A Modular Robot Transplantation Method Based on the Improved Inverse Distance Weighting Method Proxy Model

YANG LIU1, YIYING WANG2, SHANFEI YAN2, JINXI GUO2, AND YANNAN SHI2

1School of Mechanical Electronic and Information Engineering, China University of Mining and Technology (Beijing), Beijing 100083, China
2College of Mechanical and Equipment Engineering, Hebei University of Engineering, Handan 056000, China

Corresponding author: Yiying Wang (yatesyy@163.com)

This work was supported in part by the Youth Foundation Project under Grant QN2019214.

ABSTRACT When integrating evolutionary algorithms into the process of the design of modular robots, the “reality gap” problem leads to the difference between simulation and reality. A modular robot transplantation method based on the improved inverse distance weighting method proxy model is presented. Based on the plane moving task, the portability of the modular robot consisting of EMERGE modules of different shapes and types was experimentally studied. The information of other real robots can be predicted according to the limited evolutionary data of real robots. It is no need to do all the evaluation experiments on a physical robot. The number and complexity of experiments are reduced. The experiment efficiency is improved. This method can effectively reduce the performance difference between the results of simulation and the real robot caused by the “reality gap” problem, and the portability of modular robots improves greatly.

INDEX TERMS Modular robot, inverse distance weighting method, reality gap, agent model, migration method.

I. INTRODUCTION A modular robot is a complex multi-module robot system. Facing the uncertain working environment, noise interference and the error of sensors and actuators. The coordination and control of each module of the system became more complicated, it is very difficult to design a precise robot controller. In recent years, some scholars have gradually tried to integrate the evolutionary algorithm into the design process of modular robots, in order to design modular robots with better performance automatically in the interaction with the working environment.

Some scholars choose to directly evolve on the real robot to seek the optimal solution. The first relevant experiment is based on the Khepera mobile robot [1], the task objective is to evolve the controller in the cruise task of the maze, so as to find the controller with good robustness. The same method was successfully applied to the SONY AIBO robot [2] and a nine-legged robot [3]. The optimization process on the physical machine takes a long time, and the long time running will bring wear and even failure of mechanical devices and driving mechanism.

In order to solve the disadvantage of applying the evolutionary algorithm directly to the real robot, it can first evolve in the simulation environment, and then we transplant the best robot obtained in the simulation to the real robot [4], however, since various parameters in the simulation environment cannot be exactly different from those in the real world, there will be a difference between the simulation and the reality. So that the performance of the robot may be good in the simulation environment, but the performance decreases in the real application. There exist a “reality gap” problem.

A variety of methods were proposed to solve the “reality gap” problem, it can be divided into simulation-based optimization method and simulation-based transplantation optimization method. The minimum modeling method [5] is the most representative in this simulation-based transplantation optimization method, the interaction characteristics of the robot environment related to the expected behavior are correctly modeled in the simulation. The robustness of the system greatly enhances. Similar methods have been successfully applied to an eight-legged robot [6], and Khepera mobile robot based on a T-shaped maze [7]–[9]. The mechanism of
the adaptive online learning [10], [11] transplants the solution of simulation to the real robot, and establish the model of a real robot and the environment interaction. Then this model, as a benchmark model, provides instruction about evolution for the next generation, as a reference of the adaptive mechanism, adjusts the difference between simulation and reality. The robustness of these two methods is not enough to overcome the problem of performance degradation, the local optimal solution obtained on the real robot may lead to performance degradation. The transplantation optimization method based on simulation is still based on simulation optimization, but transplantation experiments can be carried out in the optimization process. There are representative studies on this aspect: The first one is the transferability approach method proposed by Koos [12] et al., which uses the interpolation method to establish the agent model of the robot, but the value of the weight power index is 2, which is not scientific. When the sample points are sparse, the weight power index should be appropriately increased to improve the interpolation accuracy. One is the multi-objective optimization method proposed by Bongard [13], [14] et al., which is optimized with two objectives, one is the controller and the other is the simulation model. Transplanting a controller evolved to the real robot, then the simulation model is optimized with the observed data from the real robot, the controller has evolved again on the optimized model, ordinal iteration until evolution condition is met. Relative research work was confirmed on quadruped robot [15]. The premise of this method is to ensure a certain number of migration times and the accuracy of the simulation model. Another method is based on reinforcement learning [16]–[18]. It does not depend on an evolutionary algorithm, however, it is operated by actual professional staff, in order to get some actual data. Then an approximate agent model based on the actual data is established, it gives corresponding learning strategies, the learning strategy is transplanted to the real robot to obtain new actual data. The agent model is corrected until the learning strategy can apply to the real robot by iterating this process. The using premise of this method is to have professional staff conduct artificial demonstration operations.

In summary, when evolutionary mechanisms are used to estimate the performance of modular robots in a robotic simulation platform, the simulated performance may be good, but when it is transplanted to the physical robot, the performance is degraded, that is, the reality gap occurs. How to reduce the difference in robot performance between simulation and physical is of great significance to the practical application of modular robots. Some studies have achieved preliminary results, but the prerequisites for using are too harsh, this is reflected in several aspects: the scope of application is limited, and the performance difference is still large. Therefore, there is an urgent need to study a transplantation approach which is easy to use and has a significant improvement in performance.

Therefore, in order to effectively reduce the performance difference between simulation results and real robots caused by the “reality gap” problem. An improved inverse distance weighting method based on the azimuth relation of the sample is proposed, the agent model of the difference between robot simulation and entity behavior is established, the algorithm process of the transplant method is studied. Based on the planar movement task, the portability of a modular robot composed of EMERGE modules of different morphologies was studied. The information of other real robots can be predicted according to the limited evolutionary data of real robots, we don’t have to do all the evaluation experiments on the real robot. This method can reduce the number of experiments and the complexity of the experiment process and improve the experiment efficiency. This method can reduce the “reality gap” problem effectively, which causes the performance difference of the simulation results and the real robot and improves the portability of modular robots.

This paper is organized as follows. It is improving the inverse distance weighting method in Section II. In sequence, the transplantation method is described in Section III. The experimental results are presented in Section IV. Some conclusions are given in Section V.

II. IMPROVING THE INVERSE DISTANCE WEIGHTING METHOD

The Inverse Distance Weighting method [19] belongs to geometric methods. The value of its estimated point is equal to the weighted average of The Distance Weighting value of the known data point. The closer the known data point is to the estimated point, the greater the weight, on the contrary, the smaller the weight, that is the approximate similarity principle.

The weighted index affects the change of the weight value of the known point to the estimated point according to the exponential rule, as the distance increases, the weight value decreases following the exponential rule, which means that the spatial correlation of other data points decreases. On the contrary, as the distance decreases, the weight value increases according to the exponential rule, which increases the spatial correlation of the known data points to the estimated point. The advantage of the inverse distance weighting method is that the principle is easy to understand, simple to use, the algorithm is easy to implement and the amount of calculation is small. The difficulty lies in how to improve the accuracy of interpolation results, which is related to the optimization of weight index and search mode of reference points in interpolation parameters. It can be seen that the difficulty and key of this research on the weight index and search mode lie in how to set them adaptively according to the research situation. Therefore, based on the azimuth and distance of the sample points known [20], the method of adaptive allocation of weight index is proposed according to the orientation of the sample points. It makes the distribution of sample points average adaptively, at the same time weight index can also be allocated adaptively, so as to improve the efficiency and precision of interpolation. The agent model based on the
behavior of the robot is more precise, the portability of the modular robot is improved further.

A. A FUNDAMENTAL ASSUMPTION

The biggest difference between the method proposed in this chapter and IDW (inverse distance weighting method) lies in that, based on the harmonic weight coefficient “T” which can reflect the azimuth relation between the estimated point and the known sample point, the weight index harmonic coefficient “K” based on the azimuth relation of the sample points is introduced. The basic assumption of the method is that: when the angle between the two sample points and the estimated point is \( \beta \leq 360^\circ / n \) (“n” is the number of sample points known), the azimuth between the sample points has blocking effect, the weight value of the blocked sample points will decrease, the degree of reduction is determined “T”. The sample points blocked completely will be removed (\( T = 0 \)). When \( \beta > 360^\circ / n \), there is no blocking effect between sample points. At this time, \( T = 1 \). According to the literature [21], it can be seen that it is not scientific that the numerical value of the weight index is 2. When the sample points are sparse, the weight index should be appropriately increased to improve the interpolation accuracy. Of course, the weight index should not be too high. Considering the calculation complexity and calculation accuracy, the upper limit of the weight index is set as 5. Therefore, based on the azimuth relation between the sample points, it is stipulated that when the angle sum between a sample point and two adjacent sample points on both sides and the estimated point is \( \alpha_1 + \alpha_2 \leq 360^\circ / n \), the sample point distribution is dense, and its weight index harmonic coefficient is \( K = 2 \). Otherwise, when \( \alpha_1 + \alpha_2 > 360^\circ / n \), the sample points are sparse. In this case, according to the different sparsity, the value of the harmonic coefficient of the weight index is also different, which can be divided into the following three cases to judge sparsity. When \( \alpha_1 < 360^\circ / n \lor \alpha_2 < 360^\circ / n \), the sparsity is small, the numerical value \( K = 3 \); When \( \alpha_1 \geq 360^\circ / n \lor \alpha_2 \geq 360^\circ / n \), the sparsity is moderate, the numerical value \( K = 4 \); When \( \alpha_1 > 360^\circ / n \lor \alpha_2 > 360^\circ / n \), the sparsity is large, the numerical value \( K = 5 \).

In order to illustrate this basic hypothesis more vividly, a schematic diagram of blocking and sparse and dense distribution between the sample points is drawn. As shown in Fig.1, \( Z_1 \sim Z_5 \) is the known sample point, \( Z_0 \) is the estimated point. According to the basic assumption, the judging angle of the relation of blocking and sparsity is \( 72^\circ (360^\circ / 5) \). In figure (a), Angle between any two sample points and the estimated point is greater than \( 72^\circ \), there is no blocking between the sample points, and the distribution is sparse and uniform. In figure (b), the Angle \( \beta \) of the three groups of sample points(\( Z_1 \) and \( Z_2, Z_3 \) and \( Z_5 \), \( Z_2 \) and \( Z_3 \) and \( Z_5 \) )and the estimated points \( Z_0 \) are all smaller than \( 72^\circ \), so \( Z_1 \) occludes \( Z_2 \) and \( Z_5 \). \( Z_2 \) occludes \( Z_5 \). \( Z_3 \) and \( Z_4 \) have no occlusion relations with other sample points.

In addition, the angle sum \( \alpha \) between two groups of sample points (\( Z_1 \) and \( Z_2, Z_1 \) and \( Z_5 \)) and the estimated points are less than \( 72^\circ \), so the distribution of sample points \( Z_1 \) is dense, while the distribution of other sample points is sparse. In figure (c), all known sample points are clustered on one side, and angle \( \beta \) between any two sample points and the estimated point is less than \( 72^\circ \), so there is a blocking relationship between all sample points. In addition, the angle

![Image](image-url)
sum $\alpha$ between the sample point ($Z_1, Z_4, Z_5$) and two adjacent sample points on both sides and the estimated point is less than 72°, so their distribution is dense.

However, in figure (c), this sample point is less likely to be gathering on one side. The general distribution of sample points is shown in figure (a) and figure (b).

**B. METHOD EXPRESSION**

According to the discussion of the basic hypothesis, compared with the traditional IDW algorithm, this chapter proposes an adaptive method to allocate the weight index based on the azimuth relation between the estimated point and the known sample point. The expression of the method is shown in formula 1

$$Z_0 = \frac{\sum_{i=1}^{n} \frac{t_i}{d_i^2} Z_i}{\sum_{i=1}^{n} \frac{t_i}{d_i^2}}$$

In formula, $Z_0$ is an estimated point, $Z_i$ is the known sample point. $d_i$ is the distance between the known sample point ith and the estimated point. $t_i$ is the harmonic weight coefficient of the blocking relation between the known sample points. $k_i$ is the harmonic coefficient of the weight index. The expression of $t_i$ is shown in formula 2:

$$t_i = \begin{cases} 1 & i = 1 \\ \prod_{j=1}^{i-1} \sin^k \theta_{ij} & i = 2, 3, \ldots, n \end{cases}$$

In formula 2, $k_i$ is the harmonic coefficient of the weight index, and $i, j$ represents the known sample points ith and jth. The sequence number of the sample points is sorted according to the distance to the estimated point $Z_0$. It is that the first known sample point is the closest to the estimation point, which means that no sample point covers the first sample point, when $t_1 = 1$, the sample point nth is the most distant from the estimation point. $\theta_{ij}$ represents the angle between the line connecting of the second sample points $(i, j)$ and the midline of crossing the estimation point (taking an acute Angle or right Angle), and $\sin^k \theta_{ij}$ represents the expression of the degree to which the sample point ith is blocked by the sample point ith, as shown in formula 3:

$$\sin^k \theta_{ij} = \begin{cases} 1 & \beta_{ij} > 360°/n \\ \left(1 - \cos^2 \theta_{ij}\right)^{k/2} & \beta_{ij} \leq 360°/n \end{cases}$$

In formula 3, $k_i$ is the harmonic coefficient of the weight index, $\beta_{ij}$ represents the Angle between the $P_i$ and the $P_j$ known sample point and the estimated point. According to the basic hypothesis, when $\beta_{ij} > 360°/n$ (”n” is the number of known sample points), there is no occlusion between the $P_i$ and the $P_j$ known sample points. Otherwise, there is occlusion.

In order to clearly illustrate the relationship between the $\sin^k \theta_{ij}$ and the degree of shielding, according to $\sin^k \theta_{ij}$ varying with the position of the sample point, the degree of shielding is illustrated in figure 2. $Z_0$ represents the estimated point, $Z_i$ and $Z_j$ represents the $P_i$ and $P_j$ known sample points.

Fixed $Z_j$, the distance of $Z_i$ and $Z_0$ was gradually shorten. In this process, it is easy to see intuitively that the degree of blocking of $Z_j$ to $Z_i$ gradually decreases. From the details, the Angle between the line connecting of the two sample points and the median line of crossing the estimation point gradually increases. According to formula 2, the harmonic weight coefficient ($t_i$) reflecting the occlusion relationship gradually increases, so the degree of occlusion gradually decreases. Until the position of $Z_i, Z_j$ and $Z_0$ into an isosceles triangle, at the same time, $\theta = 90°, t_i = 1$, the occlusion relationship $Z_i$ to $Z_j$ is completely eliminated. It can be seen from this process that if two known sample points have occlusion relations, $\theta$ fall in between 0° and 90°, the occlusion relationship will gradually change with the change of $\theta$.

The harmonic coefficients of weight index ($k_i$) in formula 1, 2 and 3 are all the same, and their expressions are shown in formula 4:

$$k_i = \begin{cases} 2 & \alpha_1 + \alpha_2 \leq 360°/n \\ 3 & \alpha_1 + \alpha_2 > 360°/n \land \alpha_1 < 360°/n \land \alpha_2 < 360°/n \\ 4 & \alpha_1 + \alpha_2 > 360°/n \land \alpha_1 \geq 360°/n \lor \alpha_2 \geq 360°/n \\ 5 & \alpha_1 + \alpha_2 > 360°/n \land \alpha_1 > 360°/n \land \alpha_2 > 360°/n \end{cases}$$

In formula 4, $\alpha_1, \alpha_2$ represents the angle between the known sample point ith and the line connecting the two adjacent sample points with the estimated point. According to the literature [20], the weight index should be improved in sparse sample points to obtain more accurate interpolation accuracy. According to the explanation of the basic hypothesis, it is stipulated that when the angle sum between the line connecting of the known sample point and its two adjacent sample points or the estimated point is $\alpha_1 + \alpha_2 \leq 360°/n$, the sample point distribution is dense and its weight index harmonic coefficient is $k_i = 2$. Otherwise, when $\alpha_1 + \alpha_2 > 360°/n$, the sample points are sparse. In this case, according to the different sparsity, the harmonic coefficient of the weight index is also different.

**FIGURE 2.** The illustration of the sample point changes in the degree of occlusion with $\sin^k \theta_{ij}$. [Diagram of sample point distribution and angle]
Z_10 is blocked by Z_1, Z_8 and Z_3 is blocked by Z_2, Z_7 and Z_8 is blocked by Z_3. For the weight index, only the sum of the Angle between Z_8 and Z_2, Z_8 and Z_3 and the estimated points is less than \( \beta \), while all other sample points are greater than \( \beta \). In conclusion, according to formula 1 and 4, the calculation formula of the estimated point \( Z_0 \) is (5), as shown at the bottom of the next page.

In formula 5, \( d_1 \) represents the distance between each sample point and estimated point, and \( \theta \) represents the angle between the line connecting of two sample points and the midline of crossing the estimated point. After processing with this method, the sample points occluded completely are removed to make the distribution of the sample points more uniform. At the same time, the sparseness degree of the distribution of the sample points can also be judged to make the weight index self-adaptive configuration.

### C. PROXY MODEL BASED ON ROBOT BEHAVIOR

It is not realistic to obtain the difference \( V_d(c) \) between all the simulation behaviors and real robot behavior. Therefore, the proxy model technique is used to approximate the behavior difference \( V_d(c) \), according to the limited real behavior of difference, the difference of the unknown behavior was estimated approximately. There is no need to conduct a large number of transplantation evaluation experiments on the real robot, so as to improve the efficiency of experiments. Based on the improved the inverse distance weighting method proposed, a proxy model of the difference between simulation behavior and the behavior of real robot can be constructed, which not only guarantees the experimental efficiency but also improves the accuracy of the model. The agent model based on robot behavior is described below.

It is defined that the aggregation of a set of controllers of the robot is represented with \( \phi \) during the simulation, and \( \phi_T \) represents a set of controllers that have been migrated. The difference value \( V_d^\phi(c)|c \in \phi \) between the simulation and the real robot behavior to be estimated is regarded as the estimation point and the real behavior difference value \( V_d(c)|c \in \phi_T \) transplanted is regarded as the known sample point. The difference of behavior is caused by the difference of individual robot controller. Therefore, the distance between the known sample point and the estimated point is defined as the difference between the controllers. The difference between controllers can be calibrated by several behavior indexes \( b(c) = \{b_1, b_1, \ldots, b_n\} \) in simulation. For example, the fitness function value based on the controller in simulations and the centroid of the robot can be taken as the behavior indexes in mobile tasks. \( b(c) \) is defined as \( \{\text{fitness, center}\} \). Therefore, the difference between any two controllers c and c’ can be compared numerically by behavioral indicators, as shown in formula 6:

\[
d(c, c') = \|b(c) - b(c')\|
\]  

(6)

To sum up, combined with formula 6 of the improved the inverse distance weighting method, this agent model of a difference value between simulation and real robot behavior...
is established, as shown in formula 7:

$$V_d^*(c) \mid_{c \in \varphi} = \max \left( \frac{\sum_{c_i \in \varphi} I_i}{\sum_{c_i \in \varphi} d(c_i, c)} \right)$$

(7)

According to formula 7, the unknown behavior difference value can be estimated approximately according to the difference between the finite and accurate simulation and the behavior of the real robot, so as to improve the efficiency of the experiment. Compared with the traditional inverse distance weighting method, this proxy model is based on an improved inverse distance weighting method, the accuracy of interpolation is higher. In addition, according to the standard of transplantation, at most one controller is transplanted to the real robot in the course of evolution of every generation population. If it meets the conditions of transplantation, it would produce a new difference value between accurate simulation

$$Z_0 = \frac{Z_1}{d_1^2} + \frac{Z_2}{d_2^2} + \frac{Z_3 \sin^2 \theta_{32}}{d_3^2} + \frac{Z_4}{d_3^2} + \frac{Z_5}{d_5^2} + \frac{Z_6}{d_5^2} + \frac{Z_7 \sin^2 \theta_{10}}{d_5^2} + \frac{Z_8 \sin^2 \theta_{18} \sin^2 \theta_{38}}{d_5^2} + \frac{Z_9 \sin^2 \theta_{13}}{d_6^2} + \frac{Z_10 \sin^2 \theta_{10}}{d_{10}^2}$$

(5)
and real robot behavior. As a new known sample point, it is added to the calculation of the proxy model, so that the proxy model can be modified continuously and the accuracy can be improved gradually in the whole optimization process.

Combined with the discussion in this section, the calculation flow chart of the difference value agent model between simulation and real robot behavior based on inverse distance weight method is drawn, as shown in figure 7. The symbols of the formulas in the figure have the same meanings as the symbols mentioned in this section.

III. TRANSPLANTATION METHOD

The optimization goal of transplantation method is fitness function value and portability, in the process of evolution to evaluate a robot individually. First of all, based on the given task, the robot individuals with good fitness function values were found. Then the proxy model is used to find the individual robot whose behavior difference value between simulation and a real robot is smaller, in other words, the individual possesses good portability.

In addition, in the process of evolution assessment, the number of transplantations should be reduced to further improve the experimental efficiency. Discussed in literature [22], [23], in order to keep the behavior of robot individual species diversity and decrease the number of transplantations, and we can set up a threshold value $\delta$ that it is the behavior difference value between simulation and the real robot. Only when the difference value of behavior between the minimum approximation simulation and the real robot is greater than this threshold value, it meets the expression $\text{min}[V^*_d(c)] \geq \delta$, it can be allowed to transplant the robot individual to the real robot. If it does not meet this condition, it cannot be allowed to transplant. The threshold value is set based on human experience, it is one or two percent of the transplant number of targets.

This chapter still uses the heterogeneous modular robot evolution designer (Edmor system [24]) integrated with JEAR (Java Evolutionary Algorithm Framework) [25] and V-REP (Virtual Robot Experimentation Platform) [26] emulator as the evolution Platform. Using EMeRGE modular robot as the implementation robot [27]. The coding method, the simulation model, and the evaluation method are the same as those described before. However, due to the inclusion of the portability evaluation index, the evolutionary strategy needs to be adjusted.

There exists a trade-off relationship between fitness function value and portability. Therefore, the optimal solution based on the evolution of the transplantation method is not necessarily the highest fitness function value, but the compromise and tradeoff between fitness function value and portability, a robot individual of the better fitness function value and portability is the optimal solution. Therefore, the selection operator of evolutionary strategy is adjusted, instead of taking a single fitness function as the selection criterion for the optimal individual, portability was added as the second selection index. After the adjustment, the selection operator is that the bottom 10% of the individuals in the population were removed, and the robot individuals generated randomly with a small number of modules and robot individuals with better fitness function value and portability were supplemented.

A. ALGORITHM PROCESS

Different from the traditional evolution method in simulation, the evolution evaluation process of the transplantation method is a way of evolution of simulation combined with the real robot: In other words, the fitness function value of each robot individual and the behavior difference value between the simulation and the real robot based on the proxy model are obtained through the evolution of the individual robots of each generation in the simulation environment. According to the portability standard, the minimum behavior difference was selected to compare with the threshold value, the real robot behavior can be obtained by transferring the robot individual meeting condition to a real robot for evaluation. Based on this, the behavior difference between the new accurate simulation and the real robot is calculated, which is added to the proxy model as a new known sample point to update and modify the proxy model. Then, according to the evolutionary strategy, the evolution of the next-generation robot population was started and the process was repeated until the end of evolution. The specific algorithm process is shown as follows:

(1) After the evolution parameters are configured, the individual robots of the first-generation population are evolved based on the given task in the simulation environment, and the fitness function value of each individual robot is obtained. At least three different robot individuals were randomly selected and transplanted to the real robot, and the difference value between their accurate simulation and the behavior of the real robot was calculated as the initial sample point of the agent model.

(2) We can carry out the evolution of the second-generation robot population in the simulation environment, and obtain the fitness function value and behavior performance of each robot individual. According to the agent model, the difference value between the approximate simulation of each robot individual and the behavior of the physical robot is calculated, then, we choose the smallest behavior difference among them to compare with the threshold. An individual robot that meets the transplant criteria $\text{min}[V^*_d(c)] > \delta$ is transplanted to a real robot for evaluation to evaluate. Based on this, the behavior difference between the new accurate simulation and the real robot is calculated, which is added to the proxy model as a new known sample point to update and modify the agent model.

(3) the evolution of the next-generation robot starts in the simulation environment according to the evolutionary configuration parameters. Repeat steps (2) and (3) until evolution is complete.

The algorithm flow chart of the transplantation method is shown in figure 5. It can be seen from the transplantation
process that this method does not need strict preconditions and is widely applicable, compared with other methods to solve the “reality gap”.

B. CASE STUDY
The example task is defined as the distance that the modular robot can walk in a straight line on a plane within 30 seconds. Simulation and the real robot behavior difference is the difference between the moving distances of the two. “S” is used to indicate the distance that the modular robot moves from the origin in the simulation, and R is the distance that the modular robot moves from the origin in practice, the normalized Mean square error (nMSE) is used to represent the difference between the exact simulation and the behavior of the physical robot, as shown in formula 8:

\[ V_d(c) = \sum_{t=1}^{T} \frac{(S_t - R_t)^2}{S \cdot R} \]  

(8)

In the formula, \( \bar{S}, \bar{R} \) represents the mean value of “S” and “R”, respectively, “t” represents the evaluation time, and “T” is the set evaluation time, “c” represents the controller after the evolution of the individual robot.

The difference in the behavior of any two individual robots is the difference between their controllers, which can be calibrated by the behavioral indicators in the simulation. There are also differences in behavioral indicators for different execution tasks. This example is a straight-line movement task based on a plane. Obviously, the moving distance and the change of the centroid change of the modular robot in the moving process are important characteristics that reflect its behavior. Therefore, the moving distance and the centroid height of the robot are defined as behavioral characteristic parameters, namely \( b(c) = \{ \text{fitness}, \text{center} \} \). Based on the behavior characteristic parameters, the behavior difference of any two controllers can be obtained, as shown in formula 8. Combined with the precise behavior difference defined by formula 7, an agent model of behavior difference between simulation and a real robot can be established. Then the transplantation method is used to evolve the modular robots composed of five different modules on the evolution platform, in order to find out the modular robot special individuals with longer moving distance and better portability under each type of form.

IV. THE EXPERIMENTAL RESULTS
In section III-A, the configuration problem of evolutionary parameters is analyzed, that is, in the case of approximate convergence, based on the trade-off between the accuracy of evolutionary results and the number of iterations is determined according to prior experience and evolutionary goals. Therefore, the configuration parameters of the evolutionary algorithm in this chapter are also set based on the above principles: The evaluation time is the 30s, the population number is 24, the maximum number of modules is 9, the operation times of each type of module form is 20, and the threshold value of behavior difference is \( \delta = 0.15 \) (the transplantation times is 15 and the weight coefficient is 1%). When the evolution process reaches the target number of transplants, evolution stops.

Figure 6 shows the fitness function values of the simulated modular robot and the corresponding physical robot, based on the optimal solution of the planar linear movement task, constructed by five different types of module forms under 20 different evolutionary runs. Figure (a) is the value obtained based on an improved inverse distance weighting method, figure (b) is the value obtained based on the inverse distance weighting method. The vertical axis represents the moving distance, and the horizontal axis represents the modular robots constructed by five types of module forms. Because the data is not normally distributed, the mean and standard deviation cannot be used to plot the error graph of the two. Therefore, the error plots of the median and the first quantile (25%) and the third quantile (75%) of the fitness function values of the robot based on the optimal solution of 20 independent evolutions were drawn. Table 1 shows the simulation of the optimal robot constructed by five types of modules and the fitness function value (the moving distance) of the
FIGURE 7. Examples of video screenshot of five best physical robot assembled by Type 1 to Type 5 modules for locomotion in the simulation and reality at \( t = 5s, t = 10s, t = 15s, t = 30s \) moment.

physical robot based on the improved inverse distance weight method and inverse distance weight method.

It can be seen from the results of figure 9 and table 1, compared with the conventional inverse distance weighting method, a transplantation method based on the improved inverse distance weighting method. The fitness function value of real robot is closer to the fitness function value of its corresponding simulation, overcoming a "reality gap" problem well, narrowing the performance difference about the modular robot simulation and reality, improving the portability of modular robots.

Figure 7 respectively shows the time when \( t = 5s, t = 10s, t = 15s, \) and \( t = 30s \), the motion video screenshots of the optimal individual robots based on the evolution of the five types of modules in simulation and practice, that is, a robotic feature with good fitness function values and portability. A yellow scale with a length of two meters is placed in the figure as a reference. It can be seen from the motion video screenshot that the action behavior of the robot in the simulation is close to that of the corresponding physical robot, which proves once again that the migration method not only reduces the difference between the fitness function values of the simulation and the physical robot but also reduces the difference in action behavior between the two and indicates that this transplantation method is effective.

In addition, in order to verify that the improved inverse distance weighting method has higher interpolation precision than the traditional inverse distance weighting method. These two methods are used to establish their respective proxy models based on the behavior data generated by the above-mentioned 20 evolutionary processes, respectively to compare the approximate behavior difference value and precise behavior difference value, it is that the different degree between the estimated value and real value. During the 20 evolutionary processes with the transplantation method, the inverse distance weighting method transplanted a total
TABLE 1. Display the simulation and reality fitness of the best robots assembled by modules from type 1 to type 5 for 20 independent evolutionary runs for locomotion.

| Type   | Moving distance (unit: mm) | Improved inverse distance weighting method | Inverse distance weighting method |
|--------|---------------------------|--------------------------------------------|----------------------------------|
|        | Simulation value          | 1370                                       | 1190                             |
| one    | Actual value              | 1150                                       | 810                              |
|        | Simulation value          | 1210                                       | 1030                             |
| two    | Actual value              | 1100                                       | 820                              |
|        | Simulation value          | 1420                                       | 1120                             |
| three  | Actual value              | 1200                                       | 830                              |
| four   | Simulation value          | 2010                                       | 1810                             |
|        | Actual value              | 1810                                       | 1350                             |
| five   | Simulation value          | 1110                                       | 1000                             |
|        | Actual value              | 1020                                       | 800                              |

of 300 robot individuals (15 transplants per run, $\delta = 0.15$), and the improved inverse distance weighting method transplanted 300 robot individuals (15 transplants per run, $\delta = 0.15$). The interpolation of the inverse distance weighting method is drawn by ArcGIS Pro, which is used to analyze and predict values related to spatial or spatiotemporal phenomena, and the improved inverse distance weighting method interpolation program is completed by python.

The experimental results are shown in Fig. 8, Fig. (a) is the calculation result based on the inverse distance weighting method, and Fig. (b) is the calculation result based on the improved inverse distance weighting method. The vertical axis $Y$ represents the approximate behavior difference value of the estimate, and the horizontal axis $X$ represents the exact behavior difference value. The dotted line represents the fitting curve of the interpolation points, while the solid line represents the fitting curve of the exact value, and the more the dotted line trend coincides with the solid line, the better the interpolation effect is. It can be seen from the figure that the trend of the dotted line of the improved inverse distance weighting method is closer to the solid line, and the interpolation effect and accuracy are better than that of the inverse distance weighting method. In addition, the discrete points of the difference value in behavior are mainly concentrated in $[1 \sim 3]$, which also indirectly indicates that the behavior difference between simulation and the physical robot is narrowed. Therefore, compared with the inverse distance weighting method, the improved inverse distance transplantation method improves the portability of the robot individual and the accuracy of the agent model.

V. CONCLUSION

The improved inverse distance weighting method based on the azimuth relation of the sample is proposed in this chapter, which can make the sample points evenly distributed and can also configure the weight index adaptively, and improve the efficiency and accuracy of interpolation. Then the agent model of the difference between robot simulation and entity behavior is established. Based on agent model technology, the algorithm flow of the migration method is studied. Finally, based on the plane moving task, the portability of the modular robot consisting of EMERGE modules of different shapes and types was experimentally studied, and compared with the transplantation method based on the inverse distance weight transplantation method. The experimental results of
simulation and the physical robot show that this method is easy to use and applies widely, the information of other real robots can be predicted according to the limited evolutionary data of real robots, we don’t have to do all the evaluation experiments on a physical robot, it can effectively reduce the performance difference between the results of simulation and the physical robot caused by “reality gap” problem, and improves the portability of modular robots.

REFERENCES

[1] D. Floreano and F. Mondada, “Evolutionary neurocontrollers for autonomous mobile robots,” Neural Netw., vol. 11, nos. 7–8, pp. 1461–1478, Oct. 1998.

[2] G. S. Hornby, S. Takamura, J. Yokono, O. Hangata, T. Yamamoto, and M. Fujita, “Evolving robust gaits with AIBO,” in Proc. IEEE ICRA, vol. 3, Apr. 2002, pp. 3040–3045.

[3] V. Zykov, J. Bongard, and H. Lipson, “Evolving dynamic gaits on a physical robot,” in Proc. GECCO, vol. 4, 2004.

[4] C. Comaunders, E. Golden, R. D. White, and J. Rife, “Experimental verification of soft-robot gaits evolved using a lumped dynamic model,” Robotica, vol. 1, no. 1, pp. 1–8, 2006.

[5] N. Jakobi, “Evolutionary robotics and the radical envelope-of-noise hypothesis,” Adapt. Behav., vol. 6, no. 2, p. 325, 1997.

[6] N. Jakobi, “Running across the reality gap: Octopod locomotion evolved in a minimal simulation,” in Proc. Eur. Workshop Evol. Robot. Berlin, Germany: Springer, 1998, pp. 39–58.

[7] N. Jakobi, “Minimal simulations for evolutionary robotics,” Ph.D. dissertation, Univ. Sussex, Brighton, U.K., 1998.

[8] O. Miglino, H. H. Lund, and S. Nolfi, “Evolving mobile robots in simulated and real environments,” Artif. Life, vol. 2, no. 4, pp. 417–434, Jul. 1995.

[9] A. Thompson, P. Layzell, and R. S. Zebulum, “Explorations in design space: Unconventional electronics design through artificial evolution,” IEEE Trans. Evol. Comput., vol. 3, no. 3, pp. 167–196, Sep. 1999.

[10] D. Floreano and J. Urzelai, “Evolution of plastic control networks,” Auton. Robots, vol. 11, no. 3, pp. 311–317, 2001.

[11] C. Hartland and N. Bredeche, “Evolutionary robotics, anticipation and the reality gap,” in Proc. ROBIO, 2006, pp. 1640–1645.

[12] S. Koos, J.-B. Mouret, and S. Doncieux, “The transferability approach: Crossing the reality gap in evolutionary robotics,” IEEE Trans. Evol. Comput., vol. 17, no. 1, pp. 122–145, Feb. 2013.

[13] J. Bongard and H. Lipson, “Once more unto the breach: Co-evolving a robot and its simulator,” in Proceedings of Artificial Life IX, Cambridge, MA, USA: MIT Press, 2004, p. 57.

[14] J. C. Zagal and J. Ruiz-del-Solar, “Combining simulation and reality in evolutionary robotics,” J. Intell. Robot. Syst., vol. 50, no. 1, pp. 19–39, Aug. 2007.

[15] S. Koos, J.-B. Mouret, and S. Doncieux, “Automatic system identification based on coevolution of models and tests,” in Proc. IEEE Congr. Evol. Comput., May 2009, pp. 119–126.

[16] T. Hemker, H. Sakamoto, M. Stelzer, and O. von Stryk, “Hardware-in-the-loop optimization of the walking speed of a humanoid robot,” in Proc. CLAWAR, 2006, pp. 614–623.

[17] P. Abbeel, A. Coates, T. Hunter, and A. Y. Ng, “Autonomous automation of an RC helicopter,” in Experimental Robotics. Berlin, Germany: Springer, 2009, pp. 385–394.

[18] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng, “An application of reinforcement learning to aerobatic helicopter flight,” in Proc. NIPS, 2007, pp. 1–8.

[19] L. Huaxing, “Research on DEM error model,” Nanjing Normal Univ., Nanjing, China, Tech. Rep., 2008.

[20] L. Zhangquan and W. Xiaoxiang, “Inverse distance weighted interpolation involving position shading,” Acta Geodaetica Cartographica Sinica, vol. 44, no. 1, pp. 91–98, 2015.

[21] W. Yong, L. Chaokui, C. Liang, and Z. Shuangning, “Analysis on impact of weight to spatial interpolation methods,” J. Hunan Univ. Sci. Technol., Natural Sci. Ed., vol. 23, no. 4, pp. 77–80, 2008.

[22] S. Doncieux and J.-B. Mouret, “Behavioral diversity measures for evolutionary robotics,” in Proc. IEEE Congr. Evol. Comput. (CEC), Jul. 2010, pp. 1–8.

[23] J.-B. Mouret and S. Doncieux, “Using behavioral exploration objectives to solve deceptive problems in neuro-evolution,” in Proc. 11th Annu. Conf. Genet. Evol. Comput. (GECCO), 2009, pp. 627–634.

[24] A. Faiha, F. Bellas, F. Lópex-Peña, and R. J. Duro, “EDHMoR: Evolutionary designer of heterogeneous modular robots,” Eng. Appl. Artif. Intell., vol. 26, no. 10, pp. 2420–2423, Nov. 2013.

[25] P. Caamaño, R. Tédin, A. Paz-Lopez, and J. A. Becerra, “JAEA: A java evolutionary algorithm framework,” in Proc. IEEE Congr. Evol. Comput., Jul. 2010, pp. 1–8.

[26] E. Rohmer, S. P. N. Singh, and M. Freese, “V-REP: A versatile and scalable robot simulation framework,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Nov. 2013, pp. 1321–1326.

[27] R. Moreno, C. Liu, A. Faina, H. Hernandez, and J. Gomez, “The EMeRGE modular robot, an open platform for quick testing of evolved robot morphologies,” in Proc. Genet. Evol. Comput. Conf. Companion, Jul. 2017, pp. 71–72.