Energy Consumption Optimization for the Formation of Multiple Robotic Fishes Using Particle Swarm Optimization

Authors:

Dong Xu, Luo Yu, Zhiyu Lv, Jiahuang Zhang, Di Fan, Wei Dai

Abstract:

The traditional leader-follower formation algorithm can realize the formation of multiply robotic fishes, but fails to consider the energy consumption during the formation. In this paper, the energy optimized leader-follower formation algorithm has been investigated to solve this problem. Considering that the acceleration of robotic fish is tightly linked to the motion state and energy consumption, we optimize the corresponding control parameters of the acceleration to reduce energy consumption during the formation via particle swarm algorithm. The whole process has been presented as follows: firstly we realize the formation on the base of the kinematic model with leader-follower formation algorithm; then the energy consumption on the base of dynamical model are derived; finally we seek the optimal control parameters based on the particle swarm optimization (PSO) algorithm. The dynamics simulation of the energy optimization scheme is conducted to verify the functionality of the proposed energy optimized leader-follower formation algorithm via MATLAB. The optimized results demonstrate that the proposed approach, reducing energy consumption during the formation, is superior to the traditional leader-follower formation algorithm and can reduce energy consumption during the formation. The novelty of the work is that we can reduce the energy consumption during the process of formation by considering the energy consumption, which is a gap in the current research field.

Energy Consumption Optimization for the Formation of Multiple Robotic Fishes Using Particle Swarm Optimization

Dong Xu [1], Luo Yu [2], Zhiyu Lv [1], Jiahuang Zhang [1], Di Fan [3] and Wei Dai [4,* ID**

[1]   School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China;
     xd@buaa.edu.cn (D.X.); zy1603145@buaa.edu.cn (Z.L.); zhang-jiahuang@outlook.com (J.Z.)
[2]   School of Mechanical Engineering and Automation , Beihang University, Beijing 100191, China;
     14071115@buaa.edu.cn
[3]   Department of Computer Science, USC Viterbi School of Engineering, University of Southern California,
     Los Angeles, CA 90089, USA; difan@usc.edu
[4]   School of Reliability and Systems Engineering, Beihang University, Beijing 100191, China
*   Correspondence: dw@buaa.edu.cn; Tel: +86-138-1058-4286

Abstract: The traditional leader-follower formation algorithm can realize the formation of multiply robotic fishes, but fails to consider the energy consumption during the formation. In this paper, the energy optimized leader-follower formation algorithm has been investigated to solve this problem. Considering that the acceleration of robotic fish is tightly linked to the motion state and energy consumption, we optimize the corresponding control parameters of the acceleration to reduce energy consumption during the formation via particle swarm algorithm. The whole process has been presented as follows: firstly we realize the formation on the base of the kinematic model with leader-follower formation algorithm; then the energy consumption on the base of dynamical model are derived; finally we seek the optimal control parameters based on the particle swarm optimization (PSO) algorithm. The dynamics simulation of the energy optimization scheme is conducted to verify the functionality of the proposed energy optimized leader-follower formation algorithm via MATLAB. The optimized results demonstrate that the proposed approach, reducing energy consumption during the formation, is superior to the traditional leader-follower formation algorithm and can reduce energy consumption during the formation. The novelty of the work is that we can reduce the energy consumption during the process of formation by considering the energy consumption, which is a gap in the current research field.

Keywords: robotic fish; leader-follower formation flocking; energy consumption; parameter optimization

1. Introduction

Robotic fish, defined as a fish-like aquatic vehicle, has the highly controllable fins, the large aspect ratio, the undulatory body motions and lunate tail [1]. Robotic fish can be applied in lots of fields, such as underwater exploration, aquatic monitoring, patrol, and mobile sensing [2,3]. Most of the robotic fishes consume energy that are supplied by batteries. But batteries have limited capacity. When robotic fishes conduct different assignments, they need to take on different formation and change the formation frequently. However, during the process of changing the formation, they will consume lots of energy. Therefore, how to reduce the energy consumption in the process of formation and improve the efficiency of formation , have the very high research value.

Migrating flocking, such as traditional leader-follower flocking, can be used to address the formation problems of multiple robotic fishes systems. In terms of formation flocking problems of

multiple robotic fishes systems, some researches have come up with lots of methods. If some leaders have global knowledge and they lead other robotic fishes swim in a desired trajectory, the system can track a specific trajectory and realize the formation [4]. But the routine is fixed and not suitable for the real life. Later, for multi-agent dynamical systems, the distributed leader-follower algorithm has been set up [5]. Under the lead of this algorithm, robotic fishes have time-dependent velocities and the routine is not fixed. However, the leader is virtual and few formation articles have contained real leaders. Based on the consensus algorithm [4] and artificial potential field method [6], a model of distributed adaptive formation of multi-agent system has been put forward [7]. The leader is a genuine fish and has bounded unknown input. The leader can lead the followers realize the formation. Considering essential features of formation behavior, such as collision avoidance, and velocity matching [8], a distributed cohesive formation algorithm has been provided to realize the formation [9]. In this algorithm, robotic fishes are divided to two parts: one real leader and several followers. However, those formation algorithms above just realize the formation and fail to consider the energy consumption during the process of formation.

In order to predict, monitor and increase the energy of robotic fish, many researches have been conducted. On the basis of the ionic polymer-metal composite (IPMC) physical properties, in order to save energy, an energy consumption model has been set up to predict single robotic fish energy consumption [10]. A real-time energy monitoring and management system has been designed to monitor the energy consumption of the single robot fish [11]. A reversible energy conversion mechanism has been proposed, which can convert wave energy to electricity for single robotic fish [12]. Besides, an energy-saving cluster head method with fuzzy logic can be utilized to extend the single robotic fish lifetime [13]. Although those methods above can be utilized to predict, monitor and save the energy consumption of robotic fish, they are just devised for single robotic fish, rather than total energy of all the robotic fishes during the process of changing the formation. Although previous methods in literatures have been proposed to realize the formation construction of the group and decline energy consumption of single robotic fish, they involve few information about energy consumption about all the robotic fishes during the process of formation. In the process of swimming, the velocity of each robotic fish is different, and thus the energy consumption of each robotic fish differs from each other. In an attempt to reduce the energy consumption of multi-agent systems, the condition of each robotic fish should be researched in the global point of view.

The paper aims at providing a energy optimized leader-follower formation algorithm to minimize the energy consumption of multi-agent systems during the process of changing formation. Holding that the interconnection between a genuine leader and follower is unidirectional [14], and the leader does not have external control input, thus the formation is stable. Control parameters are utilized to express the acceleration equation of each robotic fish. With the different value of control parameters, the acceleration equation of each robotic fish will be different. As a result, the energy consumption will be different. Compared to the traditional distributed leader-follower formation algorithm, this paper adds new control parameters to the acceleration equation and utilizes particle swarm optimization algorithm to optimize the control parameters. Therefore, the energy consumption will be declined. The main contributions of this paper are as below: to the best of authors' knowledge, this is the first paper to discuss how to reduce the energy consumption of multi-agent system during the process of formation and propose an energy optimized leader-follower formation algorithm. Firstly, with an accurate kinematic model, we can achieve the formation and the acceleration equation including control parameters. Then, a dynamical model of the robotic fish is built to analyze the energy consumption of total robotic fish. The total energy consumption of the multi-agent systems is regarded as a function to guide the choice of control parameters. Then, under the lead of particle swarm algorithm, the control parameters of the acceleration equation can be optimized to lower total energy consumption during the process of formation. Finally, tests are performed to validate the proposed method in three kinds of formation.

The rest of this paper is organized as follows. Section 2 gives the kinematic model and dynamical model of the robotic fish, and proposes the energy optimized leader-follower formation algorithm to reduce the energy consumption of the multiple robotic fish system. And stability of the algorithm is verified. Simulation results are provided in Section 3. Section 4 provides discussion about the advantages of the algorithm and compares it with the traditional leader-follower algorithm. At the end, Section 5 renders several concluding remarks.

2. Materials and Methods

2.1. Methods of Robotic Fish

2.1.1. The Kinematic Model

When robotic fish makes the formation, we need to record the state of fish. Figure 1 shows the graphical representation of the kinematic model which helps record the process of the formation. As the Figure 1 shows, M_k is regarded as the mass center while D_k is regarded as the geometrical center. $p_k(t) = [x_k(t), y_k(t)]^T \in R^2$ is the position vector of robotic fish k at time t, $\theta_k(t) \in [0, 2\pi)$ is the heading angle and is measured from the x-axis in the anticlockwise rotation , $q_k(t) = [v_k(t), l_k\omega_k(t)]^T = [v_k(t), u_k(t)]^T$ is the velocity vector, $v_k(t) \in R$ is the forward speed, $\omega_k(t) \in R$ is the rotational speed, $u_k(t) = l_k\omega_k(t)$ is the tangential speed, and l_k is the distance between D_k and M_k. Assume that the size of the robotic fish is the same. Every value of the l_k is same and we can describe that $l_k = l_d(k = 1, \ldots, N)$, where l_d is a positive constant. Besides, $\overrightarrow{t_k}(t)$ and $\overrightarrow{n_k}(t)$ are two unit vectors orthogonal to each other, and $\overrightarrow{t_k}(t) = [\cos\theta_k(t) \quad \sin\theta_k(t)]^T$, while $\overrightarrow{n_k}(t) = [\sin\theta_k(t) \quad \cos\theta_k(t)]^T$.
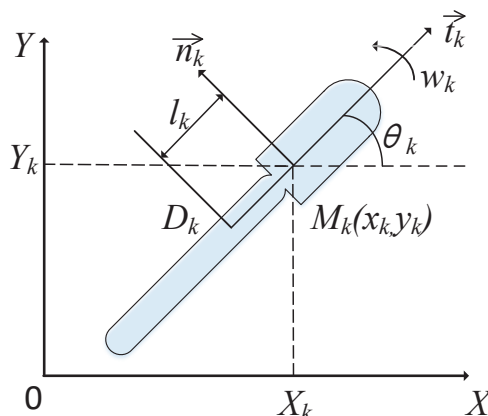

Figure 1. Kinematic model for the robotic fish k.

Kinematic model is adopted to describe the state of the robotic fish when realizing the formation [9]. Considering the fact that underwater communication is limited and technology for localization is not perfect, it is restricted to realize the coordination algorithms on robotic fishes. Despite that three-dimensional swimming capability about robotic fish has been researched [15], robotic fish is still modeled as unicycles swimming in a two-dimensional Euclidean space [16,17]. Because the robotic fish relies on the latter part of its body to provide force, kinematic model of fish is extended, whose mass center and geometrical center are not the same. To sum up, the kinematic model of the robotic fish k ($k = 1, \ldots, N$) is as follows:

$$\dot{x}_k(t) = v_k(t)\cos\theta_k(t) - \omega_k(t)l_k\sin\theta_k(t)$$
$$\dot{y}_k(t) = v_k(t)\sin\theta_k(t) + \omega_k(t)l_k\cos\theta_k(t)$$
$$\dot{\theta}_k(t) = \omega_k(t) \tag{1}$$
$$\dot{v}_k(t) = a_k(t)$$
$$\dot{\omega}_k(t) = b_k(t)/l_k$$

where: $a_k(t) \in R$ and $b_k(t) \in R$ are the thrusting and rotational acceleration, respectively. If $U_k(t) = [a_k(t) \quad b_k(t)]$ and

$$H_k(t) = \left[\begin{array}{cc} \cos\theta_k(t) & \sin\theta_k(t) \\ -\sin\theta_k(t) & \cos\theta_k(t) \end{array} \right]$$

the kinematic model can be simplified:

$$\dot{p}_k(t) = H_k(t)^T q_k(t)$$
$$\dot{q}_k(t) = U_k(t) \tag{2}$$

2.1.2. The Dynamical Model

When robotic fishes swim in the water, they will consume much energy because they need to overcome some force. In order to analyze the resistance force, we need to create the dynamic model of robotic fish. Figure 2 shows the graphical representation of the dynamic model. On the basis of the dynamic model, the resistance force of the robotic fish can be analyzed. Then, the function W about energy consumption can be calculated. Robotic fish is regarded as an integral rigid body, at the same time, the body and tail of robotic fish can be seen as a rigid flat because of the oscillation [18]. Making use of hydrodynamic pressure equation to analyze the hydrodynamic is a good way. The hydrodynamic can be seen as the concentrated force. F_0 and F_1 are the hydrodynamic forces on the head and body of the robotic fish, respectively [19,20].
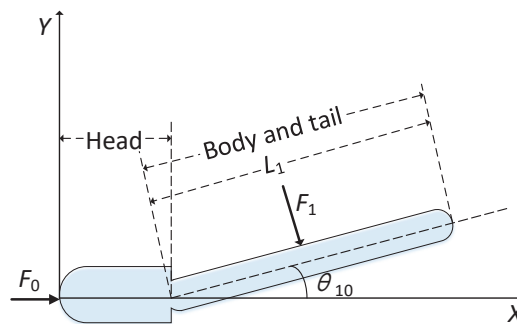

Figure 2. Dynamic model for the robotic fish k.

By neglecting the force of friction, the resistance force to the head of the robotic fish is:

$$F_0 = \frac{1}{2}\rho D_0 V_0^2 A_0 \tag{3}$$

where ρ is the density of the fluid, D_0 is the resistance coefficient of the head, V_0 is the velocity of the robotic fish and A_0 is the maximum sectional area of the head.

By regarding the body as a flat, the hydrodynamic force to the body is:

$$F_1 = \frac{1}{2}\rho D_1 V_{c1}^2 A_1 \tag{4}$$

where D_1 is the resistance coefficient of the body. X_0 is the $x-$coordinate of the head. A_1 is the sectional area of the body. V_{c1} is the velocity perpendicular to the body. The vector of the velocity is $\overrightarrow{V_{c1}} = (\dot{X}_0 - L_1 sin\theta_{10}\dot{\theta}_{10}/2, L_1 cos\theta_{10}\dot{\theta}_{10}/2)$. The normal vector of the body is: $\overrightarrow{I_1} = (sin\theta_{10}, -cos\theta_{10})$. And the velocity is: $V_{c1} = \overrightarrow{V_{c1}}\,\overrightarrow{I_1}$.

Because the swing of the joints generate the hydrodynamic force to overcome the resistance of the water, the total energy consumed in one period is:

$$W_{total} = \int_0^T (F_0(t)V(t))dt + \int_0^T (F_1(t)\omega(t)l_d)dt \tag{5}$$

The total energy consumption of all robotic fishes can be expressed and calculated. Let the total energy be a function and the particle swarm algorithms is utilized to optimize the value of the function.

2.1.3. The Energy Consumption of the Formation

Figure 3 gives the graphical representation of energy consumption during the process of changing formation. In the initial stage of the experiment, robotic fishes are in a state of disorder. When the leader-follower formation algorithm performs, each robotic fish swims to the appropriate place and take on expected formation. During this process, all robotic fish will consume the energy. Each follower possesses respective input control parameter in the acceleration equation based on the leader-follower formation algorithm. The change of the control parameter will lead to the change of acceleration equation. So is the profile of each robotic fish. Therefore, the formation process will alter and the energy consumption will undulate. We can get the each proper control parameter and reduce the total energy consumption of the formation process. The solid line represents energy consumption after the optimization, while the dotted line represents energy consumption before optimization. Particle Swarm Optimization algorithm can be used to get the optimal value of control parameter, and the experimental results are verified in the simulation experiment.
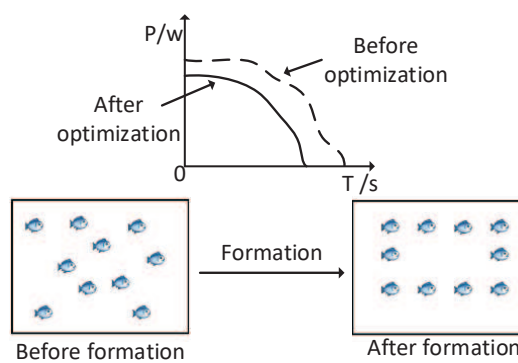
Figure 3. The energy consumption in the process of formation.

In order to illustrate the main work, we can see the Figure 4. A group of chaotic robotic fishes are divided into two parts: one leader and several followers. The leader takes the uniform motion and the followers take the variable motion. In the acceleration equation of followers, we add the control parameters and use the particle swarm optimization algorithm to optimize the control parameters. Finally, we can reduce the energy consumption. The energy optimized leader-followers formation algorithm will be explained in the next text.

Figure 4. Flowchart for proposed algorithm.

2.2. Energy Optimized Leader-Follower Formation Algorithm

2.2.1. The Leader-Follower Formation and Acceleration Equation
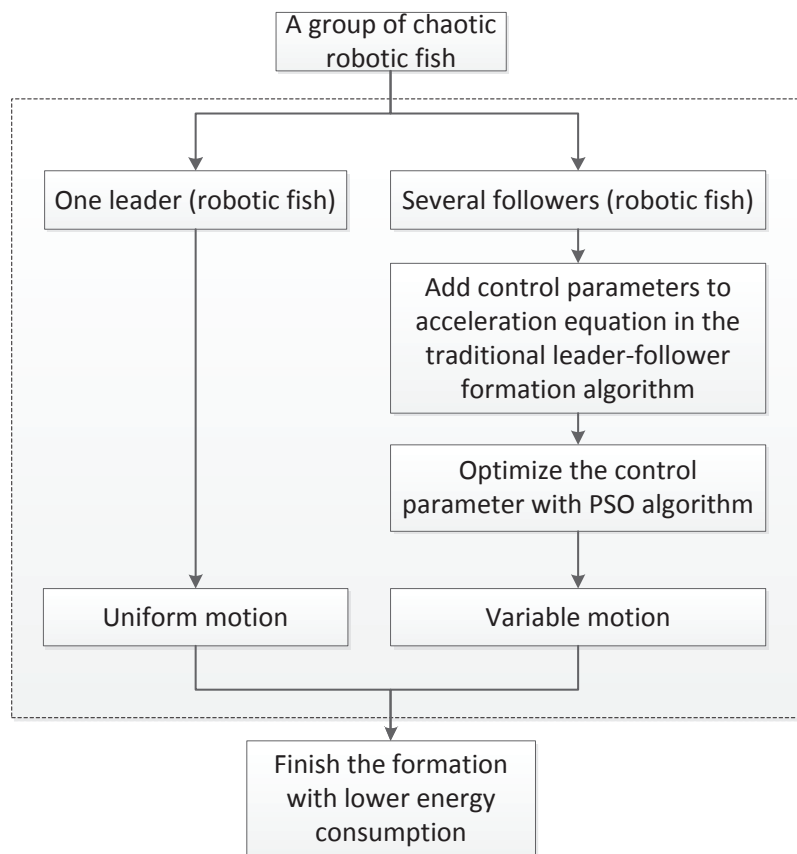
The leader has external control input and is set $L = \{1\}$ while the follower does not have and is set $F = \{2, \ldots, N\}$. Supposing that for a follower, interconnection with the follower is bidirectional, while interconnection with the leader is unidirectional [14]. $N_k(t)$ represents the neighbor set of follower $k \in F$ at time t, and is defined initially as $N_k(0) = \{n = \mid \parallel p_k(0) - q_n(0) \parallel < D, n = 1, \ldots, N, n \not\equiv k\}$. $D > 0$ is a constant, and $\parallel \bullet \parallel$ is the Euclidean norm. Consider the leader $k(k = 1)$ swims at a constant speed, the thrusting acceleration $a_1(t)$ and rotational acceleration $b_1(t)$ are equal to zero.

The control input of the follower $k(k = 2, \ldots, N)$ can be designed by:

$$
\begin{aligned}
a_k(t) = &- \sum_{n \in N_k(t)} c_k(\hat{v}_k(t) - \hat{v}_n(t)) \\
&- \sum_{n \in N_k(t)} c_k \dot{\hat{p}}_{kn}(t)^T \overrightarrow{t_k}(t) \\
&- \sum_{n \in N_k(t)} \nabla_{\hat{p}_{kn}(t)} V(\parallel \hat{p}_{kn}(t) \parallel)^T \overrightarrow{t_k}(t)
\end{aligned}
\tag{6}
$$

$$
\begin{aligned}
b_k(t) = - &\sum_{n \in N_k(t)} c_k l_d(\hat{\theta}_k(t) - \hat{\theta}_n(t)) \\
- &\sum_{n \in N_k(t)} c_k l_d(\hat{\omega}_k(t) - \hat{\omega}_n(t)) \\
- &\sum_{n \in N_k(t)} c_k \dot{\hat{p}}_{kn}(t)^T \overrightarrow{n_k}(t) \\
- &\sum_{n \in N_k(t)} \nabla_{\hat{p}_{kn}(t)} V(\parallel \hat{p}_{kn}(t) \parallel)^T \overrightarrow{n_k}(t)
\end{aligned}
\tag{7}
$$

where c_k is the control parameter. $\hat{\omega}_k(t) = \omega_k(t) - a_{k1}(t)\omega_1(t)$, $\hat{v}_k(t) = v_k(t) - a_{k1}(t)v_1(t)$, $\hat{\theta}_k(t) = \theta_k(t) - a_{k1}(t)\theta_1(t)$, $\hat{q}_n(t) = q_n(t) - a_{k1}q_1(t)$, $\hat{p}_{kn}(t) = \hat{p}_k(t) - \hat{p}_n(t)$, c_k is a positive constant varying from follower to follower. Consider $p_k^*(t) = \int_0^t \omega_{k1}(t) H_k^T(t) q_1 d\tau$, $\dot{\hat{p}}_k(t) = \dot{p}_k(t) - \dot{p}_k^*(t) = H_k(t)^T q_k(t) - a_{k1} H_k(t)^T q_1(t) = H_k(t)^T \hat{q}_k(t)$. The acceleration equation is consisted of two parts.

The specific potential function is [7]:

$$
V = \sum_{k \in F} \sum_{n \in N_k} V(\parallel \hat{p}_{kn} \parallel) + \sum_{n \in N_k} V(\parallel \hat{p}_{1n} \parallel)
\tag{8}
$$

Definition 1. *Definition 1: $V(\parallel \hat{p}_{kn} \parallel)$ is a potential function of the Euclidean norm $\parallel \hat{p}_{kn} \parallel$ between agent k and n. This potential function is radially unbounded, nonnegative, differentiable, such that:*

1.　*When $\parallel \hat{p}_{kn} \parallel \to 0$, $V(\parallel \hat{p}_{kn} \parallel) \to \infty$;*
2.　*When $\parallel \hat{p}_{kn} \parallel \to 2D$, $V(\parallel \hat{p}_{kn} \parallel) \to \infty$;*
3.　*When $\parallel \hat{p}_{kn} \parallel$ is equal to a certain value between 0 and 2D, $V(\parallel \hat{p}_{kn} \parallel)$ is located in its unique minimum.*

When potential $V(\parallel \hat{p}_{kn} \parallel)$ reaches its unique minimum, the formation can be completed. In order to verify the existence of potential $V(\parallel \hat{p}_{kn} \parallel)$. A potential function can be made [9]. Specific potential $V(\parallel \hat{p}_{kn} \parallel)$ is:

$$
V(\parallel \hat{p}_{kn} \parallel) = \frac{b}{\parallel \hat{p}_{kn} \parallel^2} - a \ln\left(4D^2 - \parallel \hat{p}_{kn} \parallel^2\right) + c
\tag{9}
$$

The gradient of the potential function is:

$$
\nabla_{\hat{p}_{kn}} V(\parallel \hat{p}_{kn} \parallel) = -2\hat{p}_{kn} \left(\frac{b}{\parallel \hat{p}_{kn} \parallel^4} - \frac{a}{4D^2 - \parallel \hat{p}_{kn} \parallel^2} \right)
\tag{10}
$$

where a, b and c are positive number. $\frac{b}{\parallel \hat{p}_{kn} \parallel^4}$ stands for the magnitude of the repulsion term, and $\frac{a}{4D^2 - \parallel \hat{p}_{kn} \parallel^2}$ stands for the magnitude of the attraction term. When $\frac{b}{\parallel \hat{p}_{kn} \parallel^4} = \frac{a}{4D^2 - \parallel \hat{p}_{kn} \parallel^2}$, there exits a balance between attraction and repulsion term [21]. So, the formation can be finished.

2.2.2. Optimization of Control Parameter

Each acceleration equation of follower has respective control parameter C_k, which can alter the movement condition of corresponding robotic fish at time t. When C_k has different values, the value of the acceleration will be different. As a result, the velocity and route of the robotic fish will be changed. So, energy consumption of the formation will be changed. In order to reduce the energy consumption, particle swarm optimization can be utilized to optimize the parameter C_k.

The Particle Swarm Optimization algorithm was proposed in 1995, which combined with the process of the study of swarm intelligence and human [22]. In an E-dimensional optimization space, the swarm is composed of N particles named $N = (n_1, \ldots, n_N)$. Each particle k is an E-dimensional vector, whose position and velocity are respectively expressed with $X_k = (X_{k1}, \ldots, X_{kE})^T$ and

$V_k = (V_{k1}, \ldots, V_{kE})^T$. The individual optimal position is stored with $p_i = (P_{k1}, \ldots, P_{kE})^T$ while the global optimal position is stored with $P_g = (P_{g1}, \ldots, P_{gE})^T$. The position and velocity of each particle k are updated by:

$$V_{kd}(t+1) = WV_{kd}(t) + H_1r_1(P_{kd}(t) \\ - X_{kd}(t) + H_2r_2(P_{gd}(t) - X_{gd}(t))) \tag{11}$$

$$X_{kd}(t+1) = X_{kd} + V_{kd}(t+1) \tag{12}$$

where $1 \le d \le E$, $1 \le I \le N$, Learning factors, H_1 and H_2 are accelerative coefficients, which impact the convergence speed of the PSO algorithm. H_1 and H_2 are determined as rand (0, 2] and rand (0, 2] respectively While r_1 and r_2 are determined as rand (0, 1) and rand (0, 1). W is the inertia weight which determines how much the previous velocity influence the current speed. Thus W can balance the ability of global search and local search. The ability of global search includes the previous and current search, while the ability of local search includes the current search. A linear decrease strategy has been put forward for a better optimization [23,24]:

$$W = W_{max} - \frac{h}{h_{max}}(W_{max} - W_{min})$$

where W_{max} is the maximum value of W and W_{min} is the minimum value. h and h_{max} indicate the current and maximum iterations, respectively.

According to the principle of particle swarm optimization, and the instance of distributed formation flocking algorithm, control parameter C_k is regarded as the variable while the total energy consumption in the process of robotic fish formation is regarded as the function. The relation between variable and function is built by formation algorithm and power consumption algorithm.

Figure 5 shows the process of the particle swarm algorithm.

Step one: Initialize the size of the population. Presuppose the size of the population $N = 30$. Because there are ten robotic fishes for formation and parameter C_k varies from individual to individual, parameter C_k is a ten-dimensional vector. considering that parameter C_k has scope limitation, we initialize the position X_{kd} and velocity V_{kd} of each particle k in a certain range.

Step two: Calculate the function as follows, according to the relationship between the parameter C_k and energy loss. Firstly, calculate the fitness value($f(X_{kd}(t))$) of each particle in the population; Then, calculate the individual optimal position p_k of each particle in the current moment; Finally, calculate the global optimal position p_g of robotic fish system in the current moment.

Step three: Calculate and update the velocity and position of the particle. Because parameter C_k is in a range, the velocity and position of the particle should also be in a range. The velocity shouldn't be greater than maximum velocity V_{max}. Otherwise, the velocity will be replaced with maximum velocity V_{max}. In the same way, if the position updated of the particle is beyond the range of C_k, the position should be replaced with boundary value of C_k.

Step four: Compare the current fitness value of each particle with the optimal value of individual fitness. P_k should be replaced with the better one.

Step five: Compare the current fitness value of each particle with the optimal value of global fitness. P_g should be replaced with the better one.

Step six: If the precision meets the requirement or the number of iteration reaches the maximum, the algorithm will be stopped. Otherwise the circulation should return to the step two.
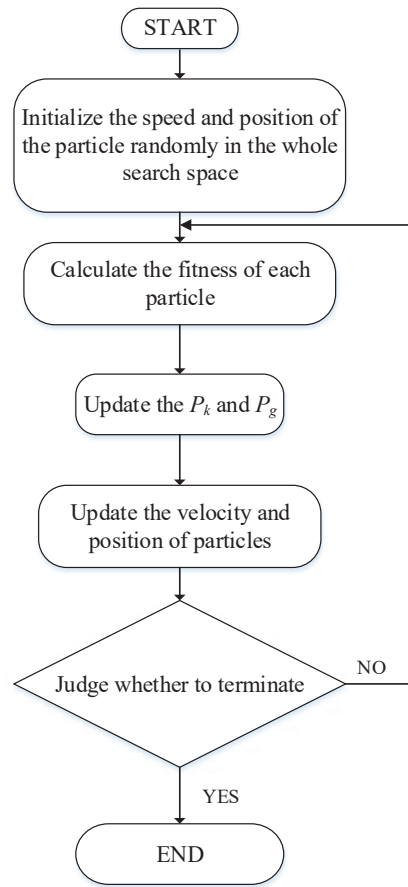
Figure 5. Flowchart for PSO algorithm.

2.2.3. The Proof of the Stability

It is defined that $\hat{\theta}(t) = (\hat{\theta}_2(t), \ldots, \hat{\theta_N}(t))^T$, $\hat{q}(t) = [\hat{q}_2(t)^T, \ldots, \hat{q}_N(t)^T]^T$, as while as $\tilde{p}(t) = [\tilde{p}_{11}(t)^T, \ldots, \tilde{p}_{1N}(t)^T, \ldots, \tilde{p}_{N1}(t)^T, \ldots, \tilde{p}_{NN}(t)^T]^T$. Make $N_l = \{k | a_{k1} = 1, k \in F\}$ denote the set of followers who has one leader neighbor on time interval $[t_r, t_{r+1})$. Let the following energy function as the common Lyapunov function [7]. If the value of derivative is less than zero, the energy optimized leader-follower formation algorithm is stable.

$$E(\hat{\theta}(t), \hat{q}(t), \tilde{p}(t)) = E_1(t) + E_2(t) + E_3(t) + E_4(t) \tag{13}$$

Therein:

$$
\begin{aligned}
E_1(t) &= \frac{1}{2}V(t) = \sum_{l \in F}\sum_{n \in N_k(t)} V(\| \hat{p}_{kn} \|) + \sum_{n \in N_k(t)} V(\| \hat{p}_{1n} \|) \\
E_2(t) &= \frac{1}{2}\sum_{i \in F} \hat{q}_k(t)^T \hat{q}_k(t) \\
E_3(t) &= \frac{1}{2}\theta(t) = l_d^2 \sum_{k \in F}\sum_{n \in N_k(t)} C_k \hat{\theta}_k(t)(\hat{\theta}_k(t) - \hat{\theta}_n(t)) \\
E_4(t) &= \frac{1}{4}\sum_{k \in F}(C_k - m)^2
\end{aligned}
\tag{14}
$$

where: C_k is the control parameter and the constants that are in a range. The derivative of $E(\hat{\theta}(t), \hat{q}(t), \tilde{p}(t))$ during the time interval $[t_r, t_{r+1})$ is:

$$\frac{dE}{dt} = \frac{dE_1}{dt} + \frac{dE_2}{dt} + \frac{dE_3}{dt} + \frac{dE_4}{dt} \tag{15}$$

Consider parameter C_k is a constant that is in a range, $E_4(t)$ is a constant, in which case, $\frac{dE_4}{dt}$ is equal to zero. So:

$$\begin{aligned} \frac{dE}{dt} &= \frac{dE_1}{dt} + \frac{dE_2}{dt} + \frac{dE_3}{dt} \\ &= -\sum_{k\in F}\sum_{n\in N_k} C_k \hat{q}_k(t)^T (\hat{q}_k(t) - \hat{q}_n(t)) \\ &\quad - \sum_{k\in F}\sum_{n\in N_k} C_k \dot{\hat{p}}_k(t)^T (\dot{\hat{p}}_k(t) - \dot{\hat{p}}_n(t)) \end{aligned} \tag{16}$$

Because that $\frac{dE}{dt} \leq 0$ and thus, the distributed formation flocking optimization algorithm asymptotically is stable [9].

When the potential energy reaches a minimum value, the individual distance of the system reaches a stable value. Therefore, if we can ensure the system's potential energy reaches a minimum value when the individual distance of the system reaches a expected value, we can realize any formation we want. For one geometry formation, it is composed of N vertexes which are built by the regular $p_k^d = [x_k^d, y_k^d], k = 1, \ldots, N$. In order to realize the expected formation, the acceleration equation of the followers should be changed as follow:

$$\begin{aligned} a_k(t) = &-\sum_{n\in N_k(t)} c_k(\hat{v}_k(t) - \hat{v}_n(t)) \\ &- \sum_{n\in N_k(t)} c_k \dot{\hat{p}}_{kn}(t)^T \overrightarrow{t_k}(t) \\ &- \sum_{n\in N_k(t)} \nabla_{\hat{p}_{kn}(t)} V(\| \tilde{p}_{kn}(t) \|)^T \overrightarrow{t_k}(t) \end{aligned} \tag{17}$$

$$\begin{aligned} b_k(t) = &-\sum_{n\in N_k(t)} c_k l_d(\hat{\theta}_k(t) - \hat{\theta}_n(t)) \\ &- \sum_{n\in N_k(t)} c_k l_d(\hat{\omega}_k(t) - \hat{\omega}_n(t)) \\ &- \sum_{n\in N_k(t)} c_k \dot{\hat{p}}_{kn}(t)^T \overrightarrow{n_k}(t) \\ &- \sum_{n\in N_k(t)} \nabla_{\hat{p}_{kn}(t)} V(\| \tilde{p}_{kn}(t) \|)^T \overrightarrow{n_k}(t) \end{aligned} \tag{18}$$

where $\| \tilde{p}_{kn}(t) \| = \frac{\|\hat{p}_{kn}(t)\|}{p_{kn}^d}$, when the robotic fish system realizes a stable formation, potential function $V(\| \tilde{p}_{kn}(t) \|) = 1$.

3. Results

In order to evaluate to the proposed optimization scheme reducing energy consumption during the formation, dynamic simulations are conducted on MATLAB2018b. Initial conditions of each robotic fish are built. The simulation time is 200 s while the sample step is 0.01 s. we can express the state of the robotic fish at time t with $S_k(t) = (x_k(t), y_k(t), \theta(t), v_k(t), \omega_k(t))$. The units of components are m, m, rad, m/s, rad/s respectively. Give the initial conditions:

$S_1(0) = (-12.2, -13.09, -0.34, 0.76, -0.29)$,
$S_2(0) = (-3.7, -11.78, -1.76, 0.54, 0.7)$,

$$S_3(0) = (13.8, -4, 0.6, 5.3, 0.08),$$
$$S_4(0) = (4.82, 0.1, -1.1, 7.8, -0.38),$$
$$S_5(0) = (-5.8, 4.6, 0.07, 9.34, 0.013),$$
$$S_6(0) = (4.97, 19.2, -0.84, 1.3, -0.22),$$
$$S_7(0) = (-2.6, 6.1, 0.34, 5.7, -0.275),$$
$$S_8(0) = (15.1, 0.6, -0.92, 4.7, -0.34),$$
$$S_9(0) = (-4.56, 8.45, -0.12, 0.12, 0.6),$$
$$S_{10}(0) = (5.45, -1.42, -0.92, 3.37, 0.31).$$

Lambdoid-shaped, ring-shaped and rectangle-shaped formation are studied. The vertex positions for lambdoid-shaped formation can be denoted as $p_1^{d_1} = [-3, 0]$, $p_2^{d_1} = [-2.2, 0.6]$, $p_3^{d_1} = [-1.4, 1.2]$, $p_4^{d_1} = [-0.6, 1.8]$, $p_5^{d_1} = [0.2, 2.4]$, $p_6^{d_1} = [1, 3]$, $p_7^{d_1} = [1.8, 2.4]$, $p_8^{d_1} = [2.6, 1.8]$, $p_9^{d_1} = [3.4, 1.2]$, $p_{10}^{d_1} = [4.2, 0.6]$. The vertex positions for ring-shaped formation can be denoted as $p_k^{d_2} = [2 \times \cos(k \times \pi/5), 2 \times \sin(k \times \pi/5)], k = 1, \ldots, 10$ while for rectangle-shaped formation the positions of the vertex are denoted as $p_1^{d_3} = [0, 0]$, $p_2^{d_3} = [1, 0]$, $p_3^{d_3} = [2, 0]$, $p_4^{d_3} = [3, 0]$, $p_5^{d_3} = [3, 1]$, $p_6^{d_3} = [3, 2]$, $p_7^{d_3} = [2, 2]$, $p_8^{d_3} = [1, 2]$, $p_9^{d_3} = [0, 2]$, $p_{10}^{d_3} = [0, 1]$. The number of iterations h_{max} is 500.

The Change of W_{max}

A linear decrease strategy about W can improve the ability of optimization [23]. When W is given different values, different results for optimization will be produced. In the beginning of particle swarm algorithm, the larger inertia weight is, the stronger ability of global search is. As time goes on, the smaller inertia weight is, the stronger ability of local search is. Let W_{min} is equal to 0.5, and add 0.1 every time. Make $W_{max} - W_{min} = 1$.

Table 1 shows that the value of C_i and total energy consumption vary from different W_{max} in the lambdoid-shaped formation. When W_{max} is equal to 0.8, the total energy consumption in the formation is the minimum. In the beginning, the value of W_{max} is 0.8, the ability of global search is strong to explore more value of control parameter. As time goes, the value of W_{max} declines, and the ability of local search become stronger. Thus, particle swarm algorithm can select the appropriate value. Under the given initial conditions, when $C_k = (1.3, 1.3, 0.6608, 1.3, 1.3, 0.7657, 0.6, 1.3, 1.8738), k = 2, \ldots, 10$, ten robotic fishes consume the least energy.

Table 1. Value of C_k and energy consumption in the lambdoid-shaped formation due to different W.

W_{max}	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4
C_2	1.5088	1.1378	1.2397	1.3	1.4793	0.861	1.3	1.3	1.5989	1.4157
C_3	0.8598	1.3096	1.3521	1.3	1.8527	0.7395	1.3	1.3	1.0538	0.8627
C_4	1.2889	1.4393	1.071	0.6608	0.8887	0.7652	0.6	0.6	1.0298	1.0283
C_5	1.842	1.364	1.8197	1.3	1.6125	0.9338	1.3	1.3	1.1966	1.135
C_6	1.734	1.7699	1.4217	1.3	1.7243	1.1482	1.3	0.8494	1.1617	1.5876
C_7	1.2904	0.8963	1.2625	0.7657	1.5769	0.9909	1.3	0.6	1.6805	1.6112
C_8	1.5752	0.7295	0.9108	0.6	1.3054	1.0422	0.8243	1.3	0.8337	1.827
C_9	1.3586	0.9115	1.7968	1.3	1.8707	0.1306	1.3	0.6	1.6113	1.7266
C_{10}	1.2396	0.9763	0.6826	0.8738	1.3624	1.0658	0.6	1.3	0.9258	0.9837
Energy/$10^6 J$	1.44	1.4	1.41	1.17	1.41	1.92	1.3	1.18	1.44	1.54

Table 2 demonstrates that the value of C_k and total energy consumption vary between different W_{max} in the ring-shaped formation. When W_{max} is equal to 0.7, the total energy consumption of ten robotic fishes in the formation is the minimum. When the value of W_{max} is 0.7, the ability of global search and local search can be balanced well. As a result, the algorithm can find more suitable value of control parameter. Under the given initial conditions, when $C_k = (0.74, 0.6, 0.9041, 1.3, 1.3, 0.74, 0.74, 0.74, 0.74), k = 2, \ldots, 10$, ten robotic fishes consume the least energy.

Table 2. Value of C_k and energy consumption in the ring-shaped formation due to different W.

W_{max}	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4
C_2	1.3	0.6964	0.74	1.3	0.8047	1.3	1.3	1.3	0.6	1.3
C_3	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	1.3	0.6
C_4	1.3	1.3	0.9041	1.3	1.3	1.3	1.3	0.9855	1.1883	1.3
C_5	1.1178	0.6	1.3	1.3	1.3	0.6	1.3	1.2423	0.6	0.6
C_6	1.3	1.3	1.3	0.816	1.3	1.3	0.971	0.6	1.3	0.6
C_7	0.7205	0.6608	0.74	1.3	0.9921	1.3	0.6	1.3	1.3	1.0629
C_8	1.1277	0.6608	0.74	1.1326	1.246	1.3	1.3	1.3	1.3	1.3
C_9	1.2549	0.6608	0.74	1.3	1.3	1.3	1.3	0.6978	1.3	0.9269
C_{10}	1.3	1.3	0.74	1.0112	0.6	0.6	1.3	1.3	1.3	1.3
Energy/$10^5 J$	7.92	7.99	7.62	8.21	8.37	8.27	7.88	8.06	7.93	8.53

Table 3 shows that the value of C_k and total energy consumption vary from different W_{max} in the rectangle-shaped formation. When W_{max} is equal to 0.6, the total energy consumption of ten robotic fishes is the minimum. when $C_k = (1.2075, 1.3, 0.8908, 0.7156, 1.055, 1.3, 0.6534, 1.3, 1.3), k = 2, \ldots, 10$, ten robotic fishes consume the least energy.

Table 3. Value of C_k and energy consumption in the rectangle-shaped formation due to different W.

W_{max}	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4
C_2	0.6	1.2057	1.3	1.3	1.3	1.3	1.3	1.3	0.7074	1.3
C_3	0.6	1.3	0.6	0.6	1.2075	0.6	0.6	0.6	0.7074	1.3
C_4	0.9978	0.8908	1.3	1.1176	1.1588	1.3	1.3	0.6	1.3	1.3
C_5	1.3	0.7156	1.3	1.3	1.2166	1.3	1.3	1.3	0.6	0.6
C_6	1.3	1.055	1.3	0.8204	0.9112	0.6	0.6	1.3	1.3	1.3
C_7	1.3	1.3	1.3	1.3	1.3	1.3	1.2881	0.6	1.3	1.3
C_8	1.3	0.6534	1.284	1.3	0.9766	1.3	1.3	1.3	0.6	1.3
C_9	1.3	1.3	0.7129	0.6	0.9719	0.7216	0.6	0.9332	1.3	1.3
C_{10}	1.3	1.3	1.3	1.2942	1.3	1.3	1.3	1.3	1.3	1.3
Energy/$10^5 J$	7.23	6.35	7.48	6.65	6.73	7.82	7.06	7.48	6.65	7.85

4. Discussion

Different formation has different set of C_k. To better illustrate the proposed optimization algorithm, the optimized energy is compared with the energy without optimization(fixed value).The experiment is finished in the computer whose motherboard is Z270 and CPU is i7-7700.

Table 4 shows the value of C_k of three kinds of formation before and after optimization. Table 4 indicates comparison of total energy consumption of three kinds of formation. Make ten fishes as a whole. Each formation consumes a corresponding energy under three different kinds of parameters C_k. Compare the three cases of energy, and Table 4 shows that the whole fishes in the three kinds of formation consume the least energy in which case that parameter C_k is optimized. Lambdoid-shaped formation gets best energy optimization. During the process of formation, because the control parameters are optimized, the corresponding robotic fish will choose the better routine and consume little energy to finish the formation. To sum up, when ten robotic fishes conduct only one formation from the very beginning, the proposed algorithm can succeed to optimize the total energy consumption. However, as we can see from the Table 4, optimal control parameters of different formations are different. If we want to get each formations with least energy consumption, we need to choose different set of control parameters. This is the disadvantage of our algorithm. We cannot get the same value of control parameters to finish different formation.

Table 4. The comparison of total energy consumption.

C_k	(a) The Value of C_k.			
	After Optimization			Before Optimization
	C_k^{re}	C_k^{la}	C_k^{ri}	Fixed
C_2	1.2057	1.3	0.74	1
C_3	1.3	1.3	0.6	1
C_4	0.8908	0.6608	0.9041	1
C_5	0.7156	1.3	1.3	1
C_6	1.055	1.3	1.3	1
C_7	1.3	0.7657	0.74	1
C_8	0.6534	0.6	0.74	1
C_9	1.3	1.3	0.74	1
C_{10}	1.3	0.8738	0.74	1
(b) Total Energy of Each Formation /J.				
	Rectangle-Shaped	Lambdoid-Shaped		Ring-Shaped
After optimization	6.35×10^5	1.17×10^6		7.62×10^5
Before optimization	6.4×10^5	2.27×10^6		1.22×10^6

Besides, power of ten agents in the formation can be simulated. Figure 6 demonstrates the power of ten agents in the ring-shaped formation. Figure 6b shows that when the control parameters are optimized, the whole fishes will quickly end the energy fluctuation period, and will finish the formation with the steady energy consumption. So, the formation will be completed quickly. Figure 7 indicates the power of ten agents in the rectangle-shaped formation. The two curves are similar. The time when formation transits to the smooth energy consumption is similar. However, before formation, the energy is consumed a little more. Figure 8 shows the power of ten agents in the lambdoid-shaped formation. Although the volatility is larger after optimizing control parameter, finishing time is shorter. thus, the energy consumption is reduced. Figure 9 shows the finishing time in three kinds of situation. This indicates that the proposed algorithm not only reduces the energy consumption in the process of formation, but also shorten the formation time.

(**a**) before optimization   (**b**) after optimization
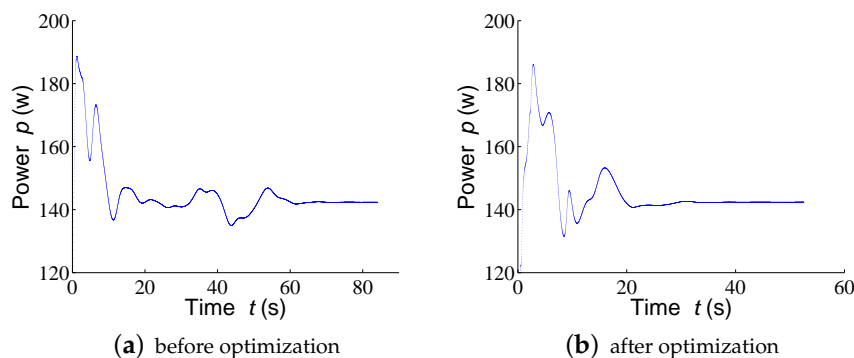
Figure 6. Power of ten agents in the ring-shaped formation.

(**a**) before optimization

(**b**) after optimization
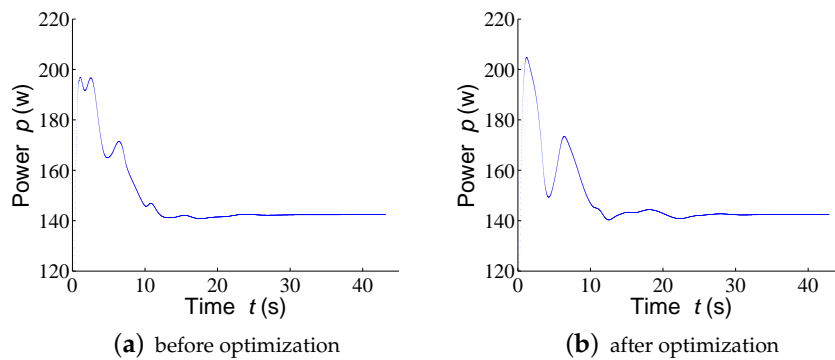
Figure 7. Power of ten agents in the rectangle-shaped formation.

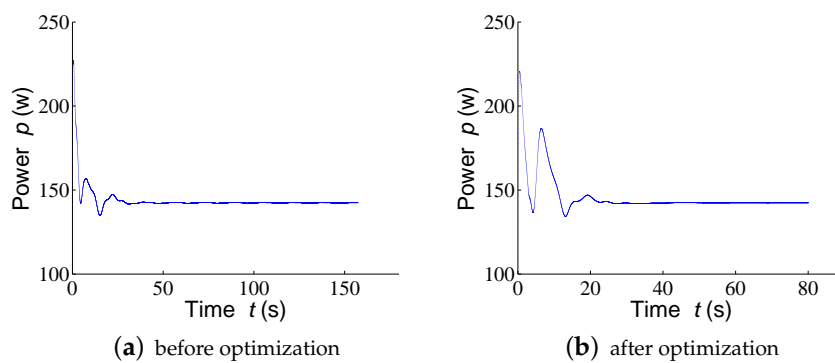(**a**) before optimization

(**b**) after optimization

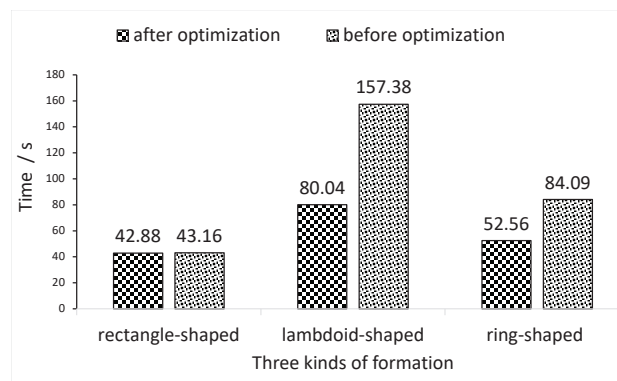**Figure 8.** Power of ten agents in the lambdoid-shaped formation.

Figure 9. Total energy consumption in the different situations.

Nowadays, we are trying to build the experimental platform to do do the experiment in a real world. We are using hardware such as STM32F407 and software such as Keil to build platform. In the future, we will build up a more general swimming model of the robotic fish by considering the complex stress. In addition, we will take the task to seek general control parameter to adapt to different formation transformation.

5. Conclusions

Nowadays, few papers focus on the energy consumption of total multi-agent during the formation. Our work can fill in this gap. This paper has presented an application of PSO to optimize the control parameters C_ks to efficiently reduce energy consumption during the formation. Although that not

all formation algorithms have C_ks, each algorithm has control parameters that control the movement of agents. Although different formations have different control parameter, we can choose different optimal parameters by using particle swarm optimization algorithm to finish the formation with lower energy consumption. After optimizing the control parameter, individual robotic fish has their own suitable parameters and they will choose perfect routine to finish their formation. Besides, finishing time will be shorter.

Author Contributions: For research articles with several authors, D.X. conceived and designed the experiments, and analyzed the data ; L.Y. contributed to the writing and editing of the manuscript; Z.L. is helpful to the comparative experiment of resubmitted paper; J.Z. performed the experiments; D.F. built the model; W.D. is experienced in the assembly process in the case.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yu, J.; Tan, M.; Wang, S.; Chen, E. Development of a biomimetic robotic fish and its control algorithm. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2004**, *34*, 1798–1810. [CrossRef]
2. Yu, J.; Wu, Z.; Wang, M.; Tan, M. CPG Network Optimization for a Biomimetic Robotic Fish via PSO. *IEEE Trans. Neural Net. Learn. Syst.* **2016**, *27*, 1962. [CrossRef] [PubMed]
3. Triantafyllou, M.S.; Triantafyllou, G.S. An Efficient Swimming Machine. *Sci. Am.* **1995**, *272*, 64–70. [CrossRef]
4. Gu, D.; Wang, Z. Leader-Follower Flocking: Algorithms and Experiments. *IEEE Trans. Control Syst. Technol.* **2009**, *17*, 1211–1219.
5. Yu, W.; Chen, G.; Cao, M. Distributed leader-follower flocking control for multi-agent dynamical systems with time-varying velocities. *Syst. Control Lett.* **2010**, *59*, 543–552. [CrossRef]
6. Su, H.; Wang, X.; Chen, G. Rendezvous of multiple mobile agents with preserved network connectivity. *Syst. Control Lett.* **2010**, *59*, 313–322. [CrossRef]
7. Jia, Y.; Zhang, W. Distributed adaptive flocking of robotic fish system with a leader of bounded unknown input. *Int. J. Control Autom. Syst.* **2014**, *12*, 1049–1058. [CrossRef]
8. Reynolds, C.W. *Flocks, Herds and Schools: A Distributed Behavioral Model*; ACM: New York, NY, USA, 2015; pp. 25–34.
9. Jia, Y.; Wang, L. Leader-Follower Flocking of Multiple Robotic Fish. *IEEE/ASME Trans. Mechatron.* **2015**, *20*, 1372–1383. [CrossRef]
10. Hu, Q.; Zhou, H. IPMC propelled biomimetics robotic fish energy consumption model construction and its application to energy-saving control. In Proceedings of the 2009 IEEE International Conference on Robotics and Biomimetics, Guilin, China, 19–23 December 2009; pp. 2151–2156.
11. Wang, M.C.C.; Wang, W.; Xie, G.; Shi, H. Real-time energy monitoring and management system and its application on bionic robotic fish. In Proceedings of the 11th World Congress on Intelligent Control and Automation, Shenyang, China, 29 June–4 July 2014; pp. 3256–3261.
12. Zhu, W.; Wang, X.; Xu, M.; Yang, J. A wave energy conversion mechanism applied in robotic fish. In Proceedings of the Ieee/asme International Conference on Advanced Intelligent Mechatronics, Wollongong, Australia, 9–12 July 2013; pp. 319–324.
13. Shen, Y.; Guo, B. Energy-efficient cluster-head selection with fuzzy logic for robotic fish swarm. In Proceedings of the International Conference on Fuzzy Systems and Knowledge Discovery, Zhangjiajie, China, 15–17 August 2015; pp. 513–518.
14. Tanner, H.G. On the controllability of nearest neighbor interconnections. In Proceedings of the 43rd IEEE Conference on Decision and Control, Nassau, Bahamas, 14–17 December 2004; pp. 2467–2472.
15. Low, K.H. Locomotion and Depth Control of Robotic Fish with Modular Undulating Fins. *Int. J. Autom. Comput.* **2006**, *3*, 348–357. [CrossRef]
16. Zhang, D.; Wang, L.; Yu, J.; Tan, M. Coordinated Transport by Multiple Biomimetic Robotic Fish in Underwater Environment. *IEEE Trans. Control Syst. Technol.* **2007**, *15*, 658–671. [CrossRef]

17. Klein, D.J.; Bettale, P.K.; Triplett, B.I.; Morgansen, K.A. Autonomous underwater multivehicle control with limited communication: Theory and experiment. *IFAC Proc. Vol.* **2008**, *41*, 113–118. [CrossRef]

18. Anderson, J.M.; Streitlien, K.; Barrett, D.S.; Triantafyllou, M.S. Oscillating foils of high propulsive efficiency. *J. Fluid Mech.* **1998**, *360*, 41–72. [CrossRef]

19. Liang, J.; Wang, T.; Wen, L. Development of a two-joint robotic fish for real-world exploration. *J. Field Robot.* **2011**, *28*, 70–79. [CrossRef]

20. Xu, D.; Zhang, S.; Wen, L. A Stiffness-Adjusting Method to Improve Thrust Efficiency of a Two-Joint Robotic Fish. *Adv. Mech. Eng.* **2014**, *2014*, 1–7. [CrossRef]

21. Latombe, J.C. *Robot Motion Planning*; Kluwer Academic Publishers: Alphen, The Netherlands, 1991; pp. 628–649.

22. Kennedy, J.; Eberhart, R. Particle swarm optimization. In *Encyclopedia of Machine Learning*; Springer: Boston, MA, USA, 1995; pp. 1942–1948.

23. Shi, Y.; Eberhart, R. A modified particle swarm optimizer. In Proceedings of the IEEE World Congress on Computational Intelligence, 1998 IEEE International Conference on Evolutionary Computation Proceedings, Anchorage, AK, USA, 4–9 May 1998; pp. 69–73.

24. Shi, Y.; Eberhart, R.C. Empirical study of particle swarm optimization. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99, Washington, DC, USA, 6–9 July 1999; pp. 1945–1950.