A Performance Optimization Strategy Based on Improved NSGA-II for a Flexible Robotic Fish*

Ben Lu[1], Jian Wang[1], Xiaocun Liao[1], Qianqian Zou[1], Min Tan[1], and Chao Zhou[1]

Abstract— The high speed and low energy cost are two conflicting objectives in the motion optimization of bio-inspired underwater robots, but playing a very important role. To this end, this paper proposes an optimization strategy for swimming speed and power cost using an improved NSGA-II for a flexible robotic fish. A dynamic model involving flexible deformation is established for speed prediction with the hydrodynamic parameters identified. A back propagation (BP) neural network is applied to perform compensation of power cost prediction with the dynamic model's prediction as input. In particular, an NSGA-II-AMS method is developed to improve the efficiency of solving the two-objective optimization problem based on NSGA-II. Finally, extensive simulations and experimental results demonstrate the effectiveness of the proposed optimization strategy, which offers promising prospects for the flexible robotic fish performing aquatic tasks with different performance constraints.

I. INTRODUCTION

Natural fishes are gifted with superb aquatic capabilities, such as high swimming speed and low energy cost. To mimic the fish mechanism, many studies focus on bionic mechanisms [1], dynamic modeling [2], and motion control [3]. However, there still remain many difficulties for robotic fish to achieve both high swimming speed and low energy cost, which the natural fishes are endowed with.

To imitate the physiological structure of fish, many researchers have adopted rigid and discrete designs. Yan *et al.* [4] developed a robotic carp driven by four servos, which achieved a speed of 0.58 body lengths per second (BL/s). Liu *et al.* [5] designed a robotic fish with three servos on the tail, which could reach 1.02 BL/s. The multi-motor design can intuitively simulate fish swimming, while it introduces lots of friction, resulting in large energy loss. In nature, fish can achieve high performance by using flexible muscles, which has attracted many interests to construct flexible robotic fish. Erturk *et al.* [6] developed a bimorph robotic fish, which achieved 0.3 BL/s. Liu *et al.* [7] developed a robotic fish with an undulating fin to control swimming motion, which achieved 0.703 BL/s. Flexible materials are ideal for continuous passive kinematics. More importantly, their periodic energy storage characteristics during deformation

are beneficial to high efficiency. Thus, the combination of a single motor and flexible materials endowed the robotic fish with both high swimming speed and low power cost [8]. With these flexible robotic fish proposed, performance optimization problems have attracted attention.

For the robotic fish, the morphology and control strategy directly affect the thrust and efficiency, which ultimately determine the speed and energy cost. Thus, it is worth studying to improve swimming performance through optimizing the mechanism and control method, which has been proved feasible. Clapham *et al.* [9] developed a robotic fish of which the speed was significantly optimized via full-body carangiform kinematics compared with the traditional undulatory motion. Yu *et al.* [10] adopted particle swarm optimization (PSO) algorithm to seek the controller parameters, which optimized speed for a multijoint robotic fish. Li *et al.* [11] analyzed the relationship between energy cost and control parameters like frequency and amplitude, then derived the control strategy for the lowest energy cost at different speeds.

While both impact the performance during a task such as execution time and cruise range, speed and power cost are two contradictory goals. Particularly, nondominated sorting genetic algorithm II (NSGA-II) [12] has been successfully applied to solve optimization problems with conflicting objectives. When using NSGA-II for performance optimization, many populations will be generated to solve the dynamic model, which consumes lots of time. In fact, limited by the manufacturing process and processor performance, the optimized parameters are usually discrete with limited resolution, which provides a basis for improving the execution efficiency of the optimization algorithm.

In this paper, we propose an optimization strategy to solve the two conflicting objectives of the speed and power cost optimization problem of a designed flexible robotic fish from the perspective of mechanism design and control strategy. The main contributions of this paper are as follows:

- An identified dynamic model considering the flexible deformation is established to conduct the speed prediction. Furthermore, a back propagation (BP) neural network for compensation in conjunction with the dynamic model is particularly applied to predict the power cost.
- An NSGA-II-AMS algorithm is designed to efficiently offer an optimization scheme of structure design and control strategy for the two conflicting objectives balancing. With the effectiveness validated by various conducted simulations and experiments, the proposed optimization strategy provides a valuable sight into the practical application of the flexible robotic fish.
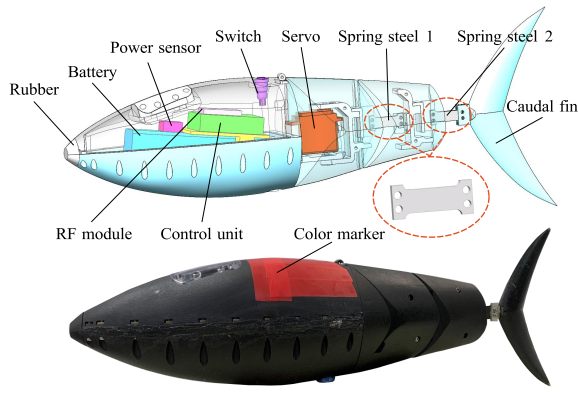
Fig. 1. Illustration of the flexible robotic fish. Up: side view of the mechanical structure. Down: side view of the developed prototype.

Fig. 2. Schematic illustration of coordinate frames and notations.

II. MECHATRONIC DESIGN

In nature, tuna are known as excellent aquatic creatures. Their streamlined body can reduce fluid resistance, and the stiff caudal fins help to generate large thrust [13]. To achieve the above characteristics, we propose a flexible robotic fish. In detail, a waterproof servo is used to drive the flexible fishtail. According to the mechatronic design, the prototype is implemented, with a length of 50.45 cm and a weight of 1.67 kg, as shown in Fig. 1.

As mentioned above, using flexible materials is critical for high efficiency. In addition, the multi-joint design is proved effective in reducing energy loss [14]. Thus, the "single-motor multi-joint" configuration is employed, which involves two spring steels installed behind a servo serving as the compliant joints to transmit torque to the rear fishtail when bent passively under hydrodynamic forces. 65MN is used to construct the spring steels, making the stiffness easy to modify by altering the thickness. With the compact and lightweight design philosophies, the moment of inertia of the fishtail is effectively reduced compared with the design of multiple motors in series, laying the foundation for high and efficient swimming performance.

III. PREDICTION OF SPEED AND POWER COST

A. Dynamic Modeling for Speed Prediction

To analyze the influence of flexible joints on swimming performance, a Lagrangian dynamic approach is adopted to model the underactuated flexible robotic fish [15].

Firstly, based on Pseudo-Rigid-Body Model (PRBM) theory [16], the compliant joint is considered as a series system where the resistance is modeled as a torsional spring at the midpoint and connected to rigid links, as shown in the lower right corner of Fig. 2. The total torque on the torsional spring can be written as $T = F(l_s/2 + L)$, where l_s and L are the lengths of the spring steel and posterior body, respectively. The spring constant K can be expressed as $K = EI/l_s$, where E and I are Young's modulus and the area moment of inertia of the spring steel, respectively.

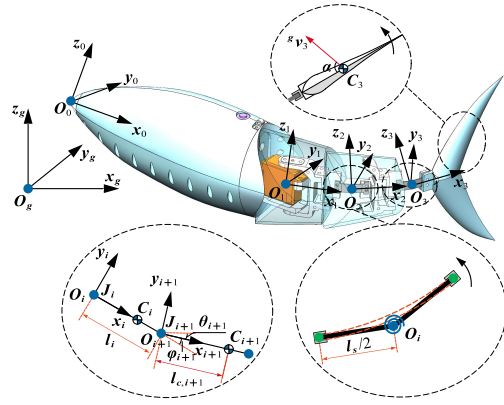Secondly, to analyze the kinematics, the coordinate frames are illustrated in Fig. 2, all of which follow the right-hand rule. The inertia frame and body-fixed frame are defined as $O_g x_g y_g z_g$ and $O_i x_i y_i z_i$ $(i = 0, 1, 2, 3)$, respectively. l_i is the length of ith link. φ_i $(i = 1, 2, 3)$ and θ_i $(i = 1, 2, 3)$ represent the angles between the ith link and its prior link and axis $o_g x_g$, respectively. The center of mass (COM) of the ith link is denoted as C_i, and $l_{c,i}$ indicate its distance relating to ith joint J_i. Consequently, the coordinate transformation and translational velocity can be obtained as:

$$
\begin{pmatrix} \boldsymbol{r}_i \\ 1 \end{pmatrix}_{4 \times 1} = \begin{pmatrix} {}^g R_i & {}^g P_i \\ \boldsymbol{0}_{1 \times 3} & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{c}_i \\ 1 \end{pmatrix}
$$
$$
{}^{i-1}P_i = \begin{pmatrix} l_{i-1} \\ 0 \\ 0 \end{pmatrix}, {}^g R_i = \begin{pmatrix} \cos\theta_i & -\sin\theta_i & 0 \\ \sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{pmatrix}
$$
$$
{}^g P_i = {}^g P_0 + \sum_{j=1}^{i} {}^g R_{j-1}{}^{j-1}P_j, (i = 1, 2, 3)
$$
$$
{}^g \boldsymbol{v}_i = {}^g \dot{R}_i \boldsymbol{c}_i + {}^g \dot{P}_i, {}^g \dot{R}_i = {}^g R_i \hat{\omega}_i
$$
$$
{}^g \omega_i = {}^g \omega_{i-1} + {}^g R_i \omega_i, \omega_i = (0, 0, \dot{\varphi}_i) \ (i = 1, 2, 3)
$$

(1)

where the position vectors of C_i and O_i in $O_g x_g y_g z_g$ are denoted as $\boldsymbol{r}_i$ and ${}^g P_i$, respectively. ${}^g \omega_i$ and ${}^g v_i$ represent the angular and translational velocity of C_i in $O_g x_g y_g z_g$, respectively. $\hat{\omega}_i$ indicate the skew-symmetric matrix of ω_i which is the angular velocity of C_i with respect to $O_i x_i y_i z_i$.

Moreover, the flexible robotic fish is perceived to swim freely only in $O_g x_g y_g$, and all forces are regarded as acting on the COM. Due to the different shapes of the anterior body and tail, different methods are applied to analyze the hydrodynamic forces. For the anterior body, the simplified Morison equation including drag force $F_{d,i}$ and added mass force $F_{a,i}$ [17] is employed, which can be obtained as $F_{d,i} = {}^g R_i {}^i F_{d,i}$ and $F_{a,i} = -m_{a,i} \ddot{\boldsymbol{r}}_i$, where:

$$
{}^i F_{d,i} = \begin{pmatrix} {}^i F_{dx,i} \\ {}^i F_{dy,i} \end{pmatrix} = \begin{pmatrix} -0.5 c_{f,i} \rho S_{x,i} {}^i v_{x,i} \left| {}^i v_{x,i} \right| \\ -0.5 c_{d,i} \rho S_{y,i} {}^i v_{y,i} \left| {}^i v_{y,i} \right| \end{pmatrix}
$$

(2)

$m_{a,i} = c_{ma,i} m_i$ is the additional mass, and $c_{ma,i}$ is the additional mass constant. $c_{d,i}$ and $c_{f,i}$ denote the dimensionless coefficients of drag and friction on the ith link, respectively. $S_{x,i}$ and $S_{y,i}$ indicate the corresponding characteristic cross-sectional areas of the ith link. ${}^i v_{x,i}$ and ${}^i v_{y,i}$ represent the corresponding translational velocities of C_i in $O_i x_i y_i$.

It should be noted that the drag force along the x-axis is only considered to act on the fish head, which means $c_{f,i} = 0 (i = 1, 2)$ in this paper. As for the lunate caudal fin, the quasi-steady wing theory involving lift force $F_{l,i}$ and drag force $F_{D,i}$ [18] are considered, which can be calculated as $F_{l,i} = 0.5\rho c_l S_i v_i^2$ and $F_{D,i} = 0.5\rho c_D S_i v_i^2$, where c_l and c_D are the lift and drag coefficients relating to the attack angle α, respectively. S_i is the wetted area.

At last, using the Lagrange–Euler equation, the dynamic model of the flexible robotic fish can be expressed as:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = Q_i \, (i = 1, ..., 5) \quad (3)$$

where L is the Lagrange equation $L(\boldsymbol{q}, \dot{\boldsymbol{q}}) = E_k(\boldsymbol{q}, \dot{\boldsymbol{q}}) - E_p(\boldsymbol{q})$. E_k and E_p denote the kinetic and potential energies, respectively. $\boldsymbol{q}$ represents the generalized coordinates vector, which is given as $\boldsymbol{q} = [x_1, y_1, \theta_1, \varphi_2, \varphi_3]^T$. In consideration of the flexible joints, it can be derived as $E_k = \sum_{i=0}^3 0.5 {}^g v_i^T M_i {}^g v_i + \sum_{i=0}^3 0.5 {}^g \omega_i^T I_i {}^g \omega_i$, $E_p = 0.5 \sum_{i=2}^3 K_i(\varphi_i)^2$, where M_i and I_i denote the mass and inertia matrix of the ith link, respectively. Q_i is the generalized force produced by the surrounding fluid, which can be derived as $(Q_1, Q_2)^T = \sum_{i=0}^2 F_{d,i} + \sum_{i=0}^2 F_{a,i} + F_{l,3} + F_{D,3}$ and $(Q_3, Q_4, Q_5)^T = \sum_{i=0}^2 \tau_{d,i} + \sum_{i=0}^2 \tau_{a,i} + \tau_{l,3} + \tau_{D,3}$, where $\tau_{a,i}$, $\tau_{d,i}$, $\tau_{l,3}$, and $\tau_{D,3}$ are the moments on joint J_i. Finally, by numerically solving (3), the motion state like the swimming speed of the robotic fish can be obtained.

B. Error Compensation for Power Cost Prediction

To fully analyze the power cost, the servo output torque τ can be obtained by using (3) where $q_i = \varphi_1$ and $Q_i = \tau$. Then the power cost can be estimated as $p_d = \frac{1}{T}\int_0^T \tau \dot{\varphi}_1 dt$, where T denote the swing period during steady swimming.

However, due to the complex and highly nonlinear friction properties [19] and inconsistent motor properties [20], it is difficult to accurately model the power cost. With the development of machine learning, the compensation method using neural networks has shown applicability [19], [21]. In this paper, a BP neural network is adopted for power compensation. Considering that power cost is mainly related to hydrodynamic forces and control parameters, the thicknesses of the spring steels (i.e., $thk_i, i = 1, 2$) and the swing frequency of the servo f are selected as the inputs. In addition, to make full use of the prior knowledge, the power prediction p_d of the dynamic model is introduced into the network. As a result, the input vector can be expressed as: $\boldsymbol{x} = [thk_1, thk_2, p_d, f]^T$, and the output is the final power estimation $y = p_f$. To choose the appropriate numbers of hidden layers and neurons, the k-fold cross-validation is used where $k = 10$ and the hidden layers are selected as 6 neurons for the first layer and 5 neurons for the second layer. Tanh is adopted as the activation function and all inputs are normalized.

IV. IMPROVED NSGA-II FOR PERFORMANCE OPTIMIZATION

In this paper, an improved NSGA-II with an adaptive memory space method, named NSGA-II-AMS, is proposed
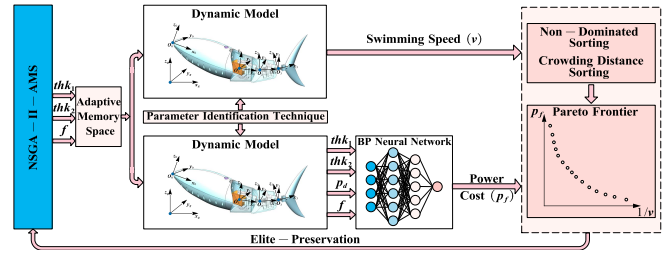

Fig. 3. Framework of the optimization procedure.

to achieve practical structure and control strategy design for performance optimization. The proposed optimization framework is shown in Fig. 3, and the details are described as follows, where $\boldsymbol{o} = [1/v, p_f]$ are the objectives, v denotes the speed, and $\boldsymbol{d} = [thk_1, thk_2, f]$ are the decision variables. Thus, the optimization problem can be formulated as:

$$\begin{cases} \min \quad 1/v, p_d \\ \text{s.t.} \quad thk_1, thk_2, f \text{ with certain ranges} \end{cases} \quad (4)$$

where the certain ranges are described below.

A. Fixed-resolution Binary Encoding

In practical applications, limited by the manufacturing process and processor performance, $[thk_1, thk_2, f]$ can only be achieved within a certain range. In this paper, the binary coding is used to represent thk_i and f, where the code length l can be calculated as $l \geq \log_2((U_{\max} - U_{\min})/\delta + 1)$ where $[U_{\min}, U_{\max}]$ denote the actual value ranges, which are $[0.2 \text{ mm}, 1.0 \text{ mm}]$ for $thk_i(i = 1, 2)$ and $[1.0 \text{ Hz}, 3.0 \text{ Hz}]$ for f, respectively, and δ represents the resolution, which are 0.1 mm for $thk_i(i = 1, 2)$ and 0.01 Hz for f, respectively. As a result, the code lengths of $thk_i(i = 1, 2)$ and f are 4 and 8, respectively. Due to the binary-coded representation, the single-point crossover with a probability of $p_c = 0.8$ and bitwise mutation with a probability of $p_m = 0.01$ are used in this paper.

B. Adaptive Memory Space Method

According to the general NSGA-II, the population members need to be used for fitness calculation in each iteration. Based on the above dynamic modeling, each swimming speed and power cost calculation involves complex numerical solving operations, making the time cost unignorable. Actually, due to the limited range and resolution of $\boldsymbol{d}$, there may be different members with the same $\boldsymbol{d}$ during each iteration, although crossover and mutation operations are performed. Thus, there will be some redundant computations in each iteration, which can be reduced.

In this paper, an adaptive memory space method is proposed to reduce the number of repeated calculations. Specifically, a memory matrix M_a is created to record the Pareto frontier (PF) from the last l iterations, which can be expressed as $M_a = [PF_k, \cdots, PF_{k-l+1}]$, where k denotes the k-th iteration. When updating $\boldsymbol{o}$ during each iteration, it should be conditionally judged whether the member has undergone crossover or mutation operation first, and those

Algorithm 1 Algorithm of NSGA-II-AMS

1: Initialize $m_{\max}$, $m_{\min}$, p, l, k, $k_{\max}$, p_c, p_m, M_a, $\boldsymbol{o}$, $\boldsymbol{d}$
2: Initialize the population and calculate the objectives
3: Perform non-dominated sorting (NDS) and calculate crowding distance (CD)
4: **repeat**
5:    Perform binary tournament to get parent population
6:    **repeat**
7:       Select two members randomly, and perform the single-point crossover based on p_c and bitwise mutation based on p_m
8:       **if** The parent is crossed or mutated, **then**
9:          **if** Offspring's $\boldsymbol{d}$ exits in M_a, **then**
10:          Update offspring's $\boldsymbol{o}$ with the values in M_a
11:          Accumulate the numbers of the update
12:          **else**
13:          Update $\boldsymbol{o}$ through calculation
14:       **else**
15:          No update for $\boldsymbol{o}$
16:    **until** Operation numbers reach p
17:    Perform NDS, CD sorting, and elite-preservation
18:    Update the size of the memory matrix by (5)
19:    Update M_a: rows 1 to $p \cdot Int(m/p)$ with the PF from k-th to $(k - Int(m/p) + 1)$-th iterations and the rest rows with the $Mod\,(m/p)$ members which are randomly sampled from the PF of the $(k - Int\,(m/p))$-th iteration.
20: **until** $k = k_{\max}$

which do not will skip the update. Then, query whether the offspring's $\boldsymbol{d}$ exists in M_a. If so, update offspring's $\boldsymbol{o}$ directly with the values in M_a, otherwise calculate $\boldsymbol{o}$. Finally, considering that the optimization results will gradually converge to the PF, the size of M_a can be adaptively modified to reduce the waste of storage space. In detail, the size of M_a is initialized to $m \times n$, where $m = p \cdot l$, p denotes the population size, and n represents the total number of members in $\boldsymbol{o}$ and $\boldsymbol{d}$. Afterward, m will be updated using an adaptive clipping method during each iteration as follows:

$$m = Int\,(m_{\max} - (m_{\max} - m_{\min})k/k_{\max}) \qquad (5)$$

where $m_{\max}$ and $m_{\min}$ are the maximum and minimum space capacity of m, respectively. $k_{\max}$ is the maximum number of iterations, and $Int\,(\cdot)$ is the round down function.

Based on the above description, to illustrate the NSGA-II-AMS more clearly, we present the algorithm flowchart as Algorithm 1, where $Mod\,(\cdot)$ represents the remainder function. Therefore, based on the dynamic model and the BP neural network, the PF can be efficiently obtained via the NSGA-II-AMS algorithm.

V. SIMULATIONS AND EXPERIMENTS

In this section, extensive simulations and experiments were conducted to demonstrate the effectiveness of the proposed optimization strategy. Specifically, the aquatic experiments were conducted in a water tank with a size of 5 m × 4 m × 1.2 m, with a global camera to record the trajectory
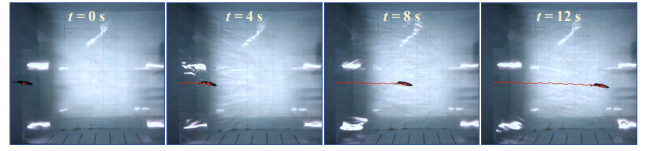

Fig. 4. Snapshot sequence of straight forward swimming motion with the configuration of $[thk_1 = 0.6$ mm, $thk_2 = 0.4$ mm, $f = 1.0$ Hz$]$.
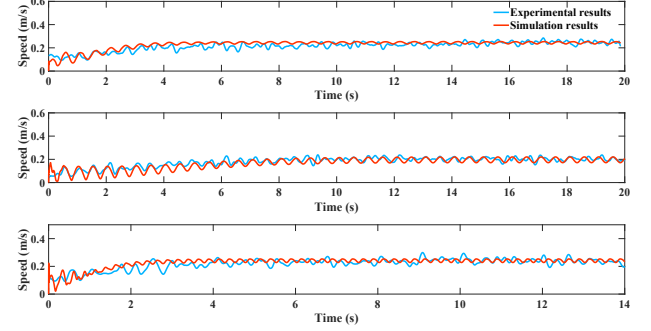

Fig. 5. Comparison of simulation results with the experimental results. Here, the configurations of $[thk_1, thk_2, f]$ are [0.4 mm, 0.2 mm, 1 Hz], [0.8 mm, 0.6 mm, 1 Hz] and [0.3 mm, 0.2 mm, 2 Hz] for the top, middle and bottom images, respectively.

and calculate the speed. Furthermore, a current and voltage sensor was adopted to measure both signals with a sampling rate of 100 Hz for more than 5 s. The current and voltage data were processed to calculate the average power cost of the servo. Limited by the manufacturing process and the capability of the servo, $thk_i (i = 1, 2)$ and f were confined to $[0.2$ mm, 1.0 mm$]$ with a step of 0.1 mm and $[1.0$ Hz, 3.0 Hz$]$ with a step of 0.5 Hz for all experiments, respectively. Fig. 4 depicts the snapshot sequence of straight forward swimming of the robotic fish during an experiment. Due to the tortuous trajectory, the speed is calculated as $v = s_{smo}/t_{total}$, where s_{smo} is the smoothed trajectory during time t_{total} using the moving average method. The servo control law adopted a general form, which could be expressed as $\varphi_1 = Amp \sin(\omega t)$, where φ_1 is the servo output, Amp denotes the output amplitude, which is set as 30° in this paper, and $\omega = 2\pi f$ represent the angular frequency.

A. Speed and Power Prediction Results

To effectively implement velocity prediction, appropriate hydrodynamic parameters $\boldsymbol{c_h}$ in the dynamic model need to be selected, which can be expressed as $\boldsymbol{c_h} = [c_{ma,0}, c_{f,0}, c_{d,0}, c_{ma,1}, c_{d,1}, c_{ma,2}, c_{d,2}, c_l, c_D]$. Since the hydrodynamic parameters directly determine the speed and are very difficult to measure due to the complex shape of the robotic fish, the nonlinear gray-box identification method [22] is utilized to identify the parameters. In this paper, extensive experiments with different configurations of $thk_i (i = 1, 2)$ and f were conducted. Moreover, 80% of the experimental data were selected as the training set, and the rest were selected as the validation set. Finally, with the acquired training set and the initial values based
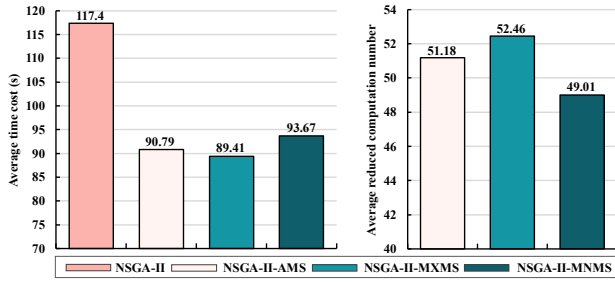
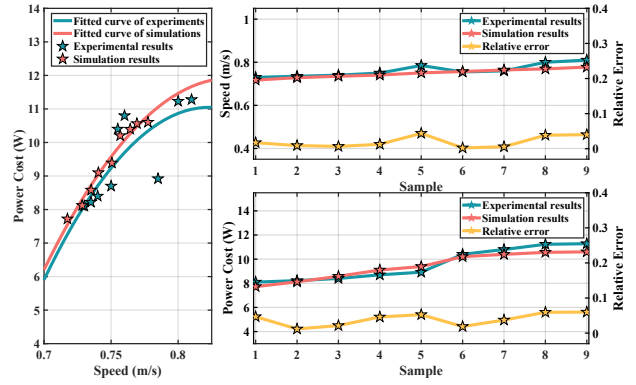Fig. 6. Algorithm performance comparison between NSGA-II, NSGA-II-AMS, NSGA-II-MXMS, and NSGA-II-MNMS.

Fig. 7. Simulations and experimental results. Here, the high swimming speed (i.e., above 0.7 m/s) regions in the optimization results with a total of 9 data are displayed.

on the empirical data [17], the hydrodynamic parameters were identified as $c_h = [0.01, 1.46, 0.01, 0.34, 4, 0.74, 4, 0.56, 0.17]$. The root-mean-square error (RMSE) between the experimental and simulation results in the validation set is calculated, yielding a value of 0.05 m/s. The comparison between simulations and experiments is also made, as shown in Fig. 5, which illustrates the identified dynamic model can predict the swimming speed well.

As mentioned above, the power cost can be predicted by adopting the compensation result from the BP network. In this paper, the total number of samples was 386, of which 90% were used for training and 10% for validation. The network was trained according to Levenberg-Marquardt optimization with maximum epochs of 10000, and the learning function adopted gradient descent with momentum, where the learning rate was 0.001, and the momentum constant was 0.1. The mean squared error (MSE) was used as the loss function. After training, the mean error on the validation set was 10.8%, showing good performance in the prediction of power cost.

B. Algorithms Performance Comparison

To verify the effectiveness of the proposed NSGA-II-AMS, four optimization algorithms were tested in this paper, which were denoted as NSGA-II, NSGA-II-MXMS, NSGA-II-MNMS, and NSGA-II-AMS, respectively. Specifically, NSGA-II-MXMS and NSGA-II-MNMS refer to the improved NSGA-II with the maximum and minimum memory space, respectively. Therefore, these two algorithms can be regarded as the basic versions of NSGA-II-AMS. All the four algorithms adopt the fixed-resolution binary encoding, $p = 100$ and $k_{max} = 500$, and those which use memory space utilize the same configurations: $[m_{max} = 300, m_{min} = 100]$.

These four algorithms were used to solve the above optimization problems, and the simulation results showed that the optimization results of these four algorithms were completely consistent, which could be understood that the proposed algorithms did not interfere with the population genetics, nor would they affect the final PF. To illustrate the execution efficiency, two indicators are discussed: the average time cost α and the average reduced computation number β, where α is the ratio of the total time cost of the algorithm to k_{max}, and β is the ratio of the total update numbers mentioned in line 11 of Algorithm 1 to k_{max}. The

performance comparisons of the four algorithms are shown in Fig. 6. It can be seen that the three improved algorithms can reduce α significantly. Specifically, compared with the general NSGA-II, the execution efficiencies of NSGA-II-AMS, NSGA-II-MXMS, and NSGA-II-MNMS are improved by 22.7%, 23.8%, and 20.2%, respectively. Correspondingly, the comparisons of β illustrate that the efficiency is improved because many repetitive computations are avoided in each iteration, and the more computations that are avoided, the more efficient the algorithm will be. The performance of NSGA-II-AMS is basically the same as that of NSGA-II-MXMS, and it is better than that of NSGA-II-MNMS. Moreover, the average memory storage space of NSGA-II-AMS is about 200 with the configurations: $[m_{max} = 300, m_{min} = 100]$, which is reduced by 33.3% compared to that of NSGA-II-MXMS, proving the effectiveness of the proposed algorithm in this paper.

C. NSGA-II-AMS Optimization Results and Experiments

To verify the validity of the NSGA-II-AMS optimization results (i.e., the PF), extensive experiments were carried out. Considering that high performance is currently the main concern of robotic fish, the high speed (i.e., above 0.7 m/s) regions in the PF are mainly discussed first. The PF with a total of 9 samples are shown in Fig. 7, which are represented in red, and the optimized thk_i and f of each sample are tabulated in Table I. It is observed that the speed and power cost basically meet the linear trend, which is mainly caused by the crowding distance sorting in NSGA-II-AMS. More interestingly, the increase of speed mainly depends on the increase of f, and the appropriate stiffness configuration can heighten the ceiling of speed at the same f.

In addition, the experiments which employed the configurations in Table I were conducted. The results are offered in Fig. 7, and the relative errors are calculated as $\varphi = abs\,(g_{sim} - g_{exp})/g_{exp}$ where g_{sim}, g_{exp}, and $abs\,(\cdot)$ are the simulation result, experimental result, and absolute value function, respectively. The small φ validate that the identified dynamic model and BP neural network perform well in the prediction of speed and power cost, with the RMSE of

Sap	thk_1/mm	thk_2/mm	f/Hz	Sap	thk_1/mm	thk_2/mm	f/Hz
1	1.0	0.4	2.81	6	1.0	0.5	2.97
2	1.0	0.4	2.84	7	1.0	0.5	3.00
3	1.0	0.4	2.88	8	0.7	0.5	2.95
4	1.0	0.4	2.95	9	0.7	0.5	2.97
5	1.0	0.4	3.00				

Note: "Sap" refers to the sample number for simulations and experiments.

Item		1	2	3
Configurations	Original	[0.9,0.4,3.0]	[0.9,0.7,3.0]	[0.6,0.6,3.0]
	Optimized	[1.0,0.4,2.81]	[1.0,0.4,2.84]	[0.7,0.5,2.95]
Performance	Original	[0.74,9.91]	[0.742,13.01]	[0.807,11.39]
	Optimized	[0.74,8.1]	[0.742,8.22]	[0.807,11.23]

Note: "Configurations" refers to $[thk_1(mm), thk_2(mm), f(Hz)]$, and "Performance" refers to [Speed (m/s), Power Cost (W)].
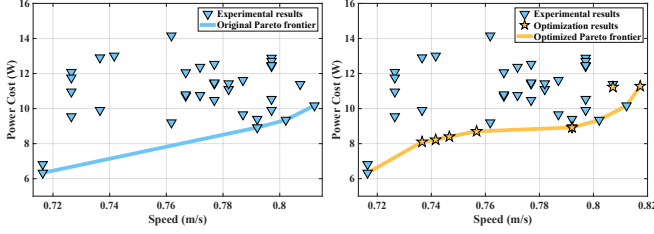

Fig. 8. Experimental and optimized results. Here, the high swimming speed (i.e., above 0.7 m/s) regions are displayed.

0.03 m/s and 0.43 W, respectively. Moreover, it should be noted that although we mainly focused on the performance at high speed, verification experiments were also conducted at speeds under 0.7 m/s, with the RMSE of 0.04 m/s and 0.24 W, respectively.

Furthermore, the superiority of the optimization results was also investigated, with the results shown in Fig. 8, where the experimental results are obtained from the conducted experiments with limited configurations as described above, and the optimization results are selected from the experimental results in Fig. 7 to form a PF. The original PF is formed from the experimental results, and the optimized PF is formed from both experimental and optimized results. Careful inspection shows that the optimized results describe the PF of the flexible robotic fish more accurately and thoroughly, though one of them is dominated by others due to the slight errors from the predictive model, and complement many areas not covered in the original PF, which is very important for the practical application. More importantly, the optimized results achieve better performance compared to the original ones, which are discussed with fixed speeds in Table II. Conclusions can be drawn that the proposed optimization strategy can help to explore configurations with better performance, with the power cost reduced by 18.3%, 36.8%, and 1.4%, respectively, further demonstrating the effectiveness of the proposed method in this paper.

D. Discussion

Due to the multiple degrees of freedom requirement and the unavoidable friction loss of the actuators, it is hard to achieve both high speed and low power cost for robotic fish. A flexible robotic fish is proposed to pursue both via adopting two spring steels as the flexible joints. Compared with [4]–[7], the robotic fish shows good performance (more than 1.6 BL/s) with low power cost. More importantly, a

speed and power cost prediction method is proposed and the two-objective optimization problem is solved by adopting an improved NSGA-II. Compared with [9]–[11], the proposed strategy can not only obtain the modification strategy of morphology and control method but balance the multiple conflicting optimization objectives. That is to say, we can simultaneously obtain the optimal design scheme for the minimum power cost and maximum swimming speed at a given swimming speed and power cost, respectively. Furthermore, with the proposed NSGA-II-AMS, the execution efficiency is improved by 22.7% compared with the general NSGA-II while considering saving memory space. With the obtained results, it is considered convenient to implement tasks with different performance constraints.

Nevertheless, there are some limitations. The flexible joints cannot achieve the real-time stiffness adjustment, and some discontinuities may appear in the obtained PF due to the discrete values of thk_i and f. Therefore, the variable stiffness design and resolution improvement of thk_i and f deserve further investigation, which are ongoing endeavors.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have developed an optimization strategy to solve the two conflicting objectives of speed and power cost optimization problems for a flexible robotic fish. Considering the flexible structure and control frequency, an NSGA-II-AMS algorithm is proposed to obtain the PF based on a validated performance prediction model. First, an identified dynamic model with full consideration of the flexible joints is used to predict the swimming speed. For reliable power cost prediction, a BP neural network is adopted to compensate the prediction results of the dynamic model. Moreover, to balance the conflicting objectives of speed and power cost, an improved NSGA-II with an adaptive memory space is offered to efficiently solve the optimization problem. Finally, extensive simulations and experiments of a designed flexible robotic fish have demonstrated the effectiveness of the optimization strategy, and the obtained results offer valuable insight into complex ocean tasks using flexible robotic fish.

Our future work will concentrate on solving the optimization problem with the consideration of more structural and control parameters such as the length of the spring steel and swing amplitude of the robotic fish. In addition, we will aim to develop a robotic fish with a variable-stiffness tail to achieve real-time swim speed and efficiency optimization.

REFERENCES

[1] S. Zhang, Y. Qian, P. Liao, and J. Yang, "Design and control of an agile robotic fish with integrative biomimetic mechanisms," *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 4, pp. 1846–1857, Aug. 2016.

[2] W. Wang, X. Dai, L. Li, B. H. Gheneti, Y. Ding, J. Yu, and G. Xie, "Three-dimensional modeling of a fin-actuated robotic fish with multimodal swimming," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 4, pp. 1641–1652, Aug. 2018.

[3] A. Keow, Z. Chen, and H. Bart-Smith, "PIDA control of buoyancy device enabled by water electrolysis," *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 3, pp. 1202–1210, Jun. 2020.

[4] Q. Yan, Z. Han, S. Zhang, and J. Yang, "Parametric research of experiments on a carangiform robotic fish," *J. Bionic Eng.*, vol. 5, no. 2, pp. 95–101, Jul. 2008.

[5] J. Liu, and H. Hu, "Biological inspiration: From carangiform fish to multijoint robotic fish," *J. Bionic Eng.*, vol. 7, no. 1, pp. 35–48, Apr. 2010.

[6] A. Erturk, "Macro-fiber composite actuated piezoelectric robotic fish," in *Robot Fish: Bio-Inspired Fishlike Underwater Robots*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 255—283.

[7] H. Liu, and O. Curet, "Swimming performance of a bio-inspired robotic vessel with undulating fin propulsion," *Bioinspir. Biomim.*, vol. 13, no. 5, Jul. 2018. Art. no. 056006.

[8] J. Zhu, C. White, D. K. Wainwright, V. Di Santo, G. V. Lauder, and G. V. Lauder, "Tuna robotics: A high-frequency experimental platform exploring the performance space of swimming fishes," *Sci. Robot.*, vol. 4, no. 34, Sep. 2018, Art. no. aax4615.

[9] R. J. Clapham, and H. Hu, "iSplash-I: High performance swimming motion of a carangiform robotic fish with full-body coordination," in *Proc. IEEE Int. Conf. Robot. Autom.,* Hong Kong, China, 2014, pp. 322–327.

[10] J. Yu, Z. Wu, M. Wang, and M. Tan, "CPG Network Optimization for a Biomimetic Robotic Fish via PSO," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 9, pp. 1962–1968, Sept. 2016.

[11] G. Li, G. Liu, Y. Li, and S. Chen, "On Optimal Energy Consumption Control Method for Tail Fin Bionic Robotic Fish," in *Proc. Int. Conf. Mach. Learn. Cybern.,* Kobe, Japan, 2019, pp. 1–6.

[12] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002, doi: 10.1109/4235.996017.

[13] S. Du, Z. Wu, J. Wang, S. Qi, and J. Yu, "Design and control of a two-motor-actuated tuna-inspired robot system," *IEEE Trans. Syst. Man Cybern. -Syst.*, vol. 51, no. 8, pp. 4670–4680, Aug. 2021.

[14] C. White, G. V. Lauder, and H. Bart-Smith, "Tunabot Flex: a tuna-inspired robot with body flexibility improves high-performance swimming," *Bioinspir. Biomim.*, vol. 16, no. 2, Mar. 2021. Art. no. 026019.

[15] D. Korkmaz, Z. H. Akpolat, S. Soygüder, and H. Alli, "Dynamic simulation model of a biomimetic robotic fish with multi-joint propulsion mechanism," *Trans. Inst. Meas. Control*, vol. 37, no. 5, pp. 684–695, Jan. 2015.

[16] L. L. Howell, *Compliant mechanisms*, New York, NY, USA: Wiley, 2013.

[17] V. Kopman, J. Laut, F. Acquaviva, A. Rizzo, and M. Porfiri, "Dynamic modeling of a robotic fish propelled by a compliant tail," *IEEE J. Ocean. Eng.*, vol. 40, no. 1, pp. 209–221, Jan. 2015.

[18] K. A. Morgansen, B. I. Triplett, and D. J. Klein, "Geometric methods for modeling and control of free-swimming fin-actuated underwater vehicles," *IEEE Trans. Robot.*, vol. 23, no. 6, pp. 1184–1199, Dec. 2007.

[19] L. Qiu, Y. Zhao, and Y. Zhang, "Adaptive friction identification and compensation based on RBF neural network for the linear inverted pendulum," in *Proc. Int. Conf. Electron. Mech. Eng. Inf. Technol.,* Harbin, China, 2011, pp. 385–388.

[20] J. Yu, S. Chen, Z. Wu, X. Chen, and M. Wang, "Energy Analysis of a CPG-controlled Miniature Robotic Fish," *J. Bionic Eng.*, vol. 15, no. 2, pp. 260–269, Mar. 2018.

[21] V. Vitiello, and A. Tornambe, "Adaptive compensation of modeled friction using a RBF neural network approximation," in *Proc. IEEE Control Syst.,* New Orleans, LA, United states, 2007, pp. 4699–4704.

[22] D. Chen, Z. Wu, H. Dong, M. Tan, and J. Yu, "Exploration of swimming performance for a biomimetic multi-joint robotic fish with a compliant passive joint," *Bioinspir. Biomim.*, vol. 16, no. 2, Dec. 2020. Art. no. 026007.