Cybernetics*, vol. 34, no. 4, August 2004, pp. 1798-1810.

[2] J. Yu, and L. Wang, "Parameter optimization of simplified propulsive model for biomimetic robot fish", *in Proc. IEEE Int. Conf. Robotics and Automation*, Barcelona, Spain, April 2005, pp. 3317-3322.

[3] Y. Fang, J. Yu, R. Fan, L. Wang, and G. Xie, "Performance Optimization and Coordinated Control of Multiple Biomimetic Robotic Fish", *in Proc. IEEE Int. Conf. Robotics and Biomimetics*, Hongkong and Macau, June 2005, pp. 206-211.

[4] J. Yu, L. Wang, and M. Tan, "A Framework for Biomimetic Robot Fish's Design and Its Realization", *in Proc. of American Control Conference*, Portland, USA, June 2005, pp. 1593-1598.

[5] J. Yu, Y. Fang, W. Zhao, and L. Wang, "Behavioral Design and Strategy for Cooperative Multiple Biomimetic Robot Fish System", *in Proc. IEEE Int. Conf. Robotics and Biomimetics*, Hongkong and Macau, June 2005, pp. 472-477.

[6] D. Barrett, M. Triantafyllou, D. K. P. Yue, M. A. Grosenbaugh, and M. J. Wolfgang, "Drag reduction in fish-like locomotion", *J. Fluid Mech.*, vol. 392, 1999, pp. 183C212.

[7] M. J. Lighthill, "Note on the swimming of slender fish", *J. Fluid Mech.*, vol. 9, 1960, pp. 305C317.

[8] Gerkey, B.P., and Mataric, M.J., "Sold!: Auction Methods for Multirobot Coordination", *IEEE Trans. Robot. Autom.*, vol. 18, October 2002, pp. 758-768.

[9] M. Sfakiotakis, D. M. Lane, and J. B. C. Davies, "Review of fish swimming modes for aquatic locomotion", *IEEE J. Oceanic Eng.*, vol. 24, April 1999, pp. 237-252.

[10] D. Barrett, M. Grosenbaugh, and M. Triantafyllou, "The optimal control of a flexible hull robotic undersea vehicle propelled by an oscillating foil", *in Proc. IEEE AUV Symp.*, June 1996, pp. 1-9.

[11] R. C. Arkin, "Behavior-Based Robotics", Cambridge, MA: MIT Press, 1999.

[12] J. Minguez and L. Montano, "Nearness diagram (ND) navigation: collision avoidance in troublesome scenarios", *IEEE Trans. Robot. Autom.*, vol. 20, February 2004, pp. 45-59.