



PEARL

BAS Optimized ELM for KUKA iiwa Robot Learning

Li, Chunxu; Zhu, Shuo; Sun, Zhongbo; Rogers, James

Published in:

IEEE Transactions on Circuits and Systems II: Express Briefs

DOI:

[10.1109/tcsii.2020.3034771](https://doi.org/10.1109/tcsii.2020.3034771)

Publication date:

2020

Link:

[Link to publication in PEARL](#)

Citation for published version (APA):

Li, C., Zhu, S., Sun, Z., & Rogers, J. (2020). BAS Optimized ELM for KUKA iiwa Robot Learning. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 0(0), 1-1. <https://doi.org/10.1109/tcsii.2020.3034771>

All content in PEARL is protected by copyright law. Author manuscripts are made available in accordance with publisher policies. Wherever possible please cite the published version using the details provided on the item record or document. In the absence of an open licence (e.g. Creative Commons), permissions for further reuse of content should be sought from the publisher or author.

BAS Optimized ELM for KUKA iiwa Robot Learning

Chunxu Li^{1,2*}, Shuo Zhu¹, Zhongbo Sun³, James Rogers¹

Abstract—In this paper, an enhanced robotic learning interface has been investigated using Beetle Antennae Search (BAS) and Extreme Learning Machine (ELM). The initial values of learning weights and bias of the network have significant effect on the performance of the ELM, hence, BAS algorithm was employed to optimize the initial values of learning weights and bias. Kinect v2 camera sensor was applied to obtain the endpoint's position of the upper limb, MYO armband was used to measure the corresponding joint angle values. Those aforementioned data formed the dataset to be trained by ELM and after training the ELM model was able to generate angle values by only giving position as input without a need to carry out kinematic calculations. The proposed method has been validated by conducting series of experimental studies on a KUKA iiwa robot.

Index Terms—Extreme learning machine, Beetle Antennae Search, MYO Armband, Kinect v2, KUKA iiwa robot.

I. INTRODUCTION

THE field of research with respect to machine learning attracted great attention during the past decade. Machine learning has been used in many fields, such as autopilot filtering system of email, the recommendation system of e-shopping, handwriting recognition in the post office and automatic driving system of the vehicle. The research value of machine learning has attracted a growing number of institutions, companies as well as researchers devoting themselves into the field. Google announced its driverless car project in 2010, and released a video, wherein its one of the employees Steve Mahan who lost 95 percentage of the vision, safely drove 12 miles [1]. Wherein, ELM is an essential method employed in many domains already, which is developed from single-hidden layer feedforward neural networks (SLFNs) [2]. Hence, a growing number of researchers focus on Extreme Learning Machine in-depth and extensive research. Ding et al proposed optimization Extreme Learning Machine (OM-ELM) method of the SMO, however, owing to the large data set optimal parameters C , the algorithm requires more iterations to converge to the optimal solution of the optimization problem [3]. Huang et al., describes the enhanced incremental Extreme Learning Machine (EI-ELM) and convex incremental Extreme Learning Machine (CI-ELM) [4]. Rong et al describes learning machine algorithm based on fuzzy limits [5]. Li et al proposed a comprehensive complex Extreme Learning Machine (FC-ELM) [6]. A research team also describes a wind power pre-side model of wavelet decomposition and Extreme Learning

Machine to verify that this method has higher prediction accuracy [7]. A scientist added a class classification of neurons to form new network structure in the fermentation process Nosiheptide, in order to realize a better line estimation [8]. However, using the traditional ELM model alone to make predictions often has limitations in terms of prediction accuracy.

In recent years, intelligent algorithms such as neural network (NN), genetic algorithm (GA) and particle swarm optimization (PSO) were applied to solve the aforementioned optimization problems. NN has the capabilities of adaptive learning, which can deal with parallel distributed processing. GA is often combined with polynomial interpolation for robot motion planning tasks, with low risk of falling into local optimal solutions [9]. However, GA is with complex proceedings which needs to go through the encoding, decoding, selection, crossover and mutation processes. This causes its operation time being difficult to be guaranteed at all time. The PSO algorithm is with a simple structure and its parameters are easy to adjust. Compared to GAs, PSO does not need to perform for convex problems and the optimal solution can be found with faster convergence [10]. However, the PSO needs to randomly generate plenty of particles when the next populations are generated, which decreases the convergence rate. To break through the above-mentioned limits, BAS was selected to perform the optimization of ELM in this paper, which is a new type of computational intelligence algorithms developed in [11]. The main contribution of this brief paper is investigating an ELM model to output angle values while inputting the endpoint position in Cartesian space for robot control, which is much faster while learning than traditional artificial neural network models and can be used to replace the complicated kinematics calculation; developing a strategy to create home-made dataset of joint values - endpoint positions; employing BAS algorithm to optimize the initial parameters of ELM to maximum the learning accuracy.

II. DATASET ACQUISITION

In this paper, the endpoint position was directly collected using Kinect SDK 2.0. Kinect v2 produced by Microsoft, is an RGB-D device, which can capture depth, colour, and IR images (also sound). Using the SDK, captured colour and depth information can be consolidated (transformed) into real-world co-ordinates, called Camera Space; these co-ordinates are referenced to the centre of the depth sensor [12]. Skeletal tracking can also be achieved with the use of the Kinect 2.0 SDK. Both the colour image, and depth image transformed into a common frame, the origin of which is located at the centre of

*Corresponding author: C. Li. Email: chunxu.li@plymouth.ac.uk

¹Center for Robotics and Neural Systems, Plymouth University, UK, PL4 8AA. ²College of Automation and Electronic Engineering, Qingdao University of Science and Technology, Qingdao, China. ³Department of Control Engineering, Changchun University of Technology, Changchun, China.

the depth camera, called Camera space; the co-ordinate system of this space follows a right-handed convention.

By calculating the joint angles of the human arm joints to capture the operator's arm posture, the control of the remote mechanical arm is realized. The human arm includes 7 degrees of freedom (DoFs), among which, the shoulder joint has 3 DoFs, the elbow joint has 2 DoFs, and the wrist joint has 2 DoFs. It is assumed that the shoulder and elbow joints of the human arm are 5 DoFs chain manipulators. The modelling method of the upper limb using the standard DH parameter can be found from our previous research [12]. In order to calculate the joint angles of the operator's elbow and shoulder joints (show in Fig.1), the operator will wear pair of MYO Armbands on the upper arm and lower arm both, and define the global coordinate system (x_g, y_g, z_g) , where: Z axis is up, Y-axis is forward, X is perpendicular to Z-axis and Y-axis, pointing to the human body.

On the upper and lower arms of the human body, separate coordinate systems are established to describe the spatial posture of the arms. The coordinate system (x_u, y_u, z_u) is established on the upper arm coordinate system, and (x_l, y_l, z_l) is the lower arm coordinate system. We assume that initially the upper frame and the lower arm frame coincides with the global frame. Define the initial posture of the upper arm and lower arm as [13]:

$$R_{gl}^i = [x_{gl}^i \ y_{gl}^i \ z_{gl}^i], \quad R_{gu}^i = [x_{gu}^i \ y_{gu}^i \ z_{gu}^i] \quad (1)$$

where R represents the rotation matrix, the superscript i represents the initial position, the subscript g represents the global coordinate system, and u and l represent the coordinate system of the upper arm and the lower arm, respectively. The rotation matrix $R_{AB} = [x_{AB} \ y_{AB} \ z_{AB}]$ represents the spatial posture of the coordinate system A in the coordinate system B. The vectors x_{AB} , y_{AB} and z_{AB} represent the position description of the coordinate axis of the coordinate system B in the coordinate system A. The posture of the

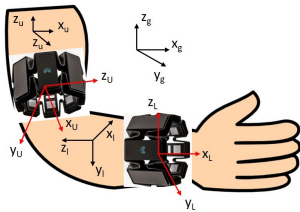


Fig. 1. Schematic diagram of global coordinate system, local coordinate system and MYO Armband coordinate system

operator's upper arm in the coordinate system of the first MYO Armband (worn on the upper arm) and the lower arm in the coordinate system of the second MYO Armband (worn on the lower arm) can be expressed as [13]:

$$R_{Uu}^i = (R_{gU}^i)^T R_{gu}^i, \quad R_{Ll}^i = (R_{gL}^i)^T R_{gl}^i \quad (2)$$

where the subscript U represents the coordinate system of the MYO Armband worn on the upper arm, and the subscript L represents the coordinate system of the MYO Armband worn on the lower arm. When the operator moves the arm to a new

position, the posture of the operator's upper arm and lower arm relative to the global coordinate system can be expressed as [13]:

$$R_{gu}^n = (R_{gU}^n)^T R_{Uu}^i, \quad R_{gl}^n = (R_{gL}^n)^T R_{Ll}^i \quad (3)$$

where the superscript n represents the new posture reached by the operator's arm.

If we first rotate a coordinate system around the X axis of a second known coincident coordinate system by an angle of γ , then rotate around the Y axis by an angle of β , and finally rotate the coordinate system by an angle of α around the Z axis. Here the rotation angles γ , β and α are defined as roll angle, pitch angle and yaw angle respectively. Therefore, the rotation matrix can be written as (c for cos, s for sin) [13]:

$$R_{gu}^n = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\gamma & c\beta s\gamma & c\beta c\gamma \end{bmatrix} \quad (4)$$

Using the MYO's gyroscope, we can obtain a quaternion ($q = xi + yj + zk + w$, where the (x,y,z) is a vector and w is a scalar quantity) at the configuration with respect to the corresponding posture, which can be used to calculate the R_{gu}^n . From the rotation matrix of the upper arm MYO Armband coordinate system, the three joint angles of the shoulder joint can be calculated as:

$$\begin{aligned} \alpha &= \text{atan2}(2(yz - wx)/s\beta, 2(wy + xz)/s\beta); \\ \beta &= \text{atan2}(\sqrt{4(xz - wy)^2 + 4(wx + yz)^2}, 1 - 2(x^2 + y^2)); \\ \gamma &= \text{atan2}(2(xz - wy)/s\beta, -2(wx + yz)/s\beta) \end{aligned} \quad (5)$$

Let us define, r_{ij} is one element of R_{gu}^n , a_{ij} is one element of R_{gl}^n , which can be obtained by the MYO's gyroscope. The two joint angles of the elbow joint can be calculated through the MYO Armband worn on the lower arm:

$$\begin{aligned} \delta &= \arccos(a_{12}r_{13} + a_{22}r_{23} + a_{32}r_{33}); \\ \epsilon &= \arccos(a_{11}r_{11} + a_{21}r_{21} + a_{31}r_{31}) \end{aligned} \quad (6)$$

III. ELM LEARNING MODEL DESIGN METHODOLOGY

In 2006, Professor Huang in Nanyang Technological University proposed limits of learning feedforward neural networks, and introduced the basic principles of ELM [3]. Compared BP Algorithm and SVM, ELM has several salient features: no parameters need to be manually tuned; most training can be completed in milliseconds; it could obtain better generalization performance than BP in most cases [14].

ELM is a neural network with only one hidden layer, which can be obtained through the offline training, wherein the weights inputted layer to the hidden layer and the bias system can be a randomly initialized, and the value between hidden layer and output layer is required in order to adjust the output of the system to minimum target error. If the endpoint position data obtained from Kinect sensor is inputted as x , then the L number of the hidden nodes with output functions can be expressed as:

$$f_L(x) = \sum_{i=1}^L \beta_i g_i(x) = \sum_{i=1}^L \beta_i G_i(x, a_i, b_i) \quad (7)$$

$$a_i \in C^d, x_i \in C^d, b_i \in C, \beta_i \in C \quad (8)$$

where $g_i(x)$ is the activation function, β_i is output weights, a_i is the input weights and b_i is the input bias.

Given notable N training samples, the output with notable L of hidden node for learning machine are:

$$f(x) = \sum_{i=1}^L \beta_i G(a_i, b_i, x) = \beta \cdot h(x) \quad (9)$$

where $h(x)$ is the output vector of the hidden layer, hidden layer node parameters $(a_i, b_i)(i=1,2,\dots,L)$ are initialized by BAS, which is able to connect the i -th hidden node with the output nodes with learning weights β_i .

By analysing, we can work out the matrix representation of the linear system, which is shown following:

$$H \cdot \beta = T \quad (10)$$

then we have:

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix} T = \begin{bmatrix} t_1^T \\ \vdots \\ t_L^T \end{bmatrix} \quad (11)$$

According to the input x_i , there is a network of hidden layer output matrix H , and the i -th row represents the hidden layer output vector. Based on the values of all the input (x_1, \dots, x_N) , the output values of the i -th column represents the i -th hidden node. Hence, we can obtain the least squares solution of linear systems:

$$|H \cdot \hat{\beta} - T| = \min_{\beta} |H \cdot \beta - T| \quad (12)$$

where

$$H(a_1, \dots, a_L; b_1, \dots, b_L; x_1, \dots, x_L) = \begin{bmatrix} G(a_1, b_1, x_1) & \cdots & G(a_L, b_L, x_1) \\ \vdots & \cdots & \vdots \\ G(a_1, b_1, x_N) & \cdots & G(a_L, b_L, x_N) \end{bmatrix}_{N \times L} \quad (13)$$

is the output of the hidden layer nodes. Then we can calculate out:

$$\hat{\beta} = H' \cdot T \quad (14)$$

where H' is the generalized pseudo inverse Jacobian matrix of H , and the least squares solution of minimum norm β is unique, which is designed to train error until minimum. Considered the ELM is supervised-learning based, hence in this paper, the endpoint position collected from Kinect is as the input of ELM, the angular joint values measured from MYO Armband is as output, to train the ELM model. Once the training has been done with satisfied accurate rate, the model will be able to direct load to estimate series of joint angles by giving endpoint position data.

IV. BAS OPTIMIZATION OF THE INITIAL LEARNING WEIGHTS AND BIAS

In this paper, BAS algorithm is employed to optimize the ELM model by finding a series of most optimized parameter set of learning weights and bias. BAS is an intelligent optimization algorithm developed in 2017 inspired by the principle of beetle foraging. Since the process of network learning and training is to adjust the learning weights and bias of each layer, the essence of network training is to continuously

adjust the weights and bias, and the initial weights and bias are randomly generated, which will have a great impact on network performance. Therefore, the optimization of ELM mainly focuses on using optimization algorithms to adjust weights and bias. Using the optimized weight threshold for network training can greatly improve the fit of the network and reduce the error. The specific method is as follows:

1. Establish a random vector of the heading of the beetle, and normalize it:

$$\vec{b} = \frac{rands(k, 1)}{\|rands(k, 1)\|} \quad (15)$$

where $rands()$ is a random function; k represents the spatial dimension. Assuming that the model structure is $N-L-l$, N is the number of nodes in the input layer, L is the number of nodes in the hidden layer, and l is the number of nodes in the output layer, the dimensional formula of the search space of the longhorn beetle search algorithm is: $k = N * L + L * l + L + l$. In this paper k is 255.

2. Create the spatial coordinates of the left and right whiskers of longhorn beetle:

$$\begin{aligned} x_{rt} &= x_t + d_0 * \vec{b} / 2; \\ x_{lt} &= x_t - d_0 * \vec{b} / 2 \end{aligned} \quad (16)$$

x_{rt} is the position coordinate of the beetle at the t -th iteration of the right beard; x_{lt} is the position coordinate of the beetle at the t -th iteration of the left beard; x_t is the centroid coordinate of the beetle at the t -th iteration; d_0 is the distance between the two beards distance.

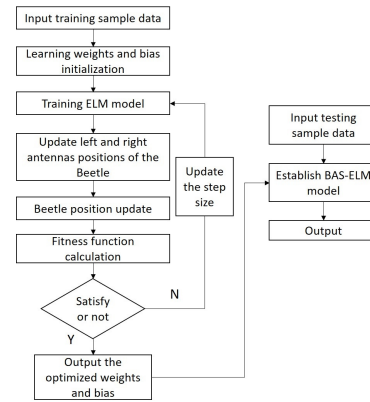


Fig. 2. Flowchart of angle joints data generation

3. Judging the odor intensity of left and right whiskers according to the fitness function, that is, the intensity of $f(x_l)$ and $f(x_r)$, where f is the fitness function. This paper uses the mean square error (MSE) of the test data as the fitness function.

4. Iteratively update the location of longhorn beetle:

$$x_{t+1} = x_t - \delta_t * \vec{b} * \text{sign}(f(x_l) - f(x_r)) \quad (17)$$

where δ_t is the step factor of the t -th iteration; $\text{sign}()$ is the sign function. The setting of the step factor in BAS adopts a linear decreasing method for trial and error. First, determine the initial step size $\delta_0=1$, the number of iterations $n=100$, and

the attenuation coefficient η is a number close to 1 between $[0,1]$. The formula is:

$$\delta_t = \delta_{t-1} * \eta \quad \eta = 1 - 5/n \quad (18)$$

5. Initialization of longhorn position. Select the random number between $[-1,1]$ as the initial position of the long beetle, and save this initial position as $Variable_1$. Solve the fitness function value corresponding to the left and right sides of the beetle in the initial position and save it in $Variable_2$.

6. After calculating the positions of the left and right beards according to (17), the fitness function values corresponding to the left and right beards are calculated respectively. After comparing the sizes, update the positions of the beetles according to (18), that is the weight and bias are adjusted, and the fitness function value at the new position is obtained after training ELM. If the function value at this time is smaller, $Variable_1$ and $Variable_2$ are updated.

7. When the MSE function value is lower than 0.1 and no longer decreases or the iteration reaches the maximum number of times 100, the iteration stops and the optimal solution is outputted. If the condition is not met, continue to return to step 6. The whole optimization process is shown in Fig.2.

V. EXPERIMENTAL STUDIES

In this section, we conducted a three-step experimental study to test the effectiveness of our proposed methodologies. Kinect SDK 2.0 for windows, Visual studio 2019 and OpenCV library are used for interaction design. MYO Armband with its SDK is used for the calculation of arm joints. The experiment environment is an indoor and adequate illumination environment. There are only 5 joints are selected in the experimental studies, which are *ShoulderPitch*, *ShoulderRoll*, *ShoulderYaw*, *ElbowPitch* and *ElbowRoll*.

A. Dataset generation

The first step of the experiment was generating our own dataset samples which include endpoint positions of the upper limb and the aforementioned five joint angles. In order to do this, a human operator wore a pair of arm band in his arm and stood in front of a table, where a Kinect v2 sensor was put on it. The operator smoothly and slowly drew circle patterns with different radians and positions and sine curves with different configurations on the table, respectively. It was significant to synchronize all the devices by setting the ‘‘scissor’’ posture as the starting signal, when MYO and Kinect detected this posture, the system began to collect data. The collection frequency of Kinect and MYO is set to 30 Fps. Each time the operator draws a graph of different sizes, it lasts for 10 seconds, corresponding to 300 sets of data, a total of 10 circles of different sizes and positions, and 10 sine curves of different lengths and small heights, totalling 6000 sets of data. Here we keep the handmade dataset diversified enough to include differ patterns, considering the generalization of neural networks. So far, the dataset has been completely generated, which would be used to train the ELM as training samples in next step.

After the completion of dataset generation, a second/compared group of dataset was also required, where

only the endpoint positions were collected. The collection frequency of Kinect sensor was set to 15 Fps, and the operator slowly and smoothly drew a circle last for 10 seconds. The outputted data for this action would be used as the test sample dataset of ELM model.

B. Parameter optimization by BAS

The second step was optimizing the learning weights and bias of the ELM using BAS algorithm. The dataset generated in the previous step was used to train the ELM model, where the endpoint positions were as the input samples, joint angles were as output samples. In this paper, we define the hidden layer number is 60. The initial step size of BAS is set as 30, with 100 iterations, the attenuation coefficient η is set as 0.8. The second group of dataset collected from step 1 was used to compare with the estimated joints values. MSE was employed as the performance index to illustrate the efficiency of the ELM model with and without BAS optimization, respectively. The whole principle flowchart of angle joints data generation using ELM optimized by BAS can be shown in Fig.2. Fig.3 is

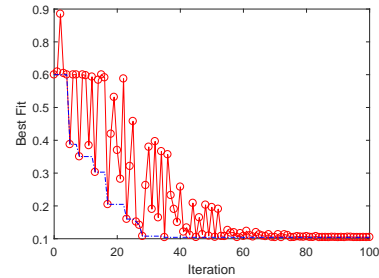


Fig. 4. The best fitness curve for BAS-ELM model

the fitting result of the BAS-ELM network prediction model training set. Comparing the unoptimized network fitting effect in Fig.3(a) and (c), it can be seen that the curve of the simulated value predicted by the BAS optimized model is closer to the true value of the predicted value shown in the comparison of Fig.3(a) and (b). In this paper, the MSE function is used to measure the performance of the ELM model. From the relative MSE results after optimization, the average relative error of BAS-ELM is 0.060256, while the average relative error of the ELM network training results before optimization is 0.13735, and the average relative error of prediction is reduced by 56.1%, indicating that after adding the BAS algorithm optimization, ELM model can deal with multi-dimensional variable problems more accurately. Fig.4 is the best fitness curve of the BAS-ELM model, that is, the curve of the best fitness function value for each update. It can be seen from the figure that the fitness function value reaches 0.1 after 60 generations and does not decrease anymore. That is, the optimal solution is found, and the convergence speed is relatively fast.

C. Estimated trajectory regeneration

The third step was to regenerate the trajectory outputted from ELM by a KUKA iiwa robot manipulator, whose

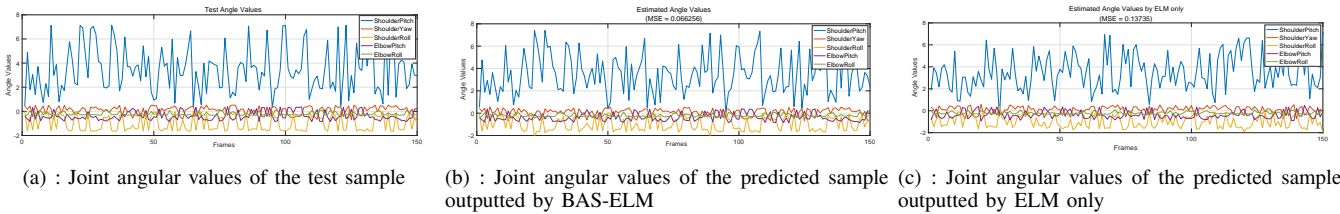
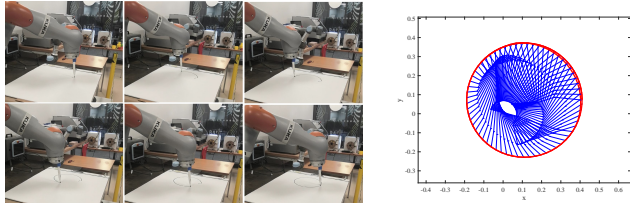


Fig. 3. Fitting result of the BAS-ELM network prediction model



(a) : Snapshots of the KUKA iiwa robot (b) : The tracked path in Cartesian for drawing the circle under the BAS-ELM space strategy

Fig. 5. Experimental validation on a real robotic platform

DH parameters can be found in our previous work [9]. All the five joint angles outputted by BAS-ELM with respect to *ShoulderPitch*, *ShoulderRoll*, *ShoulderYaw*, *ElbowPitch* and *ElbowRoll* were sent to the robot controller for the motion execution, where the angles for *WristPitch* and *WristYaw* were set as the initial constants of the robot. The parameters of the proposed BAS method follow the aforementioned configuration discipline. In term of the performance comparison results have been illustrated in the previous step (56.1% in error reducing) and the BAS has shown its sufficiently superiority than without one in predicting accuracy, we therefore directly use the strategy of BAS-ELM for experimental validation in the KUKA iiwa robot's controller. The resolved joint angles data from the BAS method are sent into the KUKA manipulator's operation system to perform path tracking control. The snapshots of the experiment session on the KUKA manipulator are shown in Fig.5(a), and its tracked path in Cartesian space is shown by Fig.5(b). From the result of trajectory regeneration, we can conclude that the proposed method can also work well on a real robotic platform.

VI. CONCLUSION

This paper developed an enhance learning interface for robotics using EML and BAS. The method of calculating the data of each joint angle through the MYO Armband has been studied and the collection of arm endpoint's coordinate data with Kinect v2 sensor has been completed, thereby a mini dataset including joint angles and endpoint positions was established. A BAS-ELM prediction model was proposed, which overcame the difficulty of selecting the initial learning weights and bias for ELM and greatly improves the prediction accuracy. Several experiments were conducted and validated the proposed method by increasing the average relative error of prediction with 56.1% than ELM only without parameter

optimization. The successful running of the trajectory generation on a real robot platform indicated that when the training sample size is big enough, the proposed BAS-ELM strategy can be used to replace the kinematics to map between the endpoint positions and relevant joint angles. Meanwhile, there are still some further tasks needing to be done in the future research works. How to improve the selection of BAS initial step size and solve the problem of inappropriate convergence is the direction of the future research.

REFERENCES

- [1] C. Li, C. Yang, and C. Giannetti, "Segmentation and generalisation for writing skills transfer from humans to robots," *Cognitive Computation and Systems*, vol. 1, no. 1, pp. 20–25, 2019.
- [2] H. T. Huynh, Y. Won, and J.-j. Kim, "An improvement of extreme learning machine for compact single-hidden-layer feedforward neural networks," *International journal of neural systems*, vol. 18, no. 05, pp. 433–441, 2008.
- [3] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006.
- [4] G.-B. Huang and L. Chen, "Enhanced random search based incremental extreme learning machine," *Neurocomputing*, vol. 71, no. 16, pp. 3460–3468, 2008.
- [5] H.-J. Rong, G.-B. Huang, N. Sundararajan, and P. Saratchandran, "Online sequential fuzzy extreme learning machine for function approximation and classification problems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 4, pp. 1067–1072, 2009.
- [6] M.-B. Li, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "Fully complex extreme learning machine," *Neurocomputing*, vol. 68, pp. 306–314, 2005.
- [7] B. P. Chacko, V. V. Krishnan, G. Raju, and P. B. Anto, "Handwritten character recognition using wavelet energy and extreme learning machine," *International Journal of Machine Learning and Cybernetics*, vol. 3, no. 2, pp. 149–161, 2012.
- [8] Q. Yang, H. Gao, W. Zhang, Z. Chi, and Z. Yi, "A new data-driven modeling method for fermentation processes," *Chemometrics and Intelligent Laboratory Systems*, vol. 152, pp. 88–96, 2016.
- [9] Y. Cheng, C. Li, S. Li, and Z. Li, "Motion planning of redundant manipulator with variable joint velocity limit based on beetle antennae search algorithm," *IEEE Access*, 2020.
- [10] K. Ishaque, Z. Salam, M. Amjad, and S. Mekhilef, "An improved particle swarm optimization (pso)-based mppt for pv with reduced steady-state oscillation," *IEEE transactions on Power Electronics*, vol. 27, no. 8, pp. 3627–3638, 2012.
- [11] X. Jiang and S. Li, "Beetle antennae search without parameter tuning (bas-wpt) for multi-objective optimization," *arXiv preprint arXiv:1711.02395*, 2017.
- [12] C. Li, C. Yang, Z. Ju, and A. S. Annamalai, "An enhanced teaching interface for a robot using dmp and gmr," *International journal of intelligent robotics and applications*, vol. 2, no. 1, pp. 110–121, 2018.
- [13] Y. Xu, C. Yang, P. Liang, L. Zhao, and Z. Li, "Development of a hybrid motion capture method using myo armband with application to teleoperation," in *2016 IEEE International Conference on Mechatronics and Automation*, pp. 1179–1184, IEEE, 2016.
- [14] G.-B. Huang, L. Chen, C. K. Siew, et al., "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 879–892, 2006.