

Building a 3D Simulator for Autonomous Navigation of Robotic Fishes

Jindong Liu

Department of Computer Science, University of Essex
Wivenhoe Park, Colchester CO4 3SQ, U.K.
Email: jliua@essex.ac.uk

Huosheng Hu

Department of Computer Science, University of Essex
Wivenhoe Park, Colchester CO4 3SQ, U.K.
Email: hhu@essex.ac.uk

Abstract-- This paper presents a 3D simulator used for studying the motion control and autonomous navigation of a robotic fish. The simplified kinematics and hydrodynamics models are created for the simulator, including many other object models such as water, obstacles, sonar sensors and a swimming pool. The experimental results show that the use of this simulator is a realistic and convenient way to develop autonomous navigation algorithms for robotic fishes.

I. INTRODUCTION

In nature, fish has astonishing swimming ability after thousands years evolution. Instead of the conventional rotary propeller used in ship or underwater vehicles, the undulation movement provides the main energy of fish. The observation shows that this kind of propulsion is more noiseless, effective, and maneuverable than the propeller-based propulsion. Recently, some researchers started to build an aquatic man-made robotic systems namely Robotic Fish which could apply the fish-like swimming in order to explore the fish behaviors, detecting the leakage of oil pipeline, sea bed exploration, mine countermeasures and robotics education, etc.

In 1994, the first robot fish named robotuna was developed by MIT [2]. After that, many researchers put forward several kinds of robot fish. The Northwestern University applied Shape Memory Alloy(SMA) on the robotic lamprey [4] which aimed to realize mine countermeasures. In Japan, Nagoya University developed a micro robotic fish using ICPF Actuator [5] and Tokai University realized a robotic Blackbass [3] to research the propulsion of pectoral fins. National Maritime Research Institute in Japan developed many kinds of robotic fish prototypes from PF300 to PPF-09 [6] to exploit the up-down and effective swimming. The Mitsubishi Heavy Industries built a robotic fish named coelacanth robot [16].

Most of previous research focused on the hydrodynamics mechanism of fishlike swimming, the special skin material and mechanical structure of robotic fish models. Although, autonomous navigation poses major challenge for a robotic fish, no research has been done up to now. In order to study the fish-like motion control algorithm (MCA) and autonomous navigation algorithm (ANA) on a robotic fish, it is important to make a quantitative evaluation of the algorithm through many trials of experiments by changing parameters. Although, we already have a real robotic fish, making such trials is a difficult task in the real world, and simulators are built to perform a "virtual experiment".

However conventional simulators such as MOBS [14] for mobile robots have some limitations. They are usually

used for 2D movement and their kinematics and dynamic models are simple compared with robotic fishes because the common mobile robot takes no account of the interaction between itself and the locomotion medium-- air, while the interaction between robotic fishes and water is one of the most important aspects for the robotic fish motion control.

Some researchers focused on the animation of a real fish, such as Tu [10] and William [11]. Their main aims are the fish-like behaviors of artificial fishes but not MCA or ANA. The fish models used in their animation are extracted from real fishes but not from robotic fishes. Up to now, no 3D simulator has been built for the ANA of robotic fishes. We are the first group who try to build such kind of platform for the research of MCA and ANA of robotic fishes. Although we have not finished up it at all, the experiments already showed its great value. Here, the 3D simulator is built up based on a four-joint robotic fish [9].

The rest of this paper is organized as follows. Section II describes the aim, main challenges and supposed conditions. Section III addresses the creation of the robotic fish model and some relative models such as water, obstacles, sensors and the swimming pool. In section IV, the simplified kinematics model and the hydrodynamic model are proposed for the virtual robotic fish. Section V explains the computation flow of the simulator, which includes the fish status update and the collision detection. In Section VI, a preliminary experiment is carried out to verify the feasibility and performance of the simulator. Finally, a brief conclusion and future work are given in section VII.

II. CHALLENGE AND CONDITIONS

The focus of this paper is "how to simulate a robotic fish to make it swim like a real fish and realize its autonomous navigation". The main aims of our simulation work are:

- To simulate the hydrodynamic model of a robotic fish, and understand the relationship between the robotic fish locomotion and the motion control parameters of the joints.
- To develop fish-like MCAs for the robotic fish, and realize or mimic the real fish behaviors such as decelerating/accelerating swim, constant swim, turning and hover.
- To test the ANA in the robotic fish such as to avoid obstacle, to pursue a moving target, to swim in an appointed trajectory, etc

Compared with some traditional simulation work in robotics, the simulator for robotic fishes encounters some special difficulties due to the different locomotion medium--water. Since water is an incompressible fluid, any

movement of a robotic fish will set the water surrounding it in motion and vice versa. The main challenges are:

- The fish's swimming mechanism and mathematical models are not very mature. There is not a simple and direct method to compute the forces acting on a swimming fish.
- The collision computation between robotic fishes or between a robotic fish and other objects is more complex than those in mobile robot simulations because the body shape of a robotic fish is changing when swimming.

To deal with above challenges, this paper makes some simplification on the simulation environment and suppose following preconditions:

- The water in which robotic fishes swim is quasi-steady fluid. Most of fish swimming mechanism researches quoted in this paper is based on this supposition which also makes it easy to build hydrodynamic model.
- For a given parameter vector $E = \{c_1, c_2, k, \omega\}$ (See (2)) for the movement of the robotic fish tail, the value and the direction of thrust force acting on the robotic fish are determined with no respect to the velocity of the robotic fish.
- The viscous drag is considered as the only resistance when the robotic fish swimming.

III. MODELING A ROBOTIC FISH

The whole simulator is built by OO programming method. C++ is selected as the programming language and OpenGL is adopted for the animation display. The robotic fish used for the 3D simulator was purchased from CACIS [9]. It has no onboard sensors. Figure 1 shows its mechanical configuration of it. Based on this prototype, a virtual 3D robotic fish model is created (Figure 2), which is manually constructed using OpenGL.

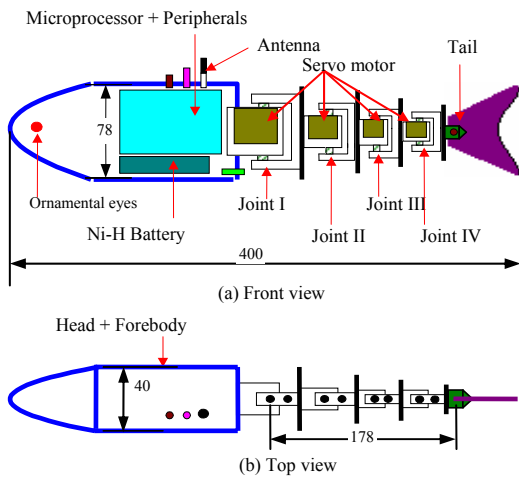


Figure 1 Mechanical Configuration of the Robot Fish[9]

As shown in Figure 2-(a), the fish trunk and fish fins are designed separately. The former consists of eight parts: the nose, the eyes, the neck plane, the first joint plane (I), the second joint plane (II), the third joint plane (III), the fourth joint plane (IV) and the tail. Each part is defined by a group

of six traverse points (A, B, C, D, E, F) which are used to compute the control points of bezier-spline in the OpenGL render processing. These points are arranged into two groups: (ABCD) and (AFED), to build the right-half and the left-half fish body respectively. The values of these traverse points are proportional to the real robotic fish. It is convenient to change the size of the robotic fish model by changing the traverse points. In each joint plane, an axis through point A and point D acts as the axis of a virtual servo motor which represents the servo motor in the real robotic fish. The model could animate fish-like movement when the virtual servo motors turn by each axis in a special way (see section V: Joint Kinematics Model).

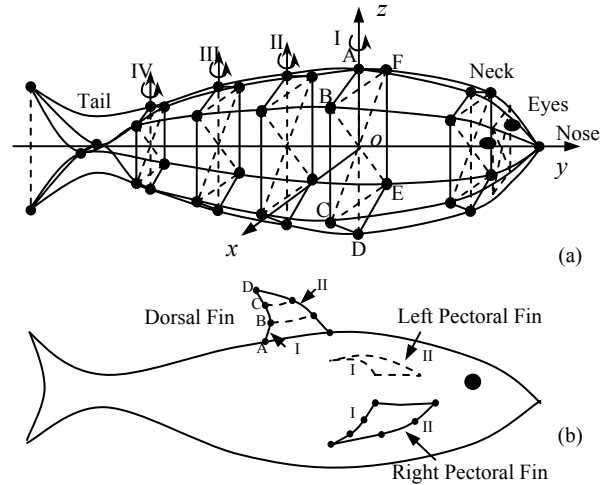


Figure 2 3D robotic fish model

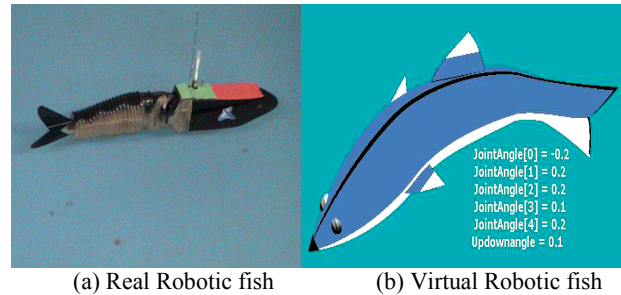


Figure 3 Virtual robotic fish vs. Real one

The robotic fish fins include dorsal fin, left pectoral fin and right pectoral fin. Each of them is described by two bezier-splines (I, II) which are defined by four traverse points (A, B, C, D) as shown in Figure 2-(b). They are designed for future balance control and up-down motion control.

From above traverse points, OpenGL could generate a mesh of a robotic fish body and then map a fish skin texture onto the mesh for realistic display. Figure 3(b) gives an example of the final robotic fish model. To realize autonomous navigation, a robotic fish needs sensors are to perceive its environment. Therefore, we designed models for virtual sonar sensors, infrared sensors, cameras and bumpers. The sonar sensors and the infrared sensors are alternatively selected. All sensors are equipped on the head of the robotic fish model. A virtual bumper is fixed onto the nose and other sensors locate around the eyes.

The camera model is created by adding two more “Context DC”s in the OpenGL program with fish eye viewpoint configuration. When the robotic fish swims, both of the camera “Context DC”s re-render all of the models in the simulator and output images that the robotic fish can “see”. Then the images could be processed by traditional image processing methods. Figure 11 shows two images generated by the camera model.

The bumper sensor model only has two statuses: “ON” or “OFF”, which detects whether the robotic fish head bumps with other objects. The sonar sensor and the infrared sensor are range sensing sensors. To simplify the computation, the sonar sensor is modeled as a wedge. Its cross-section is designed as Figure 4-(b) based on the real sonar sensors (Figure 4-(a)[8]). The transducer and the receiver are viewed as a same point (T/R). A bunch of rays is created in front of the T/R point to simulate the ultrasonic wave. When the robotic fish detects obstacle by sonar, it computes the crossing point of each ray with the obstacle in ambient. The infrared sensor model is designed in a similar way except that the sensitivity region is narrower.

The water wave in the simulator is viewed as the noise source of fish movement. We only consider the noise in the horizon plane paralleled with the water surface. It is supposed that the noise only affects the robotic fish position with no respect to the velocity and the acceleration. The water noise is decomposed into two sub-noises: E_x and E_y which are described by sine functions:

$$E_x = A_x \sin(2\pi / \lambda_x), \quad E_y = A_y \sin(2\pi / \lambda_y) \quad (1)$$

where the A_x, A_y and λ_x, λ_y are the amplitude and wavelength respectively.

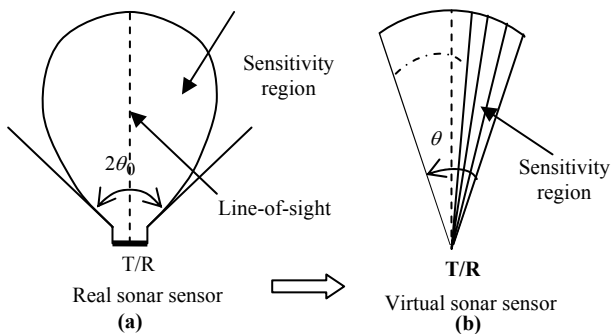


Figure 4 The sonar sensor model

The obstacle model is extracted as a solid column which floats on the water surface and is defined by two parameters: height and radius. In contrast, the swimming pool model is a transparent uncovered box.

IV. KINEMATICS MODEL AND HYDRODYNAMIC MODEL

This section presents three types of models we have built, namely the Joint Kinematics Model, the Hydrodynamics Model, and the Kinematics Model.

A. Joint Kinematics Model

The control of robotic fish movement mainly depends on how the tail joints turn. The kinematics model of the tail joints decides if the robotic fish swims like a real fish, if the

swimming is high efficient and how much propulsion force acting on the robotic fish is generated.

The motion of fish tail could be described by a traveling wave (2), which was originally suggested by Lighthill[7] Its original point is set at the conjunction point between fish head and tail. The swimming of a fish could be viewed as the generation of the traveling wave. The parameter vector $E = \{c_1, c_2, k, \omega\}$ is the key element to determine the kinematics of fish tail.

$$y_{body}(x, t) = (c_1 x + c_2 x^2) \sin(kx + \omega t) \quad (2)$$

where y_{body} is the transverse displacement of the tail unit; x is displacement along the main axis; $k = 2\pi/\lambda$ is wave number; λ is wave length; c_1 is linear wave amplitude envelope; c_2 is quadratic wave amplitude envelope; $\omega = 2\pi f$ is wave frequency; $f = 1/T$ is oscillating frequency of tail, T is cyclic; t is time.

For a real fish, it has tens of vertebrae that could be viewed as tens of mini joints to approximate the wave. So the approximate result is very smooth. But for our robotic fish, it only has four joints, which is impossible to generate exactly fish-like smooth wave. How to use limited joints to approximate the traveling wave in a real fish is one of major challenge problems for the robotic researchers. Yu [9] rewrote the Equation (2) and divided one cyclic to M divisions. A $4 \times M$ look-up table is used to control the motors to approximate the traveling line. Each line of the look-up table consists of 4 turning angles for 4 tail motors. When the data in the table is output to the four servo motors sequentially, the robotic fish will swim like a real one.

The joint kinematics model in the simulator is the same with the above model except the virtual servo motor instead of the real ones. A $4 \times M$ servo look-up table is saved in the simulator and could be directly used for the real robotic fish control after tested.

When the robotic fish turns, a deflected angle θ_d (see C. The Kinematics Model for details) acts on the first joint and the second joint as an offset angle. So, the real control data on the fish joints is the sum of deflected angle and the control data saved in a servo look-up table. If we define a vector $\Theta = \{\theta_1, \theta_2, \theta_3, \theta_4\}$ as the joints status, then:

$$\theta_1 = \theta_{1T} + \theta_d/2, \quad \theta_2 = \theta_{2T} + \theta_d/2, \quad \theta_3 = \theta_{3T}, \quad \theta_4 = \theta_{4T} \quad (3)$$

where $\{\theta_{1T}, \theta_{2T}, \theta_{3T}, \theta_{4T}\}$ are the data in the servo look-up table.

B. The Hydrodynamics Model

The forces acting on a swimming robotic fish are weight, buoyancy and hydrodynamic lift in the vertical direction. In the horizontal heading direction, thrust, friction and inertia drag are found.

In the classification of fish swimming modes [1], the robotic fish used as the prototype in this paper falls between the sub-carangiform mode and the carangiform mode (Both modes are sorted as BCF locomotion modes.) For such two swimming modes, the hydrodynamic model which is related

with the way thrust is generated has long been associated with the added-mass method [12]. As the propulsive wave passes backward along the fish, the momentum of the water passing backward is changed by the movement of the fish tail, which causes a reaction force F_R from water to fish. F_R is decomposed into a lateral F_L and a thrust F_T component which contributes to overall forward propulsion for fish.

In the simulator, the added-mass method is too complex to be applied for the real-time computation. So, we assume two pre-conditions (see Section II) to simplify the hydrodynamic model. The swimming friction drag is supposed as the only resistance against thrust, which is only determined by the travelling wave parameter vector $E = \{c_1, c_2, k, \omega\}$. The longer and faster the fish, the more resistance it would be encountered. So, in case of a given E , there must be a maximum robotic fish velocity U_{\max} to make the friction drag equal to the thrust. At the same time, the robotic fish will keep a constant swimming velocity.

The friction drag could be calculated using the standard Newtonian equation

$$D_v = 0.5C_f S U^2 \rho \quad (4)$$

where C_f is the drag coefficient which depends on the Reynolds number, S is the wetted surface area, U is the forward velocity of the robotic fish and ρ is the water density. The Reynolds number is defined as

$$Re = LU/\nu \quad (5)$$

where L is the tail length and ν is the kinematics viscosity of water (1.12mm²/s, fresh water in 60°F). The laminar and the turbulent drag coefficients are $1.328Re^{-0.5}$ and $0.074Re^{-0.2}$ respectively [13]. In this paper, the drag coefficients C_f is set as the sum of two drag coefficients.

There is a stable prominent parameter named Strouhal number for BCF movement[1]:

$$St = fA/U \quad (6)$$

where $A = 2(c_1x + c_2x^2)|_{x=taillength}$ is the tail-beat peak-to-peak amplitude and f is the oscillating frequency. The St lies in a specific range (namely $0.25 < St < 0.40$) for a constant velocity swimming. For our robotic fish, St is about 0.3. So (6) could be used to compute the maximum velocity U_{\max} . Then the maximum viscous drag $D_{v\max}$ could be calculated using (4). Let thrust force F_{thrust} equal to $D_{v\max}$ we get

$$F_{thrust} = 0.5C_{f\max} S U_{\max}^2 \rho \quad (7)$$

where $C_{f\max} = 1.328(LU_{\max}/\nu)^{-0.5} + 0.074(LU_{\max}/\nu)^{-0.2}$ and $U_{\max} = fA/St$.

In the current version, the hydrodynamic models of turning and up-down movement of the robotic fish are not considered. A simple kinematics model is adopted to compute the turning locomotion and the model of up-down movement is not included temporarily.

C. The Kinematics Model

In the hydrodynamic model, the thrust force F_{thrust} could be exclusively determined by the parameter vector E . Because only the thrust and the viscous drag are considered in a horizon plane, the linear acceleration of a robotic fish could be calculated as:

$$a_t = (F_{ty} - D_v)/m \quad (8)$$

where F_{ty} is the component of F_{thrust} in the robotic fish heading direction and $F_{ty} = F_{thrust} \cos(\theta_d)$. θ_d is the deflected angle which is defined as the angle between the robotic fish heading and the center line of the tail's oscillation (namely Deflected Axis). When the robotic fish swims without turning, $\theta_d = 0$, else $\theta_d \neq 0$. See Figure 5 for details.

For the kinematics model of turning motion, there is almost no available literature. Due to the difficulties in setting up repeatable experiments and the complexity and speed of the movements involved [1], we build it simply by the statistical data from experiments [9]. The angular velocity is calculated as:

$$V_r = 1.2f \sin(\theta_d) \quad (9)$$

where 1.2 is a coefficient from experiments and θ_d is the deflected angle. Here, we suppose the actuation duration of the angular acceleration is too tiny to be considered.

The kinematics model of the up-down motion of the robotic fish is not built in this version. But the height of the robotic fish swimming plane could be configured and the sensors also could detect the difference of height between robotic fishes in the simulator.

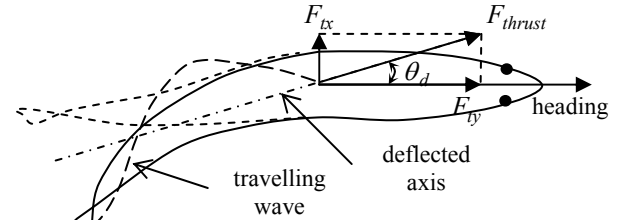


Figure 5 Robotic Fish Kinematics Model

V. THE COMPUTATION FLOW OF THE SIMULATOR

In the process of simulation, an "Update Cycle" is defined to control the status update of robotic fishes and joints in real time. Here the status means kinematics information such as position and velocity. Figure 6 shows the main components in one piece of the update cycle. There are six processing models and one fish behaviors library. The detailed process can be described as follows.

- First, the robotic fish gets the status of four joints $\Theta^i = \{\theta_1^i, \theta_2^i, \theta_3^i, \theta_4^i\}$ and its current status: position P^i , fish heading H^i , linear velocity V_t^i , angular velocity V_r^i and linear acceleration A_t^i . The "Task Compare Model" is called to compare the current fish position with the expected trajectory (namely Task), and outputs the Δ Task for "Make Decision Model".

- Second, the “Detect Obstacle Model” and “Pre-precession Model” are called to sample the environment information by the sensor models and process it to get the direction and range to an obstacle in front of the robotic fish.
- Third, in the “Make Decision Model” the robotic fish makes decisions depending on obstacle information and the result of task compare model. The decision is limited in the scope of “Fish Behaviors Library” which stores all possible fish-like behaviors.
- Fourth, the robotic fish transforms the decision into the expected linear acceleration \hat{A}_t^{i+1} and the expected angular velocity \hat{V}_r^{i+1} for the next period.
- Fifth, in the “Kinematics and Hydrodynamic Model” (see Figure 7), the oscillating frequency f^{i+1} , the deflected angle θ_d^{i+1} and the thrust force F_{thrust}^{i+1} are computed in the Hydrodynamic Model from $f^{i+1}, \theta_d^{i+1}, F_{thrust}^{i+1}$. Parameters $A_t^{i+1}, V_t^{i+1}, V_r^{i+1}, H^{i+1}, P^{i+1}$ of the robotic fish are calculated from the Robotic Fish Kinematics Model and the joints status $\Theta^{i+1} = \{\theta_1^{i+1}, \theta_2^{i+1}, \theta_3^{i+1}, \theta_4^{i+1}\}$ is obtained from the Joints Kinematics Model. Here, $A_t^{i+1}, V_t^{i+1}, V_r^{i+1}, H^{i+1}$ are viewed as the final or “real” $i+1$ status for the robotic fish but the position P^{i+1} is imported into the Noise Model for noise computation.
- Finally, the Noise Model adds virtual water noise into the expected P^{i+1} and generates the “real” P^{i+1} for the fish status $i+1$.

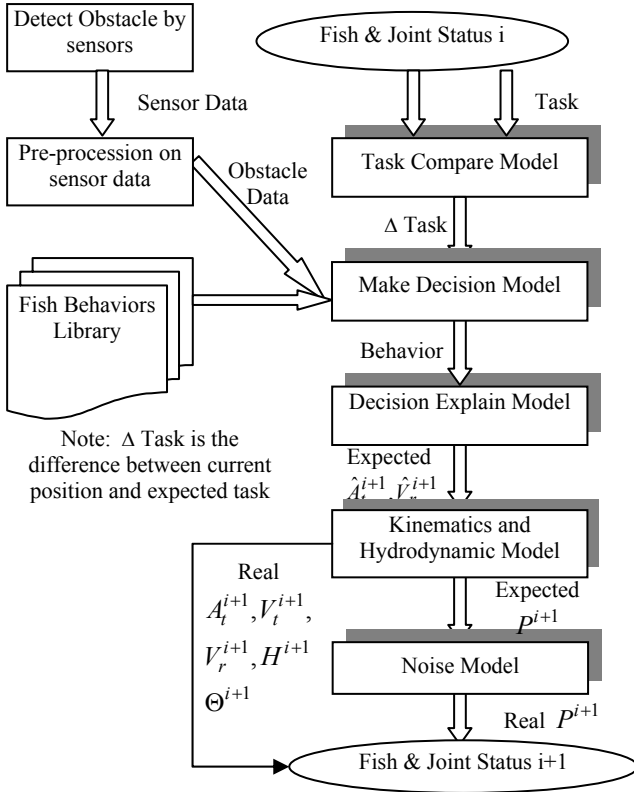


Figure 6 Update Cycle for fish and joints

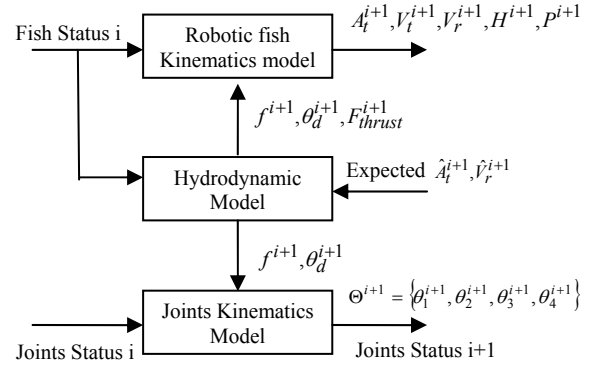


Figure 7 Details of Kinematics & Hydrodynamics Model in the Update Cycle

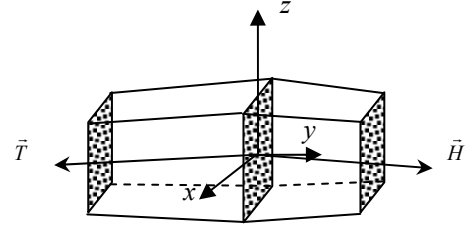


Figure 8 Simplified robotic fish mode

One of big problems in building our 3D simulator is the collision detection between fishes or between fish and other objects because the shape of the robotic fish is changing in swimming. If we adopt the robotic fish model described in section IV for collision detection, the computation cost will be huge due to so many key points in it. We solved it by simplifying the fish mode as two convex boxes as shown in Figure 8. Note that \vec{H} is the heading vector which is equal to H^i and \vec{T} is the combined fish joints vector which is calculated from Θ^i . The “Fast Collision Detection of Moving Convex Polyhedra” [15] method is adopted here.

VI. EXPERIMENTAL RESULTS

A. The Reality of the Simulator

Figure 9 shows the relationship between the oscillating frequency and the max linear velocity of the robotic fish for some given parameters. One curve is drawn from the real experiment [9] and another is sampled from the simulator.

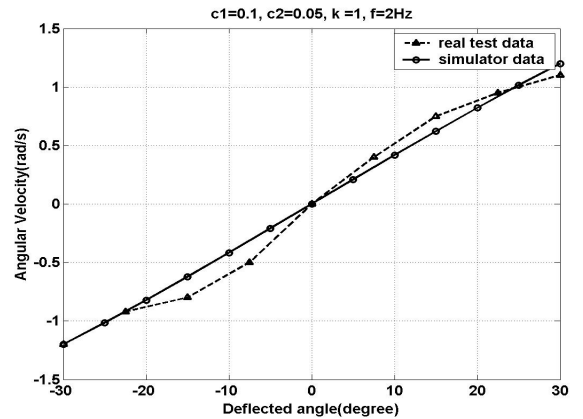


Figure 9 Deflected angle vs. angular velocity

Figure 10 reveals the relationship between the deflected angle and the angular velocity. The curves in such two figures show that the simulator has high reality and could be viewed as a recurrence of the robotic fish.

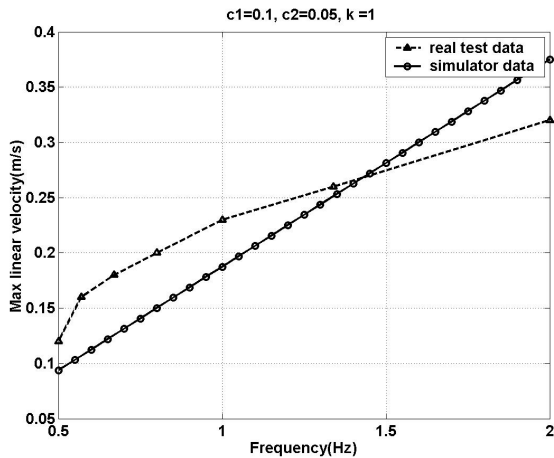


Figure 10 Frequency vs. max linear velocity

B. Autonomous Motion Control Algorithm

There are three types of tasks in the simulator: free swimming, static trajectory and dynamic trajectory. In this paper, the experiment is limited in the free swimming task. Its aim is to test the ability of avoiding obstacle and pool border. The sonar sensor and the bumper sensor are used to detect obstacles and the image sequence at the robotic fish eye viewpoint (i.e. camera model) is generated for future image processing. For example, Figure 11 is the images that fish C could see in frame (a) of Figure 12 at the left eye viewpoint and right eye viewpoint respectively.

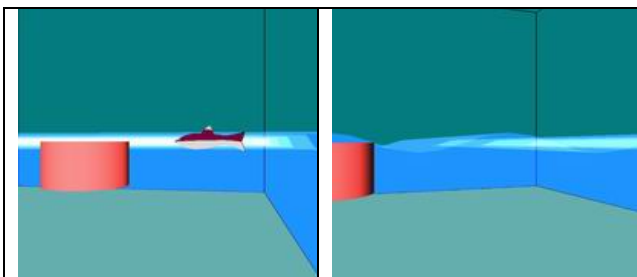


Figure 11 Two pictures at the robotic fish eye viewpoint

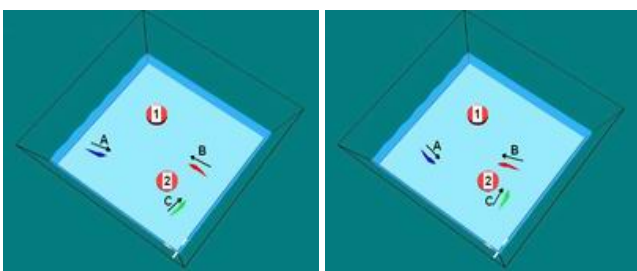


Figure 12 An experiment of free swimming

Figure 12 is two frames of a free swimming experiment. Three robotic fishes (*A, B, C*) wander in a swimming pool with two static obstacles (*1, 2*). They tried to finish a “Free Swimming” task to test a simple motion control algorithm.

When the virtual fish swims, its joint data is recorded for future analysis. The arrow near each robotic fish indicates the heading direction of swimming.

VII. CONCLUSION AND FUTURE WORK

In this paper, a 3D simulator is designed for the autonomous motion control based on a four-joint robotic fish. The complex hydrodynamic model of fish is simplified for the real-time computation of the simulator. For the convenient and quick application of the control method from the simulator to the real world, a servo look-up table is generated for the motion control of the robotic fish. The experiment results have shown that the simulator is of high reality and is a useful platform for robotic fish research. Our future research will focus on the kinematics model of the up-down swimming and the motion control algorithm for the autonomous navigation task of robotic fishes.

ACKNOWLEDGEMENTS

Thanks to London Aquarium for their financial support. Our thanks also go to Prof. M. Tan, Dr. S. Wang and Dr. J.Z. Yu at IACAS and Ian Dukes and Rob Knight at Essex for many useful discussions.

REFERENCES

- [1] M. Sfakiotakis, etc., Review of Fish Swimming Modes for Aquatic Locomotion. *Journal of Oceanic Engineering*, 1999, 24(2), pp. 237-252
- [2] K. Streitlien, G. S. Triantafyllou, M. S. Triantafyllou, Efficient foil propulsion through vortex control, *AIAA Journal*, Vol. 34, 1996, pp. 2315-2319.
- [3] N. Kato. Control performance in the horizontal plane of a fish robot with mechanical pectoral fins., *IEEE J. Oceanic Engr* 2000 25(1), pp. 121-129
- [4] <http://www.dac.neu.edu/msc/burp.html>
- [5] S. Guo, T. Fukuda, Norihiko KATO, Keisuke OGURO. Development of Underwater Microrobot Using ICPF Actuator. *ICRA 2002*, pp. 1829-1834.
- [6] <http://www.nmri.go.jp/eng/khirata/fish>
- [7] M. J. Lighthill, “Note on the swimming of slender fish,” *J. Fluid Mech.*, vol. 9, 1960, pp. 305-317.
- [8] B. Ayrulu and B. Barshan. “Reliability measure assignment to sonar forro bust target differentiation”. *Pattern Recognition* 35, 2002, pp. 1403-1419.
- [9] J.Z. Yu, S. Wang and M. Tan, A simplified Propulsive Model of Biomimetic Robot Fish and Its Realization. Accepted by *IEEE Transactions on SMC Part C*, 2003.
- [10] X. Tu. “Artificial Animals for Computer Animation: Biomechanics, Locomotion, Perception and Behavior”. New York Springer, 1999.
- [11] W. F. Gates. “Animation of Fish Swimming”. U of British Columbia, 2001
- [12] J. M. Anderson, K. Streitlien, D. S. Barrett, and M. S. Triantafyllou, Form and function in fish swimming, *Sci. Amri.*, vol. 251, 1984, pp. 58-68.
- [13] C. S. Wardle and A. Reid, The application of large amplitude elongated body theory to measure swimming power in fish, *Fisheries Mathematics*, J. E. Steele, Ed., New York: Academic, 1977, pp. 171-191.
- [14] <http://robotics.ee.uwa.edu.au/mobs/>
- [15] Rich Rabbitz, Fast Collision Detection of Moving Convex Polyhedra, *Graphics Gems IV*, Academic Press, 1994
- [16] http://www.mhi.co.jp/enews/e_0898.html