

Biological Inspiration: From Carangiform Fish to Multi-Joint Robotic Fish

Jindong Liu, Huosheng Hu

School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K.

Abstract

This paper presents a novel approach to modelling carangiform fish-like swimming motion for multi-joint robotic fish so that they can obtain fish-like behaviours and mimic the body motion of carangiform fish. A given body motion function of fish swimming is firstly converted to a tail motion function which describes the tail motion relative to the head. Then, the tail motion function is discretized into a series of tail postures over time. Thirdly, a digital approximation method calculates the turning angles of joints in the tail to approximate each tail posture; and finally, these angles are grouped into a look-up table, or regressed to a time-dependent function, for practically controlling the tail motors in a multi-joint robotic fish. The paper made three contributions: tail motion relative to the head, an error function for digital approximation and regressing a look-up table for online optimization. To prove the feasibility of the proposed methodology, two basic swimming motion patterns, *cruise straight* and *C-shape sharp turning*, are modelled and implemented in our robotic fish. The experimental results show that the relative tail motion and the approximation error function are good choices and the proposed method is feasible.

Keywords: carangiform fish, fish swimming behaviour, robotic fish, multi-joint robot

Copyright © 2010, Jilin University. Published by Elsevier Limited and Science Press. All rights reserved.
doi: 10.1016/S1672-6529(09)60184-0

1 Introduction

The astonishing swimming abilities of fish has inspired many researchers to work on a new kind of aquatic man-made robotic systems, namely Robotic Fish. Up to now, majority of research work has been focused on fish-like propulsion mechanisms^[1,2], fin materials^[3], remote operation^[4], multi-agent cooperation^[5] and mechanical structures^[6–8]. However, the motion control of robotic fish has not been investigated extensively; particularly control schemes along with modelling of carangiform fish-like swimming.

Typically, there are two main types of motions under discussion for the research of robotic fish: cruising and manoeuvring. Basically, cruising indicates swimming at a constant linear or angular speed, whereas manoeuvring involves actions such as acceleration, deceleration, quick turning, up/down motions, and hovering. Previous robotic fish research concentrated on cruising efficiency and fluid flow effects. For example, an approach carried out in MIT and Draper Laboratories, applied a parameterized kinematic model to their robotic

fish^[9–11]. The parameters were determined using extensive experimental trials. However, their accuracy and robustness remained a challenging issue when the robotic fish tracked an unknown trajectory. A nonlinear control method^[12] and a fuzzy pectoral fins control method^[2] were developed based on quasi-steady fluid flow.

Lauder^[13] studied the morphology and hydrodynamics of fish fin control surfaces, and provided several suggestions for Autonomous Underwater Vehicle (AUV) design. Low and Willy developed a robotic stingray and a robotic knifefish that are able to produce undulations movements on a pair of lateral fins and use a long anal fin to implement motion planning^[14]. McHenry^[15] investigated the mechanical control factors of swimming speed from an undulating fish model, and used a mould of pumpkinseed sunfish to cast fish models with four different flexural stiffnesses. He concluded that body flexural stiffness can control propulsive wavelength, wave speed, *etc.* Hu, *et al.*^[16] applied a modular independent motor-driven mechanism to implement the undulating prototype fin of a robotic fish modelling *Gymnarchus niloticus* fish.

As for maneuvering control, researchers at the National Maritime Research Institute of Japan^[7] analysed the turning mode for their robotic fish, and concluded that there were three different turning modes: oscillation turning, non-oscillation turning, and quick turning. Furthermore, they proposed five swim modes for up-down motions. Some of these modes were already used in their robotic fish, however some were hypothesized and needed further experimentation. Robotuna and VCUUV at MIT adopted pectoral fins to control up-down motions^[1,11,17]. The Blackbass at TOKAI university^[2] realized turning and docking using pectoral fins. Although most robotic fish were controlled using an oscillation turning mode, Kato proposed a new mode for his micro robotic fish^[6], by changing the frequency between the left and the right Ionic Conducting Polymer Film (ICPF) actuators.

Inspired by the movements of carangiform fish such as cod and carp, we investigate the modelling of carangiform fish like swimming motion for multi-joint robotic fish, which can generate variable swimming patterns via undulating their tails as a carangiform fish does. In other words, their thrust and manoeuvrable motions are generated by the last half or third of their body. It should be noticed that we are not considering non-carangiform types of fish in this paper since they need a totally different modelling and control methodology to the one being investigated here. In other words, non-carangiform fish swim using either their active fins (anal, dorsal and pectoral) or single joint in their tail.

To reproduce fish-like swimming by multiple joints robotic fish, an instinctive approach is to undulate the tail of robotic fish as real fish. Currently, in the robotic fish research community, two major methods have been used to realize the tail undulation, which have both advantage and limitation. The first method is the Central Pattern Generator (CPG) method^[18] which generates rhythmic movements for each motor within tail. This method possesses advantages of easy design, quick implementation and ability of online adjustment. It is suitable for the cruising swimming, such as cruise straight where fish repeatedly flap their tail to swim forward along a straight line. However, CPG is difficult to apply on the tail for the maneuvering swimming due to the rhythmic feature of CPG. For example, as a typical maneuvering swimming pattern, sharp turning requires the tail to quickly flick aside in order to turn with a large

turning angle. The tail movements involved in sharp turning are basically not rhythmic motion, and therefore CPG cannot emulate such movement effectively. In addition to the limitation of maneuvering swimming, the CPG method lacks hydrodynamic theory support because it is only based on the hypothesis of CPG control in real fish swimming regardless swimming motion functions. Furthermore, the parameters in the CPG controller are mainly chosen empirically according to human experience.

The second method to control the tail undulation is the trajectory approximation method^[19,20] which utilizes rigid linkages between joints to approximate a serial of tail curves as in real fish swimming. The major advantage of this method is the accurate mimicry results for both cruising swimming and maneuvering swimming, because it is essentially based on a real fish swimming motion rather than the trial-and-error procedure as in the CPG counterpart. Moreover, this method doesn't have the limitation of rhythmic motion existing in the CPG method. However, since the current robotic fish still cannot ideally approximate the target tail curves of real fish, several issues are raised such as a large swing angle of robotic fish head, slow swimming speed. In addition, this method requires off-line calculation and how to tune its parameters online efficiently is still under study.

In our research on the Essex robotic fish, we have tried both of above methods individually and obtained many experimental results. A comprehensive study combining these two approaches is also attempted. This paper proposes an improved trajectory approximation method to conquer the problem of curve disagreement and large head swing angles. In addition, it made online parameter adjustment possible. Our research was conducted on a number of robotic fish built by the Human-Centred Robotics team at Essex, under the financial support from London Aquarium between 2003 and 2007.

The rest of the paper is organised as follows. Section 2 presents a general method of modelling swim patterns for robotic fish. Section 3 proposes a digital approximation method, which is used in the general modelling process. In Section 4, two types of fish swimming motion are modelled following the proposed methodology. Section 5 presents the experimental results obtained from real robotic fishes. These results show the good performance of robotic fish after

applying the proposed modelling method. Finally, a brief conclusion and future work are given in Section 6.

2 Methodology of modelling fish swimming

Before further discussion, we assume that all the swimming activities of carangiform fish can be classified into several basic fish swimming motion patterns, namely Swim Patterns, according to their swimming morphologies, such as cruise straight, cruise in turning, and C-shape shape turning, etc. This classification simplified the mathematical model of fish swimming and the fish body undulation motion in each swim pattern can be represented by an independent function.

There are two coordinate systems are defined. One is a world coordinate system \mathbf{R}^w , as shown (x^w, y^w) in Fig. 1, where the origin is fixed at the connection point B between the fish head and tail, and its x -axis is aligned along the swimming direction of fish. The movement of a whole fish body in \mathbf{R}^w is defined as body motion described as $f_B(x, t)$. Another coordinate system is a head-fixed system \mathbf{R}^h , as shown (x^h, y^h) in Fig. 1, where the origin is the same point as \mathbf{R}^w , but its x -axis is aligned along the fish head rather than the swimming direction. The movement of the fish tail in \mathbf{R}^h is called tail motion denoted as $f_T(x, t)$.

Fig. 2 shows a typical routine to model fish swimming. The body motion function $f_B(x, t)$ of a specific swim pattern is generally obtained from biologists. In Section 4.1, we give an example of $f_B(x, t)$ for cruise straight swim pattern. After getting $f_B(x, t)$, a corresponding tail motion function $f_T(x, t)$, which is a relative motion function between the fish tail and its head can be deduced. The purpose of generating tail motion functions is to control the tail joints to move in a reference coordinate system; i.e. the head fixed coordinate system \mathbf{R}^h , because all the turning actions of joint motors occur in \mathbf{R}^h rather than in \mathbf{R}^w . Prior to calculating $f_T(x, t)$, we assume that the fish head section, i.e. the portion from the nose tip to the head-tail connection point B , is rigid. This assumption is reasonable for carangiform fish due to their biological nature features. Therefore, the head part of $f_B(x, t)$ can be replaced by a time-dependent linear function $y = c_3x$, where c_3 is the same as the derivative of $f_B(x, t)$ respect to x at $x = 0$, i.e.

$$c_3 = \left. \frac{\partial f_B(x, t)}{\partial x} \right|_{x=0}.$$

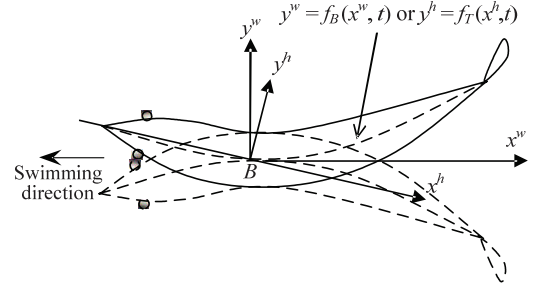


Fig. 1 Coordinate system definitions.

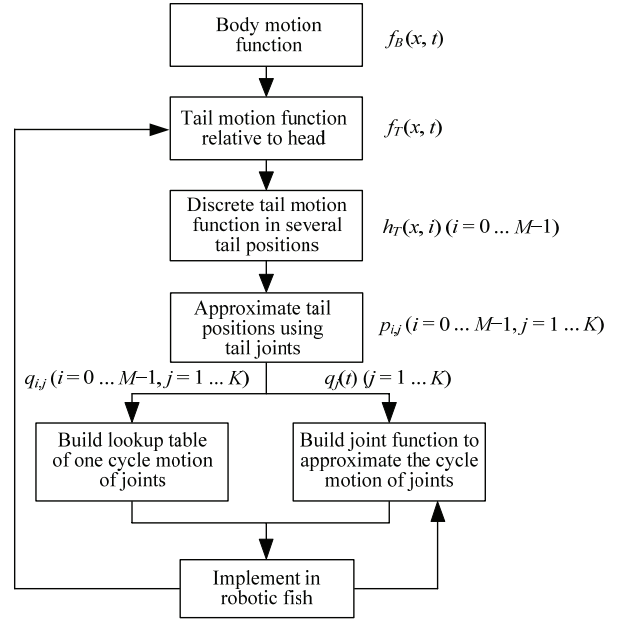


Fig. 2 Flow chart of our methodology to model fish swimming. Module A (Build lookup table) and module B (Build joint function) are exclusive each other. For a quick and light robotic fish implementation, module A is proper. For online learning purpose, module B is proper. The feedback from module of “implementation in robotic fish” only exists for module B in order to give feedback for optimisation purpose.

For simplicity, in the following part of this paper, we use $f'_B(x, t)$ to replace

$$\frac{\partial f_B(x, t)}{\partial x}.$$

The tail motion function $f_T(x, t)$ can be obtained by subtracting the linear function $y = c_3x$ from $f_B(x, t)$.

Eq. (1) gives the relationship between $f_T(x, t)$ and $f_B(x, t)$, while Fig. 3 shows the deduction procedure. In the head fixed coordinate system \mathbf{R}^h , it also should be noted that the derivative of $f_T(x, t)$ respect to x at the connection point B should be zero, considering that the fish body is a smooth curve and the fish head is fixed on the x -axis in \mathbf{R}^h . In Eq. (1), it can be readily found that $f'_T(x, t)|_{x=0} \equiv 0$. This justifies our deduction from $f_B(x, t)$

to $f_T(x, t)$. Fig. 4 shows an example of $f_T(x, t)$ of cruise straight swim pattern.

$$f_T(x, t) = \begin{cases} f_B(x, t) - xf'_B(x, t)|_{x=0} & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (1)$$

The basic idea of reproducing the fish swim patterns in our robotic fish is to drive the multiple mechanical joints within the tail to imitate the same tail motion as a real fish. However, up to now, an analytical solution is still unavailable to deduce the corresponding joint motion from the tail motion function. This is basically due to the difficulties in solving the motion function $f_B(x, t)$ with the constraints of the joint length. Therefore, a digital approximation method is proposed here and discussed fully in Section 3. In this method, the tail motion function is discretized into M tail postures $h_T(x, i)$ ($i=0 \dots M-1$) over time. Fig. 5 gives an example of the tail postures of a cruise straight swim pattern. By this means, real fish swimming can be mimicked by approximating each tail posture with rigid linkages between joints. Fig. 6 shows an example of an approximation of a tail posture $h_T(x, i)$ with four joints. In this example, there are in total four linkages, i.e. (I, II, III, IV), between joints, with position (x_k, y_k) ($k=0 \dots 3$). (x_4, y_4) represents the endpoint of the last linkage. $p_{i,j}$ ($i=0 \dots M-1, j=1 \dots 4$) are the slope angles of the linkages. $q_{i,j}$ ($i=1 \dots M-1, j=1 \dots 4$) are the actual control angles; which are angles to turn a joint relative to its anterior linkage. For example, $q_{i,3}$ means that the turning angle of the 3rd joint (at (x_3, y_3)), relative to linkage II for the tail posture $h_T(x, i)$. In general, for a robotic fish with K tail joints, the approximation result for a tail motion function $f_T(x, t)$ can be represented by $q_{i,j}$ ($i=0 \dots M-1, j=1 \dots K$).

In practice, two methods can be applied to control joints to turn according to the approximation result $q_{i,j}$. The first method is a lookup table. A lookup table is built to contain all the turning angles of $q_{i,j}$ ($i=0 \dots M-1, j=1 \dots K$) for all turning actions. The servo in each joint are one by one controlled to turn following $q_{i,j}$. This method is easy to implement. However, it is difficult to revise $q_{i,j}$ during online parameter optimisation, because the lookup table is constructed from at least 32 angle values for each joint, to guarantee a smooth approximation. It is necessary to revise the parameters of $f_T(x, t)$, in order to change $q_{i,j}$. The second method is to build a joint motion function $q_j(t)$, which is a non-linear regression of $q_{i,j}$ ($i=0 \dots M-1$). $q_j(t)$ can be a polynomial function or other

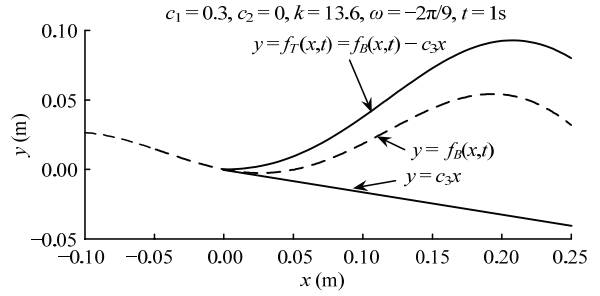


Fig. 3 The relationship between the body motion function and the tail motion function.

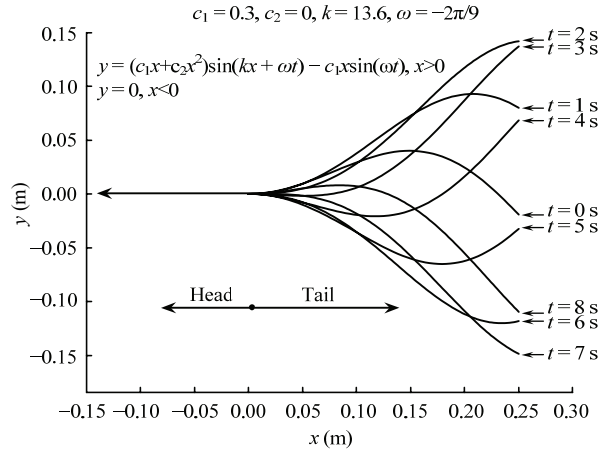


Fig. 4 The corresponding tail motion function resulting from the body motion function of cruise straight swim pattern.

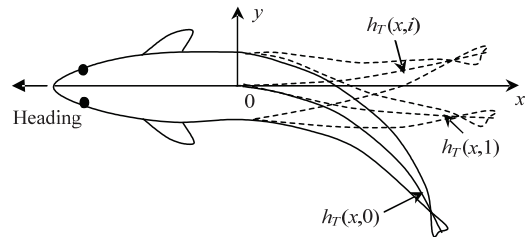


Fig. 5 Tail postures, i.e. the discretized tail motion function $h_T(x, i)$ in a head fixed coordinate system.

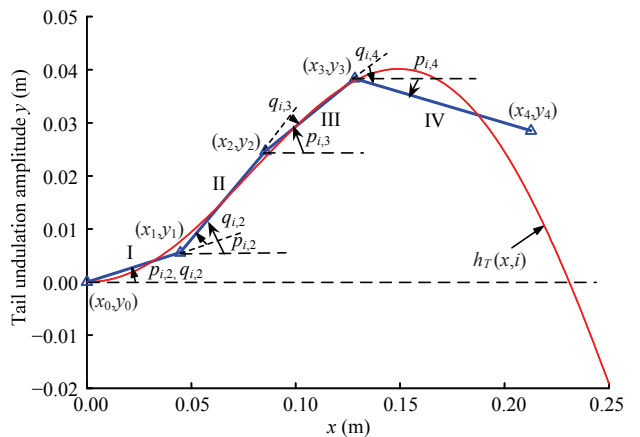


Fig. 6 An example of a tail posture approximation result using our method. The original point is the connection between the fish head and tail. (x_k, y_k) is the joint position. It is assumed that the tail has four joints in total.

non-linear function, e.g. a sine function. Therefore, it is convenient to adapt the parameters of swim patterns, because $q_j(t)$ has limited parameters, and is possible to adjust online. According to the advance of the second method, we chose it in our research. Please see Section 4.1 for the details of the regression procedure.

For different swim patterns, the body motion function also varies. Once we obtain a body motion function $f_B(x, t)$, a series of corresponding joint motion function $q_j(t)$ can be calculated. When some swim patterns haven't body motion function yet, such as sharp turning, the calculation of joint motion function can be performed from the second module, $f_T(x, t)$, in Fig. 2.

3 Digital approximation

Real fish have a number of vertebrae that act as many mini-joints to smoothly approximate a tail motion function during swimming. Mimicking fish swimming by a robot needs approximating the tail motion function. Since our robotic fish have limited joints due to mechanical limitations; it is therefore difficult to generate a smooth wave. This section is to investigate how to use limited joints to approximate a tail continuous motion function, namely digital approximation method.

The objective of the digital approximation method is to approximate smooth tail contours by using rigid tail linkages, so as to mimic real fish swimming as closely as possible. Assuming the tail motion function $f_T(x, t)$ is discretized into M tail postures, i.e. $h_T(x, i)$ ($i = 0 \dots M - 1$). Consider a specific tail posture $h_T(x, i)$ and rewrite it as $f(x)$, i.e. $f(x) = h_T(x, i)$. Meanwhile, assume there are K joints in the tail, and the length of each linkage is denoted as l_j ($j = 1 \dots K$). Before further discussion, some terms are defined in Fig. 7 for convenience. For a give linkage j ($1 \leq j \leq K$), a "Base" point is defined as its start point, which varies according to its anterior linkage $j - 1$. The coordinate of the Base point is $(Base_x, Base_y)$. Hence, one can find that the Base point of the first linkage

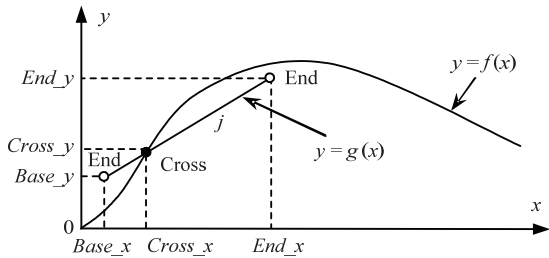


Fig. 7 Terms definition for the approximation method.

is fixed at the original point 0, which is the connection point between the fish head and flapping tail. Similarly, the "End" point is defined as the end of a linkage, which is decided by the approximation between the linkage j , and $f(x)$. The coordinate of the End point is (End_x, End_y) . Accordingly, it can be readily found that the End point of linkage j is the same point as the Base point of linkage $j + 1$. Moreover, the End point of the last linkage K is the tail tip point. By this means, linkage j can be represented by a linear function $y = g(x)$ with two constraint points: the Base point and the End point. A "Cross" point is thereafter defined as the first crossing point between $g(x)$ and $f(x)$, counting from the "Base" point to the "End" point. The coordinate of the Cross point is $(Cross_x, Cross_y)$.

A crucial issue in approximation methods is how to define a proper error representation to evaluate approximation results. Generally, Root Mean Square error (RMS) is widely adopted, which can be written as

$$\int_{Base_x}^{End_x} \sqrt{(g(x) - f(x))^2} dx,$$

for our robotic fish application. However, considering the physical meaning behind the swimming function, above RMS expression cannot be directly implemented. As a matter of fact, the basic objective of approximating a swimming function, i.e. the tail motion function, is to simulate a real fish swimming, and the basic difference is from the thrust force generated by the tail, which has to be taken into account as defining the approximation error function. There are two main methods for accommodating it: added-mass method and lift-based (vorticity) method^[21]. The carangiform type, which our robotic fish belong to, is associated with the added-mass method.

In a physical perspective, as the propulsive wave passes through the fish body, the momentum of the water passing backwards is changed by the movement of the fish tail, which in turn generates a reaction force to push the fish. The reaction force forms the forward thrust force for fish to swim. The mass of the water passing backwards is called added-mass. Usually, the bigger the added-mass, the larger the thrust force. The magnitude and direction of the reaction force is determined by how the tail pushes the water aside. Therefore, when using a fixed-length linkage to approximate the wave of a fish tail, the ideal solution is to equate the added-mass pushed away by the linkage to that by the

real fish. Therefore, two error indexes have to be considered, the magnitude error and direction error, since the added-mass is basically a vector. The magnitude error is

$$e(x) = \left| \int_{Base_x}^{End_x} [g(x) - f(x)] dx \right|. \quad (2)$$

Direction error is minimized if the magnitude error has been minimized as the Base point of the first joint is coincident with $f(x)$. Therefore, $g(x)$ is always very close to $f(x)$, due to the smooth gradient changing of $f(x)$ (this can be observed from real fish swimming). The pushing directions of $g(x)$ and $f(x)$ can be viewed as the same as long as the magnitude error of the anterior joint has been minimized, therefore no separate function is used for the direction error.

Generally, when using K linkages to approximate a given tail motion function, the task can be rephrased as finding a series of End points to minimize K error functions

$$e_{i,j}(x) = \left| \int_{Base_x_{i,j}}^{End_x_{i,j}} [g_{i,j}(x) - f_i(x)] dx \right|, \quad (3)$$

$$(i = 0 \dots M-1, j = 1 \dots K)$$

where $f_i(x) = h_T(x, i)$, $g_{i,j}(x) = k_{i,j}x + b_{i,j}$,

$$k_{i,j} = \frac{End_y_{i,j} - Base_y_{i,j}}{End_x_{i,j} - Base_x_{i,j}},$$

and $b_{i,j} = Base_y_{i,j} - k_{i,j} Base_x_{i,j}$. The constraints between the joints are

$$\begin{cases} Base_x_{i,j} = Base_y_{i,j} = 0 & j = 1 \\ Base_x_{i,j} = End_x_{i,j-1} & j = 2 \dots K \\ Base_y_{i,j} = Base_y_{i,j-1} & j = 2 \dots K \end{cases}, \quad (4)$$

and the length constraint of each linkage is as follows

$$(End_x_{i,j} - Base_x_{i,j})^2 + (End_y_{i,j} - Base_y_{i,j})^2 = l_j^2, \quad (5)$$

where l_j is the length of the j -th linkage.

Now the approximation task can be redescribed as: given a desired tail posture $f_i(x)$, the Base point of linkage j , and the linkage length l_j , find the End point which minimizes the error function $e_{i,j}(x)$. Note that the End point is not necessarily located on the curve of $f_i(x)$. One possible solution to this problem is to search the analytical expression of $e_{i,j}(x)$, and solve the equation

$$\frac{de_{i,j}(x)}{dx} = 0.$$

However, it is usually difficult to obtain an analytical form of $e_{i,j}(x)$, because of the constraints of Eqs. (4) and (5). In fact, the constraint of Eq. (4) requires that $Base_x_{i,j}$ in Eq. (3) is dependent on approximation results, i.e. $End_x_{i,j-1}$, of the anterior linkage $j-1$. The constraint Eq. (5) is a second-order polynomial function, and it is therefore hard to substitute Eq. (5) into error function of Eq. (3).

The second solution is to use numerical methods to find End points, and the prior defined Cross point (Fig. 7) is utilized. Firstly, a crossing ratio is introduced as

$$R_c = \frac{Cross_x_{i,j}}{Base_x_{i,j}}, \quad R_c \in [0, 1],$$

which is a section ratio of the Cross point on linkage j . For a given R_c , the coordinate of the Cross point can be obtained by solving the following equations

$$\begin{cases} (x - Base_x_{i,j})^2 + (y - Base_y_{i,j})^2 = (R_c l_j)^2 \\ y = f_i(x) = h_T(x, i) \end{cases}, \quad (6)$$

$$i = 0 \dots M-1, j = 1 \dots K.$$

Eq. (6) can be solved by using an iterative method, which searches the cross point between a circle, denoted by the first equation of Eq. (6), and a curve $h_T(x, i)$. The result of Eq. (6) under the condition of R_c , is denoted as $\{(Cross_x_{i,j}, Cross_y_{i,j}) | R_c\}$. Once we obtain the Cross point, the End point and $g_{i,j}(x)$ can be easily obtained, because of the linear relationships between them (Fig. 7). After substituting the End point and Base point into Eq. (3), the approximation error under the gain R_c can be computed. This conditional error is denoted as $(e_{i,j}(x) | R_c)$.

If all the $(e_{i,j}(x) | R_c)$ s ($i = 0 \dots M-1, j = 1 \dots K$) is available for R_c ranging from 0 to 1, the best approximation result can be chosen as the one with minimal error. To implement this method in a numerical way, R_c is first discretized into $R_{c,0}, R_{c,1}, R_{c,2}, \dots, R_{c,n}$ from 0 to 1 with an equal step $Step_R = 1/n$. After substituting them into Eq. (6), one can obtain a series of approximation errors: $(e_{i,j}(x) | R_{c,k})$ ($k = 0 \dots n$). Thus, the minimal approximate error $e_{i,j \min}$ can be obtained as:

$$e_{i,j \min} = \min_{k=0 \dots n} (e_{i,j}(x) | R_{c,k}).$$

The corresponding $\{(End_x_{i,j}, End_y_{i,j}) | R_{c,k}\}$ to $e_{i,j \min}$, is thereafter set as the approximation position for the j -th linkage and i -th tail posture. Further, the gradient $p_{i,j}$ of the linkage can be calculated easily.

After the approximation computation of all linkages for all tail postures, the turning angles of each joint motor, i.e. $q_{i,j}$ in Fig. 6, can be calculated using Eq. (7). They are the practical control inputs to the robotic fish used to control the motors.

$$\begin{cases} q_{i,j} = p_{i,j} & j = 1 \\ q_{i,j} = p_{i,j} - p_{i,j-1} & j = 2 \dots K \end{cases} \quad (7)$$

As an example of tail posture approximation, Fig. 6 shows an approximation result for four joints (I to IV). It is clear that when the Base point and linkage length are fixed, the End point of each joint is the key element used to minimize approximation error. In contrast, the authors in Ref. [5] assumed all End points fall onto the ideal wave. As a result, much larger error was generated according to added-mass theory, as shown in Fig. 8 for a comparison between their method and the proposed method in this paper.

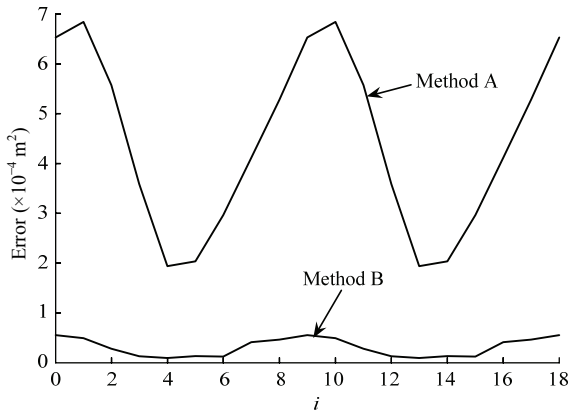


Fig. 8 The magnitude error of one fish pattern using the approximation method in Ref. [5] (labelled Method A) and the proposed method in this paper (labelled Method B). The corresponding fish swim pattern is “cruise straight” (see Section 4) in flapping frequency of 1.3 Hz. The x -axis is the number of tail postures. The interval time of each unit is about 0.043 s.

4 Modelling swim patterns

4.1 Cruise straight swim patterns

The motion of a fish body during *cruise straight* could be described by a travelling wave Eq. (8), which was originally suggested in Ref. [22] and widely used in the robotic fish research community^[4,9,23]. The parameter of the fish travelling wave changes depending on the fish type and their kinetics status in water.

$$y = f_B(x, t) = (c_1 x + c_2 x^2) \sin(kx + \omega t) \quad (8)$$

where the original point is set as the conjunction point between the fish head and tail. y is the transverse dis-

placement of a tail unit; the x -axis is the centre line of the undulation wave of cruise straight; $k = 2\pi/\lambda$ is the wave number; λ is the wave length; c_1 the linear wave amplitude envelope; c_2 the quadratic wave amplitude envelope; $\omega = 2\pi f$ the wave frequency; and t time.

Fig. 9 shows an example of the fish body function, in which the fish undulates its tail to generate a propulsion wave. Following the methodology in Section 2, a tail motion function can be deduced by subtracting a linear function $y = c_3 x$ from the body function. c_3 can be calculated as the first-order derivative of $f_B(x, t)$ at $x = 0$, i.e.:

$$\begin{aligned} c_3 &= f'_B(x, t)|_{x=0} = ((c_1 + 2c_2 x) \sin(kx + \omega t) + \\ &\quad (c_1 x + c_2 x^2) k \cos(kx + \omega t))|_{x=0} \\ &= c_1 \sin(\omega t). \end{aligned} \quad (9)$$

So the tail motion function corresponding to Eq. (8) is

$$\begin{aligned} f_T(x, t) &= f_B(x, t) - c_3 x \\ &= (c_1 x + c_2 x^2) \sin(kx + \omega t) - c_1 x \sin(\omega t). \end{aligned} \quad (10)$$

Fig. 4 shows the tail motion function according to $f_B(x, t)$ of Eq. (8). When comparing Fig. 9 and Fig. 4, it is noticeable that the amplitude of the relative motion is larger than that of body motion. In other words, the tail has to undulate itself with a bigger amplitude to achieve the swimming wave because of the head swing during swimming.

After obtaining the tail motion function $f_T(x, t)$, a digital approximation can be performed using the method described in Section 3. First, $f_T(x, t)$ is discretized as:

$$\begin{aligned} h_T(x, i) &= (c_1 x + c_2 x^2) \sin(kx - \frac{2\pi}{M} i) - \\ &\quad (c_1 x) \sin(-\frac{2\pi}{M} i), i \in [0, M - 1], \end{aligned} \quad (11)$$

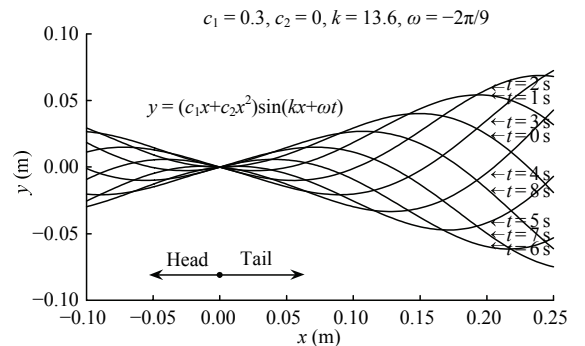


Fig. 9 An example of a body motion function.

where i is a serial number in an undulation cycle. M is a positive integer which represents the resolution of the discrete travelling wave.

Because ω in Eq. (8) is negative, to generate thrust in the $-x$ direction, a minus sign is put before $2\pi/M$ in Eq. (11). By substituting $h_T(x, i)$ into Eq. (6) in Section 3, and assuming that $K = 4, M = 18$, the digital approximation can be finished in Matlab, and the joint positions during one cycle of cruise straight is demonstrated in Fig. 10. Note that the length of the linkages is set as $[l_1, l_2, l_3, l_4] = [0.045, 0.045, 0.045, 0.085]$ m, corresponding to our robotic fish. Finally, the joint angle $q_{i,j}$ is calculated using Eq. (7) (see Fig. 11).

In our practical realization of robotic fish, the series of joint angles $q_{i,j}$ are regressed to a time-dependent function $q_j(t)$, where j is the index of joints. Because $q_j(t)$ must be periodic in periodic swimming, and its shape

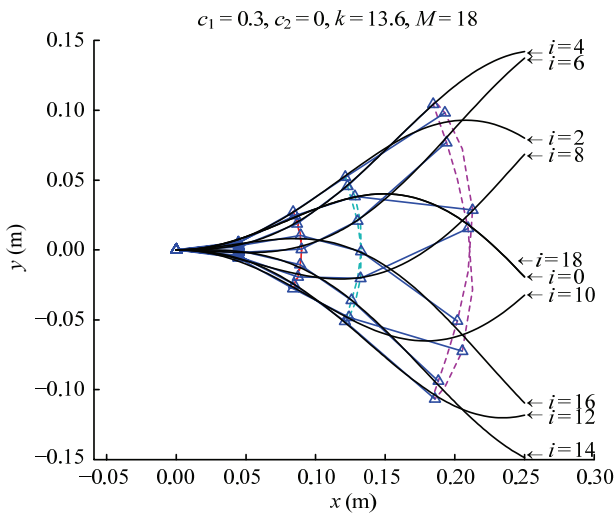


Fig. 10 The approximation result of a 4-joint tail for cruise straight.

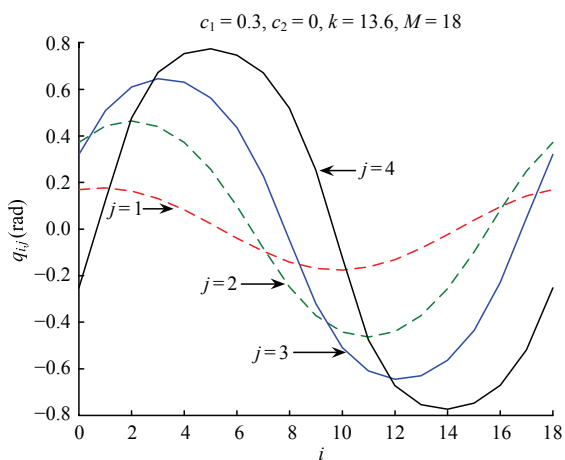


Fig. 11 The joint angle calculated using the digital approximation method for cruise straight.

similar to a sine function (see Fig. 11), it can therefore be represented using the Fourier series:

$$q_j(t) = \sum_{n=1}^{\infty} a_{n,j} \sin(n\omega t + \varphi_{n,j}) . \quad (12)$$

Generally, only the first term in Eq. (12) is taken without the higher order components in order to limit the parameter number in online optimisation (please refer to Ref. [24] for the details of parameter optimisation). The 1st phase of the first joint $\varphi_{1,1}$ is set to zero because the turning angle of this joint at the start moment of swimming is 0. Therefore, the form of the joint angle to be controlled can be written as follows:

$$q_j(t) = a_j \sin(\omega t + \varphi_j), j = 1 \dots K . \quad (13)$$

4.2 C-shape sharp turning swim patterns

This section is devoted to illustrate how to apply our method for another swim pattern, C-shape Sharp Turning (CST), which doesn't have explicit body motion yet. In our pervious research, a tail motion function has been proposed and hydrodynamically analyzed^[25]. In the following paragraph, we will briefly introduce the tail motion function, then the methodology discussed in Section 2 will be applied for it. The details of the kinematic and hydrodynamic functions are omitted here and please refer to Ref. [25] for details.

CST is a special fish maneuvering motion whereby fish bend their rear body quickly in a C-shape to achieve an emergent changing of its swimming direction. Fig. 12 is a C-shape sharp turn sequence recorded from an adult carp^[26]. This sequence is divided into two stages: a shrink stage and a release stage. In the shrink stage (0 ms to 30 ms in Fig. 12), the tail bends to one side very quickly. In the release stage (40 ms to 100 ms), the tail straightens at a relatively slow speed from the middle section of the body, to the tip of the fish's tail. The shape

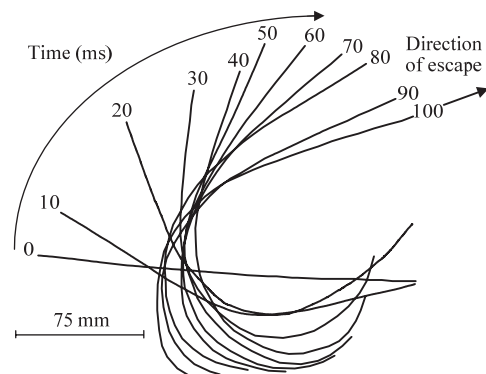


Fig. 12 A CST sequence recorded from an adult carp^[26].

changing of the tail during a sharp turn, can be imagined like that of a fish embracing a virtual circle using its flexible tail. The virtual circle changes its centre and radius over turning time. The circumference of the virtual circle determines the tail shape. For example, at time 0 ms in Fig. 12, the virtual circle has an infinite radius, and its centre point is far from the tail; therefore the tail shape is roughly a straight line. At time 10 ms, the virtual circle decreases its radius, and pulls the centre point towards the tail; then, the tail starts to bend. The smaller the circle radius, the greater the tail bending angle. Based on this observation, a circle function Eq. (14) can be written in the head-fixed coordinate system \mathbf{R}^h to describe the tail shape. Please see Fig. 13 for the shape of a virtual circle (the dashed line).

$$(x - x_c(t))^2 + (y - y_c(t))^2 = y_c^2(t) \quad (14)$$

where $x_c(t)$ and $y_c(t)$ form the changing of the circle centre over time. $y_c(t)$ is also used for the circle radius, because the virtual circle is always tangent to the x -axis.

If we obtain the function of $x_c(t)$ and $y_c(t)$, we can generate a series of circle functions by discretizing time t into $(0 \dots M - 1)$. These circle functions can be viewed as tail postures $h_T(x, i)$ (Fig. 5) of a sharp turning. The digital approximation method (Section 3) can then be used to calculate the turning angles of joints in order to approximate these postures. Now, the question of modelling CST is converted to “how to find proper functions for $x_c(t)$ and $y_c(t)$ ”.

In Ref. [25], the form of $y_c(t)$ and $x_c(t)$ were proposed as Eqs. (15) and (16).

$$y_c(t) = \begin{cases} cy_1 + (cy_0 - cy_1) \frac{(t - t_1)^k}{(t_0 - t_1)^k}, & t \in [t_0, t_1] \\ cy_1 + (cy_1 - cy_2) \frac{(t - t_1)^m}{(t_2 - t_1)^m}, & t \in [t_1, t_2] \end{cases} \quad (15)$$

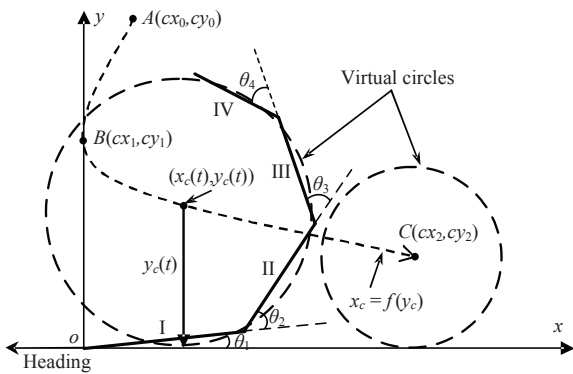


Fig. 13 The virtual circle function of C-shape sharp turning, and an example of the approximation result.

$$x_c(t) = \begin{cases} a_{11}y_c^2(t) + a_{12}y_c(t) + a_{13}, & t \in [t_0, t_1] \\ a_{21}y_c^2(t) + a_{22}y_c(t) + a_{23}, & t \in [t_1, t_2] \end{cases} \quad (16)$$

where

$$[a_{11} \ a_{12} \ a_{13}] = [cx_0 \ cx_1 \ 0] \begin{bmatrix} cy_0^2 & cy_1^2 & 2cy_1 \\ cy_0 & cy_1 & 1 \\ 1 & 1 & 0 \end{bmatrix}^{-1},$$

and

$$[a_{21} \ a_{22} \ a_{23}] = [cx_1 \ cx_2 \ 0] \begin{bmatrix} cy_1^2 & cy_2^2 & 2cy_1 \\ cy_1 & cy_2 & 1 \\ 1 & 1 & 0 \end{bmatrix}^{-1}.$$

An example of the trajectory of $(x_c(t), y_c(t))$ is shown in Fig. 13. It is divided into two secondary curves (Fig. 13) – curve (AB) and (BC) ; point $A(cx_0, cy_0)$, represents the centre of the initial circle, when the tail just starts to shrink; point $B(cx_1, cy_1)$ is the centre of the virtual circle when the tail finishes the shrink stage, and enters the release stage; while point $C(cx_2, cy_2)$ is the centre of the end circle which is at the end of the release stage. The stage of the circle centre on curve (AB) , is defined as the shrink stage ($t \in [t_0, t_1]$), and for curve (BC) , is defined as release stage ($t \in [t_1, t_2]$). t_1 is the transition time between the two stages. $k > 1, m > 1$ are parameters to decide the feature of the sharp turn, such as the tail shape, bending speed and maximum bending angle, *etc.*

A series of virtual circle functions can be now obtained from $(x_c(t), y_c(t))$. They are the tail postures to approximate using the tail linkages. The approximation procedure was illustrated in Section 3. An approximate example is displayed in Fig. 13. Fig. 14 shows a sequence of the robotic fish tail during C-shape turning.

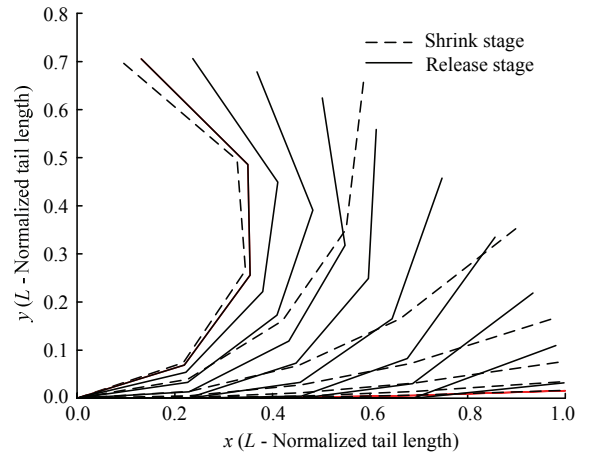


Fig. 14 The tail bending sequence in CST.

Note that the tail length is normalized to 1. Fig. 15 presents the joint angle q_{ij} against time. It is clear that there is a sharp increase in the shrink stage ($t < 1$), and a slower decrease in the release stage ($1 < t < 2.5$).

In practical CST control of robotic fish, the series of joint angles q_{ij} are regressed to a time-dependent function $q_j(t)$, where j is the index of joints. The whole motion of the CST is divided into three stages according to the joint angle calculation in Fig. 15: winding stage, peak-holding stage and unwinding stage. This division is similar to that in early of this section apart from the peak-holding stage, which is mainly designed for engineering purpose to protect servo motors from instant torque change. $q_j(t)$ can be regressed to following equations:

$$q_j(t) = \begin{cases} 0.5a_j(1 + \sin(\frac{\pi t}{T_0^j} - \frac{\pi}{2})) & 0 \leq t < T_0^j \\ 0.5a_j & T_0^j \leq t < T_1^j \\ 0.5a_j(1 + \sin(\frac{(t - T_1^j)\pi}{(T_2^j - T_1^j)} + \frac{\pi}{2})) & T_1^j \leq t \leq T_2^j \end{cases}, (17)$$

where T_i^j is the dividing time between the three stages, a_j is the maximum amplitude of each joint and K is the number of joints in the tail.

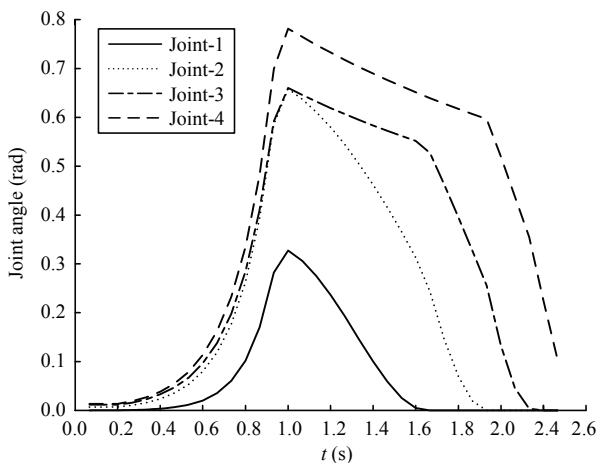


Fig. 15 The joint angle curve from the same sharp turn shown in Fig. 14.

5 Experimental results and analysis

5.1 Development of biomimetic robotic fish

In this section, each designed swim pattern is tested with our real robotic fish - G9 serial robotic fish as shown in Fig. 16. They are approximately 52 cm long, with three or four servo motors and two DC motors. The servo motors are concatenated together in the tail to act

as 3 joints; one DC motor is fixed in the head to change the Center of Gravity (COG) of the fish, while the other controls a micro-pump in order to adjust the weight of the robot by pumping water. The central administration and computation for each robotic fish is executed by a 400 MHz Gumstix computer^[27]. In detail, the Gumstix is employed to sample data from sensors, filter and process those data, perform decision-making, and produce control signals to several PIC micro-controllers, including three servo control boards, a DC motor control board, and a sensor board. The head of the robotic fish is waterproofed and rigid to counteract water pressure.

Each G9 fish is installed with over ten embedded sensors: a gyroscope, a pressure sensor, two position sensors, two electrical current sensors, a voltage meter, four infrared sensors, and an inclinometer. These embedded sensors enable the fish to detect: the depth in the water, the yaw/roll/pitch angles of its body, and the position of obstacles in front. The turning angle and electrical current consumption of each servo motor can be measured as well. In addition, Bluetooth and RS232 serial ports are available as means of communication with an external PC, which is used to command the Gumstix and PIC controllers. A sensor log can be retrieved via Bluetooth. With current settings, G9 robotic fish can be programmed to swim autonomously underwater in a 3D environment and no human intervention is necessary during their swimming.

On the top of the fish body, a dorsal fin is fixed vertically to avoid the fish wobble to side. Two pectoral fins are attached on two sides of the head to help balance the fish body. Both dorsal and pectoral fins are inactive. The propulsion force for the robotic fish swimming comes from an associated undulated movement of three joints, which make the fish body to push the surrounding water and obtain reaction force from the water.

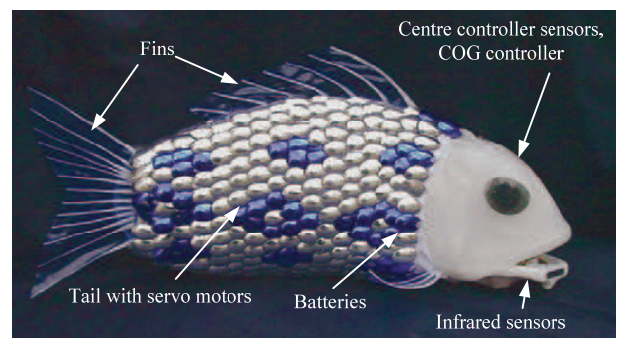


Fig. 16 The schematic structure of a G9 series robotic fish.

5.2 Cruise straight experiments

For the cruise straight swim pattern, the same kinematic parameters as in Fig. 9 were applied on G9 robotic fish apart from ω , which is 2.6π , i.e., the tail flapping frequency is 1.3 Hz which is an average flapping speed of adult carp cruising straight. The experiment tank is 5.5 m in length, 1.7 m in width and 1.7 m in height with 1.5 m water depth.

To compare previous methods with ours, we set up following three performance benchmarks:

- **Head swing amplitude** evaluates the head undulation of a robotic fish. It is calculated as the angle between two farthest head positions during swing. Its unit is degree. In Fig. 17, it is indicated as A_h .

- **Tail tip flapping amplitude** evaluates the tail flapping angle of a robotic fish. It is calculated as the angle between two farthest tail tip positions during swimming. Its unit is degree. In Fig. 17, it is indicated as A_t .

- **Swimming speed** evaluates how fast a robotic fish swims relative to its body length. Its unit is *body length per second*. It is calculated by dividing the real swimming speed over the body length of a G9 fish.

Table 1 shows the experimental results corresponding to three fish swimming modelling methods and one ideal swimming result which is measured from an adult carp. The details of three modelling methods are illustrated as follows.

- Method A doesn't consider the relative motion between fish tail and head, i.e. it skips the second module, $f_T(x, t)$ in our methodology (please see Fig. 2) and replaces $f_T(x, t)$ with $f_B(x, t)$ in the following calculation in Fig. 2. In addition, the Root Mean Square (RMS) error than the proposed error Function Eq. (2) is selected as the error function in digital approximation in Section 3. In another words, it doesn't take into account of the hydrodynamic mechanism of fish swimming. All other processes in Method A is the same as the method in this

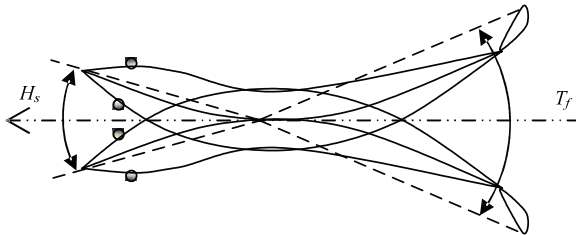


Fig. 17 Definition of head swing amplitude and tail tip flapping amplitude.

paper. This method is the common method in previous robotic fish research.

- Method B takes the same process as the proposed method in this paper, i.e. it converts the body motion function $f_B(x, t)$ to tail motion function $f_T(x, t)$ for modelling swimming. However, the error function of digital approximation in Section 3 is RMS like Method A.

- Method C is exactly the same method proposed in this paper.

Comparing the experimental results between Methods A and B, we can conclude that factors, A_h and A_t , of Method B are closer to the ideal swimming's than Method A. It means that our consideration of relative head-tail motion have taken effect. On the other hand, Method B had a higher swimming speed than Method A, since its smaller head swing amplitude decreased the forthcoming water drag, and the tail flapping with larger amplitude generated greater thrust force.

Compared with Method B, Method C decreased A_h and A_t approaching the ideal swimming results. Although the difference of A_h and A_t between Method B and C is small, the swimming speed in Method C is much higher than that in Method B since the error functions are different in two methods. In contrast to the RMS error function in Method B, the error function in Method C considered the added-mass hydrodynamic mechanism, and directed to a more realistic digital approximation results which are closer to real fish's.

Although Method C improved the performance of the robotic fish swimming, it is notable that the swimming speed of our robotic fish is still far from that of the real carp. The reason is that many of other parameters of our robotic fish, such as the mechanical structure and skin material, greatly affect the performance of the robot. These parameters have to be optimised in the future work.

Table 1 Experimental result for cruise straight swim pattern

	Ideal swimming	Method A	Method B	Method C
Head Swing amplitude A_h	33.4	43.9	36.5	34.8
Tail flapping amplitude A_t	31.3	24.5	32.8	32.1
Swimming speed S_b	2.00	0.62	0.83	1.02

Fig. 18 shows a video sequence of *cruise straight* swimming motion of a Green G9 robotic fish, which swims along one side wall of the tank. The tail flapping

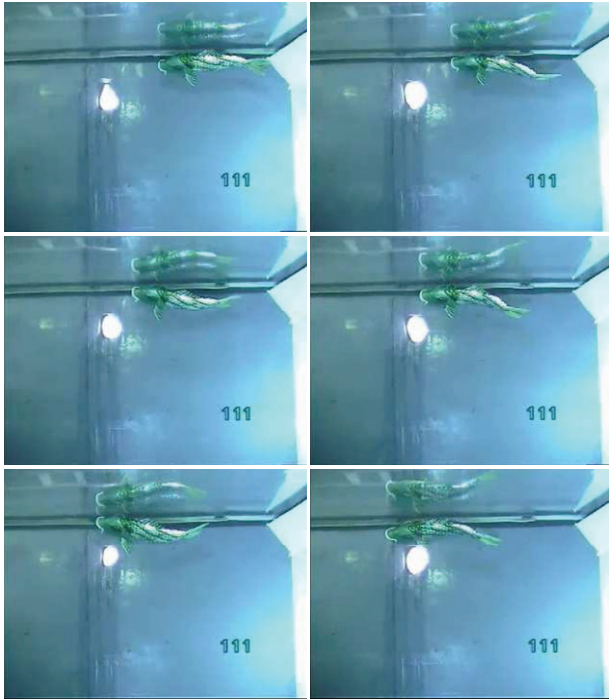


Fig. 18 A video sequence of Green G9 cruise straight motion. Note that the time step between two neighbour clips is 1s. The order of video sequence is from left to right and top to bottom.

frequency is set $f=0.5$ Hz, and it achieved a linear speed $V_p = 0.2$ m/s. This video clip shows how the *cruise straight* swim pattern of the robotic fish is smooth and similar to a real fish. Please refer to the video files associated with this paper.

5.3 CST experiments

For the CST swim pattern, Table 2 presents following parameters set in the calculation of Section 4.2. All parameters apart from t_1 and t_2 were determined by measuring a real carp performing CST corresponding to Fig. 12. t_1 and t_2 are the end time of the shrink stage and whole turning of CST respectively. The values of t_1 and t_2 for robotic fish were much higher than these for the real carp (where $t_1 = 0.03$, $t_2 = 0.1$) because the mechanics limitation of present servo motor used in our robotic fish tail cannot turn as quickly as a real carp. Therefore, a large value of t_1 or t_2 protects the servo motor from damage.

Table 2 Parameter setting for CST swim pattern

Parameters	Values
$cx_i, i = 0, 1, 2$	0.14, 0, 0.595
$cy_i, i = 0, 1, 2$	20, 0.245, 0.35
$t_i, i = 0, 1, 2$	0, 1, 2.5
a, b, k, m	0.3, 0.95, 3, 0.5

To see the effect of the proposed method on modelling CST, we set up following three performance benchmarks:

- **Turning angle**, T_a , is the angle changed in terms of the heading direction after performing CST. Its unit is degree.

- **Peak turning speed**, T_p , is the maximal turning speed during performing CST. Its unit is $\text{deg}\cdot\text{s}^{-1}$.

- **Turning radius**, T_r , evaluates how efficient the robotic fish turns relative to its body length. It is calculated by dividing the real turning radius by the body length of G9. Its unit is body length.

Table 3 shows the experimental results corresponding to two fish swimming modelling methods and one ideal swimming result which is measured from an adult carp performing CST corresponding to Fig. 12. The definitions of two modelling methods are the same as in the experiments of cruise straight. Method A was ignored here because CST was modelled starting from a tail motion function directly and it is unnecessary to consider the relative motion between head and tail.

Table 3 Experimental result for CST swim pattern

	Ideal swimming	Method B	Method C
Turning angle T_a	150	101	110
Peak turning speed T_p	4000	105	120
Turning radius T_r	0.20	0.31	0.30

Compared with Method B, Method C achieved a better performance but not much. The reason is that the error function used in Method C is proposed based on the add-mass hydrodynamics mechanism which is normally studied with the cruise straight rather than CST swimming pattern. However, both methods can approximate the ideal swimming regarding to turning angle, and their result are encouraging.

There exists still big difference on peak turning speed, T_p , between the real and robotic fish as seen in Table 3, which means the real fish can turn with a much faster speed. However, considering that the extended swimming duration for our experiments is $t_2 = 2.5$ s, which is greater than the swimming duration of the carp (where $t_2 = 0.1$ s) by 25 times, these experimental results are comparable as long as dividing the T_p of the ideal swimming by 25, i.e. $4000/25=160$.

The factor of Tuning radius has almost identical value between Method B and C. It means that the proposed error function has not much effect on the turning

radius.

Fig. 19 shows a video sequence of CST, which shows how CST helps the robotic fish to achieve sharp turn under water. It should be noticed that the time step between two neighbour clips is 0.5 s instead of 1s in Section 5.2. The maximum bending angle happens in the 5th clip, which is $\beta_c = 120$ deg.



Fig. 19 Video Sequence of blue G9 C-shape sharp turn, time step between two neighbour clips is 0.5 s. The order of video sequence is from left to right and from top to bottom.

6 Conclusion and future work

This paper presented a novel approach to modelling carangiform fish-like swimming motion for multi-joint robotic fish. It is an improved trajectory approximation method to solve the curve disagreement and large head swing problems. Firstly, our method takes the relative tail motion to the fish head into account when converting a body motion function to joint motion functions, which effectively solved disagreement problem in previous research work. Secondly, the hydrodynamic mechanism of fish swimming was considered when choosing an error function for the digital approximation method. The approximation results are more accurate than those computed without physics consideration. Thirdly, our method regressed a look-up table of joint angles into a time-independent function with limited parameters in

order to achieve online parameter optimization. We have implemented a policy gradient learning to optimize the parameters of CST^[24].

Two swim patterns were selected to illustrate the modelling procedure, namely *cruise straight* swim pattern that was modelled starting from a body motion function and was then converted to a tail motion function whereafter, and CST swim pattern that started from a tail motion directly. Both modelling work were tested in a real robotic fish, G9. The experimental results on the *cruise straight* swimming motion show that our methodology has made the robotic fish swim faster and follow the designed swimming function. In addition, the experimental results on the CST swimming motion indicate the proposed error function for the digital approximation achieved a better performance than the RMS error function. Our robotic fish operated in London Aquarium between October 2005 and June 2007, and their fish-like 3D swimming behaviours have impressed thousands of visitors worldwide.

In the future, the hydrodynamic swimming mechanism of CST should be studied to obtain better error functions to improve its performance further. The mechanical structure and skin material of our robotic fish should be further improved in order to improve their efficiency and robustness.

Acknowledgment

This research was financially supported by London Aquarium from 2003 to 2007. Our thanks also go to Ian Dukes, George Francis and Rob Knight for their excellent contribution toward the project.

References

- [1] Anderson J M, Kerrebrock P A. The vorticity control unmanned undersea vehicle (VCUUV): An autonomous robot tuna. *Proceedings of 10th International Symposium on Unmanned Untethered Submersible Technology*, Durham, NH, USA, 1997, 63–70.
- [2] Nato N. Control performance in the horizontal plane of a fish robot with mechanical pectoral fins. *IEEE Journal of Oceanic Engineering*, 2000, **25**, 121–129.
- [3] Witting J, Safak K, Adams G. Shape memory alloy actuators applied to biomimetic underwater robots. In: Ayers J, Davis J, Rudolph A (eds). *Neurotechnology for Biomimetic Robots*, MIT Press, USA, 2001, 117–136.
- [4] Liang J, Wang T, Wei H, Tao W. Researchful development

- of underwater robofish II: Development of a small experimental robofish. *Robot*, 2002, **24**, 234–238. (in Chinese)
- [5] Yu J, Wang S, Tan M. A simplified propulsive model of biomimetic robot fish and its realization. *Robotica*, 2005, **23**, 101–107.
- [6] Guo S, Fukuda T, Kato N, Oguro K. Development of underwater microprobe using IMPF actuator. *Proceedings of IEEE International Conference on Robotics and Automation*, Leuven, Belgium, 1998, 1829–1834.
- [7] National Maritime Research Institute. Welcome to Fish Robot Home Page, [2000-09-01], <http://www.nmri.go.jp/eng/khirata/fish/>
- [8] Mitsubishi Heavy Industries Ltd News. MHI to Produce First Lifelike Robotic Fish Available on Market, [2001-01-19], http://www.mhi.co.jp/en/news/sec1/e_0898.html
- [9] Nagrath S, Sequist L V, Maheswaran S, Bell D W, Irimia D, Streitlien K, Triantafyllou G S, Triantafyllou M S. Efficient foil propulsion through vortex control. *AIAA Journal*, 1996, **34**, 2315–2319.
- [10] Barrett D, Grosenbaugh M, Triantafyllou M. The optimal control of a flexible hull robotic undersea vehicle propelled by an oscillating foil. *Proceedings of the IEEE Symposium on Autonomous Underwater Vehicle Technology*, Monterey, CA, USA, 1996, 1–9.
- [11] Anderson J M. *Vorticity Control for Efficient Propulsion*, PhD Dissertation, Massachusetts Institute of Technology, 1996.
- [12] Morgansen K A, Duijndam V, Mason R J, Burdick J W, Murray R M. Nonlinear control methods for planar carangiform robot fish locomotion. *Proceeding of IEEE International Conference on Robotics and Automation*, 2001, **1**, 427–434.
- [13] Lauder G V, Drucker E G. Morphology and experimental hydrodynamics of fish fin control surfaces. *IEEE Journal of Oceanic Engineering*, 2004, **29**, 556–571.
- [14] Low K H, Willy A. Biomimetic motion planning of an undulating robotic fish fin. *Journal of Vibration and Control*, 2006, **12**, 1337–1359.
- [15] Mchenry M J, Pell C A, Long J H Jr. Mechanical control of swimming speed: Stiffness and axial wave form in undulating fish models. *The Journal of Experimental Biology*, 1995, **198**, 2293–2305.
- [16] Hu T, Shen L, Lin L, Xu H. Biological inspirations, kinematics modeling, mechanism design and experiments on an undulating robotic fin inspired by *Gymnarchus niloticus*. *Mechanism and Machine Theory*, 2008, **44**, 633–645.
- [17] Anderson J M, Streitlien K, Barrett D S, Triantafyllou M S. Oscillating foils of high propulsive efficiency. *Journal Fluid Mechanics*, 1998, **360**, 41–72.
- [18] Zhao W, Yu J, Fang Y, Wang L. Development of multi-mode biomimetic robotic fish based on central pattern generator. *Proceedings of International Conference of Intelligent Robots and Systems*, Beijing China, 2006, 3891–3896.
- [19] Yu J, Chen E, Wang S, Tan M. Research evolution and analysis of biomimetic robot fish. *Journal of Control Theory and Application*, 2002, **24**, 169–174.
- [20] Liu J, Dukes L, Knight R, Hu H. Development of fish-like swimming behaviours for an autonomous robotic fish. *Proceedings of the Control*, University of Bath, UK, 2004, 217.
- [21] Sfakiotakis M. Review of fish swimming modes for aquatic locomotion. *IEEE Journal of Oceanic Engineering*, 1999, **24**, 237–252.
- [22] Lighthill M J. Note on the swimming of slender fish. *Journal of Fluid Mechanics*, 1960, **9**, 305–317.
- [23] Triantafyllou M, Triantafyllou G. An efficient swimming machine, *Scientific American Magazine*, 1995, **272**, 40–46.
- [24] Liu J, Gu D, Hu H. Reinforcement learning for autonomous robotic fish. In: Nedjah N, dos Santos Coelho C L, de Macedo Mourelle L (eds). *Mobile Robots: The Evolutionary Approach*, Springer, Berlin/Heidelberg, Germany, 121–135.
- [25] Liu J, Hu H. Mimicry of sharp turning behaviours in a robotic fish. *Proceedings of IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 3329–3334.
- [26] Spierts I L Y, Leeuwen J L V. Kinematics and muscle dynamics of C- and S-starts of carp (*Cyprinus carpio* L.). *Journal of Experimental Biology*, 1999, **202**, 393–406.
- [27] Gumstix Co. Gumstix-Way Small Computing, [2009-09-24], <http://www.gumstix.com>