Experience-based learning for multi-parameter regression control of snake-like robots

Dhanush Varma R[1] and Abhishek[2]

[1] Vellore Institute of Technology (VIT Bhopal), Bhopal, India
dhanushvarma1975@gmail.com
[2] Indira Gandhi National Open University, New Delhi, India
abhiverma414@gmail.com

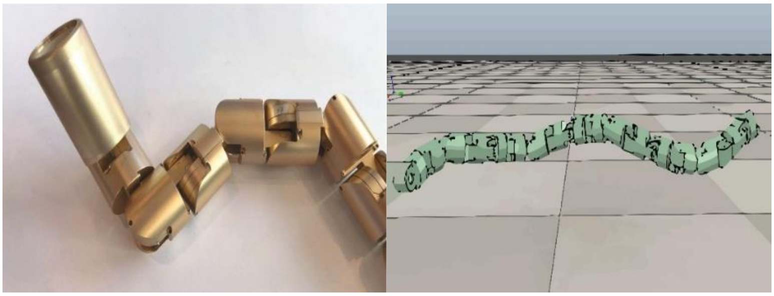
Abstract. Snake-like robots, with their complex and multi-jointed structure, hold great potential for navigating complex environments. However, real-time manipulation of their movements can be challenging. As such, achieving autonomous mobility for these robots is a major area of research. This paper introduces a new machine learning-based control framework that utilizes a clustering algorithm to classify training data into multiple clusters. The motion control of snake-like robots involves multiple regression problems due to the multi-parameter control strategy. To address this, we propose a novel strategy that uses data from previous training to convert multiple regressions into a single regression problem for parameter modification. Our experimental results demonstrate the adaptability of the robots in different pipe environments using our algorithm framework.

Keywords: Snake-inspired robot, autonomous mobility, motion control, parameter tuning, entropy variance

1 INTRODUCTION

Snake-inspired robots typically comprise a series of stiff links connected by joints, which are equipped with actuators to regulate their movement and create a structure similar to the skeletal system of a live snake. With multiple linkages and an impressive degree of freedom, snake-like robots have strong mobility in complex and unknown terrains [1]. Therefore, they can complete tasks on many occasions, such as post-disaster rescue missions [2], industrial pipe cleaning and repair [3] or scientific exploration [4]. In the past decade, novel features emerged, be it orthogonal or universal joint structure [5] [6] [7]. Consequently, snake-like robots acquired the ability of three-dimensional space movement, which strengthens their mobility in varied topography.

C. Kiran Mai et al. (eds.), *Proceedings of the Fourth International Conference on Advances in Computer Engineering and Communication Systems (ICACECS 2023)*, Atlantis Highlights in Computer Sciences 18,
https://doi.org/10.2991/978-94-6463-314-6_31



(a) Reality robot (b) Simulation robot

Fig. 1. The Snake-like robot

Although the snake-like robots have high mobility in many environments, the control of snake-like robots is complicated. A variety of motions can be constructed to get the same motion effect in the same environment. Moreover, snake-like robots and their movements cannot be easily predicted in an unknown environment.

Nowadays, there are two widely used control method. Firstly, we can pre-define the parameters to make robots move correctly in the known environment based on a widely used motion model [8] of snake-like robots. However, this method is not feasible for the unknown environment. Secondly, we can control the robot by tunning parameters artificially. While tunning parameters artificially is inefficient in unknown environments because it is difficult to tune the parameter accurately during robots' motion. In addition, controlling the robots is a professional work and tunning the parameters inaccurately will break the motion of snake- like robots. Therefore, autonomous mobility of robots is an expectation in robots' motion [9].

Autonomous mobility is an important and challenging task in robots' controlling [10]. Autonomous mobility requires high real-time performance to make robots respond to changes in the environment in time. What's more, High efficiency is critical in autonomous mobility. With high real- time performance and high efficiency, robots can make a rapid calculation when a mutation happens in the motion environment.

Our innovative experience-based proposal autonomously determines the optimal control parameters of a snake-like robot in the current environment by entropy variance [11] [12] [13]. By tunning the optimal control parameter with regression value through the weighted least squares algorithm [14] [15], our strategy can make the robot progressively adjust to its surroundings. Our control strategy can be divided into two parts:

1. data acquisition and preprocessing,
2. real-time data feedback and multi-parameter regression control.

To evaluate the effectiveness of the proposed framework, we also apply it to our existing snake-like robot platform, which is shown in Fig.1(a) and Fig.1(b). Under our frame- work, the robot climbs pipes having different diameters and the performance of climbing is reported. Mentioned below are the main points of our paper:

- A novel framework for robot adaptive control is pro- posed. This framework greatly reduces the amount of effort of the regression calculation by adopting clustering algorithm for data preprocessing and the classification of real-time data.
- A transformation of multiple regression problem is proposed to obtain the optimal control parameters in a real-time manner. The entropy variance is utilized to select the parameter which is the most sensitive in current state. Then, by regressing this parameter instead of all the parameters, we formulate a unit regression problem [16].
- Several case studies have been carried out, and the outcomes indicate that the approach we suggested is successful in achieving self-directed locomotion for robots inspired by snakes.

Early preparations are demonstrated in the Section II. In Section III, proposed strategy is explained in detailed. And then Simulation works are illustrated in Section IV. Section V shows the conclusion and our future prospects about the proposed strategy.

2 RELATIVE WORK

At present, the sinusoidal motion model is a prevalent control approach utilized for snake-inspired robots [8]. It was proposed by Professor Hirose. After that, Tesch et al. proposed a parametric equation based on the sinusoidal model for a snake-shaped robot with three-dimensional athleticism [17]. The implementation of a parametric equation facilitates the control methodology of robots inspired by snakes, resulting in a streamlined approach. This, in turn, permits the machine to identify its motion pattern using only a few control parameters, thus simplifying the entire process.

The sinusoidal motion model function is shown in the following formula:

$$T_i = \begin{cases} A.\sin(\omega.t + i.\varepsilon) & odd \\ A.\sin\left(\omega.t + i.\epsilon + \frac{\pi}{2}\right) & even \end{cases} \tag{1}$$

By adjusting the amplitude A, phase ε, and angular rate ω in Eq.1, we can alter the maximum joint rotation angle, the shaping period of the robot, and its serpentine motion rate. Such robots must possess self-adaptive capabilities to adjust to their environment.

To make the robots adapt to the unknown environment, the method which use the sensors to perceive the environment and embed the environment perception rules, has been widely used [18] [7] [19] [20]. A control methodology utilizing the central pattern generator (CPG) model was suggested by Tang et al [18]. Rollinson et al. introduced a control strategy for snake-inspired robots that adapts based on state estimation [7]. However, a complete correlation prediction model of control parameters is not given in their paper. As their methods is a gradient model based on state estimation, they are not suited to the mutation environment. Concerning robot control, some researchers have suggested utilizing machine learning techniques such as neural network models that incorporate information about the physical surroundings to establish effective control [21] [22] its effectiveness during the real-time motion of the robot is uncertain.

Our control strategy by experience-based learning is pro- posed. Combined with the clustering [25] [26] and the multi- parameter regression, we realize the real-time autonomous change of multiple control parameters in the robots' movement.

3 OVERVIEW

Our approach proposed in this paper is shown in Fig.2. We divide our approach into two main parts: off-line work and runtime execution.

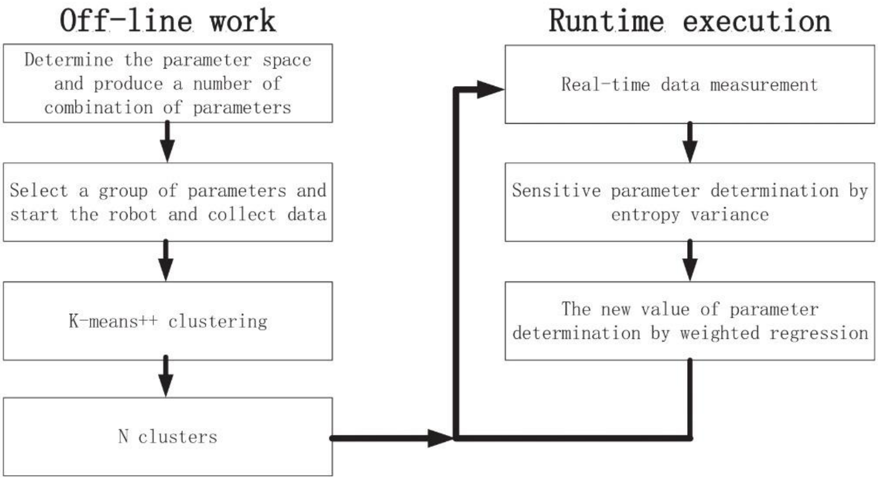


Fig. 2. The overall approach

3.1 OFF-LINE WORK

In the preprocessing work, we let the robot move along 25 cm and 35 cm pipes for a large number of times and we collect and store the data listed in the form like Table I.

Table 1. RECORDED VALUE DURING TRAINING

Symbol	Definition
$\theta_{m,i}$	The joint angle of the i_{th} joint which is measured
M	Mean of difference between measured joint
A	The value of amplifier
ω	The value of angular rate
ϵ	The value of phase

Our "Mean of difference" is $\sum_{i=1}^{n} \theta_{d,i} - \theta_{m,i}$ where n is the number of joints. $\theta_d = [\theta_{d,1}\ \theta_{d,2}\ \cdots\ \theta_{d,n}]$ is joint angles which we can calculate from Eq.1 and $\theta_m = [\theta_{m,1}\ \theta_{m,2}\ \cdots\ \theta_{m,n}]$ is the joint angles which are measured. The control vector is shown as [A ω ε] where A is the amplitude, ω is the angular rate, and the ε is the phase. All of them are applied in Eq.1.

After collecting movement data of a snake-like robot, we combine the training data obtained in different environment and then cluster the data in order to optimize real-time calculation.

In this research, collected data in preprocessing is an extensive collection of data. As k − means + + algorithm has high efficiency and scalability, we take k means + + for clustering. We classify training set into N clusters. The clustering process is shown below:

- Step 1: Decide the value of N by Eq.2.

$$N = arg\ \min_{N_k} \frac{\sum_{N_k}(S_i - E)^2}{N_k} \tag{2}$$

where $E = \frac{\sum_{N}(S_i)}{N}$ and $i \in [1, \text{N}]$.

- Step 2: perform k-means++ as Algorithm 1 shows.

Algorithm 1 k-means++

Input: Training set S, Blocks number N_k
Output: Initial cluster center X
function INITIALIZE($S^{(P)}$, N_k)
    X ← sample a point uniformly at random from S(P)
    while |X| < N_k **do**
        for $i = 1 \rightarrow |S^{(P)}|$ **do**
            I ← $arg\ \max_i(\sum_{j=1}^{|X|} ||X - P^{(i)}||^2)$
            X ← X ∪ {$P^{(I)}$}
            $S^{(P)} \leftarrow S^{(P)} - P^{(I)}$
        end for
    end while
    return X
end function

function UPDATE($S^{(P)}$, X)
    while stopping criterion has not been met **do**
        for $i = 1 \rightarrow |S^{(P)}|$ do
            $C^{(i)} \leftarrow \arg\min_k(||P^{(i)} - X^{(k)}||^2)$
        end for
        for $k = 1 \rightarrow |X|$ **do**

0: $X(k) \leftarrow \frac{\sum_i \{C^{(i)}=k\} P^{(i)}}{\sum_i \{C^{(i)}=k\}}$
0: **end for**
0: **end while**
0: **return** X
0: **end function**=0

After finishing the clustering, the result is stored in two parts:

- The value of each cluster center.
- The corresponding relationship between each group of data in training set and each cluster center.

With Kmeans++, the collected data can be divided into N_k clusters.

3.2 RUNTIME

In robot's running time, we get the real-time data periodically and then we do the following steps. Firstly, we categorize the real-time data based on the clustering result in off-line work. And then we select the most sensitive gait parameter according to the entropy variance. At last, we use the idea of weighted regression to modify the selected gait parameter and keep the other gait parameters unchanged.

A. Parameter selection by entropy variance

1. **Real-time data categorization:** Every time we get the real-time data, we relegate it to the certain cluster by Eq.3.

$$C = \arg\min_{N_k}(||X^{(k)} - P_t||_2) \tag{3}$$

$S^{(X)}$ is the cluster center set. $X^{(C)}$ is the closet vector to the real-time vector P_t. And $X^{(k)}$ is the i_{th} cluster center of the cluster center set $S^{(X)}$. With *Euclidian Distance Formula,* we make a prediction on the similarity between two vectors.

2. **The selection of the preponderant data:** After categorization of real-time data, we select those preponderant vectors whose Z-axis velocity are bigger than current (Eq.4).

$$S^{(V)} = \{P^{(i)} | V_z^{(i)} \geq V_z P^t, P^{(i)} \in S^{(C)}\} \tag{4}$$

$S^{(C)}$ is all the vectors which belong to the cluster with the cluster center $X^{(C)}$. $V_z^{(i)}$ is the vertical velocity com- ponent of the vector $P^{(i)}$ and $V_z^{(Pt)}$ is the vertical velocity component of the real-time data vector. $S^{(V)}$ is the set of all the preponderant vectors for regression.

3. **The selection of the sensitive parameter:** We adopt entropy variance as the reference to select the parameter which should be modified. The steps of selecting the sensitive parameter are as follows.

- *Step 1*: We take the preprocessing operation to discrete the preponderant data (Eq.5).

$$V_{new}^{(i)} = \begin{cases} \left\lfloor \frac{V_z^{(i)}}{L_D} \right\rfloor & V_z^{(i)} > 0 \\ \left\lceil \frac{V_z^{(i)} - L_D}{L_D} \right\rceil & V_z^{(i)} < 0 \end{cases} \tag{5}$$

In Eq.5, L_D is the adjustable step length for discretization. We eventually get the velocity discrete sequence:

$$V_{new}^{(Z)} = [\, V_{new}^{(1)}\ V_{new}^{(2)}\ V_{new}^{(3)}\ V_{new}^{(4)} \cdots\cdots] \tag{6}$$

- *Step 2*: There are a variety of possible values for each gait parameter. Thus, in order to record all the possible values, we make a set $S_{ij}(P)$ which is the data set of j_{th} possible value of the gait parameter *i*. And value of *i* is from 0 to 2 corresponding to amplitude, phase and angular rate respectively. We calculate the entropy about the vertical velocity of the j_{th} possible value of the gait parameter *i* by Eq.7 as well as the entropy variance of the gait parameter *i* by Eq.8. In Eq.7, $p(v_z)$ is the appearance rate of each member in the velocity discrete sequence as Eq.6. In Eq.8, E(H) is the mean of velocity of the j_{th} possible value of the gait parameter *i* and N_i is the number of possible values of the gait parameter *i*.

$$H(S)_{ij}^{(P)} = -\sum_{v_z \epsilon V_{new}^{(Z)}} p(v_z) \log_2 p(v_z) \tag{7}$$

$$Var_i^{(H)} = \frac{\sum_{N_i} (H(S_{ij}^{(P)} - E_{ij}^{(H)}))^2}{N_i} \tag{8}$$

- *Step 3*: We normalize the entropy variance (Eq.9).

$$R_i^{(var)} = \frac{Var_i^{(H)}}{\sum Var^{(H)}} \tag{9}$$

After the calculation of gait parameters' entropy variance. the sensitive parameter will be found. And then the selected parameter's value will be modified by regression.

B. Assignment to the value of the selected parameter

This study utilizes weighted regression to obtain the sensitive gait parameter value and applies the gradient descent method to resolve the weighted least squares problem when fitting the regression function.

- *Step 1*: List the fitting prediction function (Eq.10)

$$Fw(Pt) = W\,T\,Pt\,, W = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix} \tag{10}$$

In Eq.10, W is the coefficient sequence of the fitting equation and *m* is the number of coefficients where P_t is the real-time collected data vector. Then we can get the error function (Eq.11) which takes the square of error as the estimation with n being the number of data of $S^{(V)}$ and Q being the vector consisting of the sensitive parameter component $P^{(i)}(P^{(i)} \in P)$.

$$D(w) = \frac{1}{2n}(F_w(P)^T - Q)^T(F_w(P)^T - Q) \tag{11}$$

$$P = \left[P^{(1)}_{(n)}\ P^{(2)} \cdots P\right],\ P^{(i)} \epsilon\ S^{(V)} \tag{12}$$

$$Q = [Q^{(1)}\ Q^{(2)}\ \cdots Q^{(n)}]^T \tag{13}$$

To get the best-fit coefficient sequence W by the minimum D(w), according to gradient descent method, we turn the Eq.11 into Eq.14.

$$\nabla_w D\ = \frac{1}{n} P(F_w(P)^T\ -\ Q) \tag{14}$$

- *Step 2*: Perform the weighted operation on preponderant data vector to ensure the estimate result of fitting is good (Eq.16).

$$\nabla_w D\ = \frac{1}{n} PM(F_w(P)^T\ -\ Q) \tag{15}$$

$$M = \begin{bmatrix} \frac{V_z^{(1)}}{L_s} & 0 & \dots & 0 \\ 0 & \frac{V_z^{(1)}}{L_s} & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \frac{V_z^{(n)}}{L_s} \end{bmatrix} \tag{16}$$

In Eq.16, L_s is the learning step and M is the learning rate matrix.

- *Step 3*: Run coefficient vector by Eq.17.

$$W\ =\ W\ -\ \nabla_w D \tag{17}$$

In this way, the coefficient sequence W is updated. And then we assign the predicting result to the sensitive parameter (Eq.18).

$$Q\ =\ F_w(P_t)\ =\ W^T\ P_t \tag{18}$$

After this the new predicting result is produced.

- *Step 4*: Repeat the aforementioned steps until the most suitable coefficient is obtained, and W is stable finally.151

When iterative process is stopped, the best-fit coefficient W_{best} will be obtained. By applying W_{best} to Eq.18, we can get the regression value of the sensitive parameter. This value will be used in the robot control. It is worth noting that we only modify the sensitive parameter and others remain the same value.

4 SIMULATION

For simulations, we adopt the rolling gait as the funda- mental motion pattern. To validate the adaptive control of snake-inspired robots during locomotion, we simulate their movements on pipes of varying diameters using the V-REP robot simulation platform.

The simulation process can be divided into two categories: training and motion simulation. And we divide simulations into two parts: the adaptable motion along variable diameter pipes and the adaptable motion along straight pipes with different diameters.

A. Training Process

In the data acquisition process, we let the robot climb along the 25cm and 35cm pipes under different and collect 25 thousand volumes of training data in total. The interval of amplitude A, phase ε and angular rate ω are [40, 80], [0, 5] and [1.5, 3] respectively. What's more, the step of A is 5, the step of ε is 1 and the step of ω is 0.5. We make the robot climb with different combination of parameters and then collect the data. In the preprocessing process, we cluster the training data. We set the number of clusters as 25 by Eq.2 and Fig.3. The number of data for most of classes is 1000 ± 500 (Fig.4).

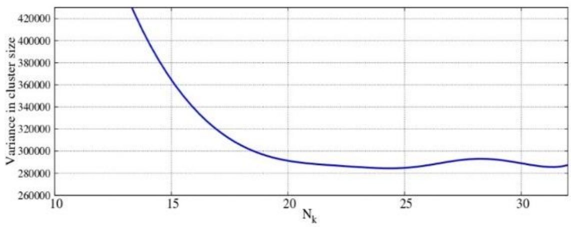


Fig. 3. The variance in cluster size of N_k

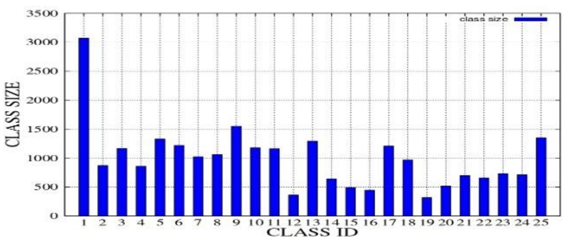


Fig. 4. The result of clustering

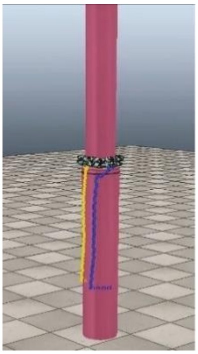
B. The autonomous motions along variable diameter pipes

Climbing simulation in this part is climbing the pipe with 35cm lower diameter and 25cm upper diameter and climbing the pipe with 25cm lower diameter and 35cm upper diameter. Next, the two groups of simulation will be analyzed in detail.

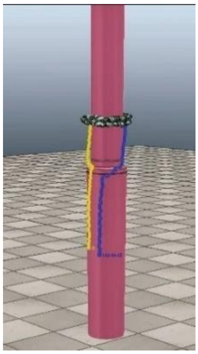
1. Simulation about climbing the pipe with 35cm lower diameter and 25cm upper diameter: The results are shown in Fig.5. During the initial 15 seconds, the robot alters its configuration to accommodate the unfamiliar pipe, resulting in a near-zero velocity. Subsequently, from 15 to 40 seconds, the robot climbs up the 35cm diameter pipe. In this phase, amplitude A, phase Ɛ and angular rate Ω are all increasing and all of the them will be stable finally. Around the 50-second mark, the robot reaches the interface where the diameter of the pipe alters. As the pipe changes obvious, the robot cannot grasp the 25cm pipe immediately, which causes the velocity of the robot fluctuate around zero. The robot keeps learning and autonomously adjust its parameters to continue its climbing motion. From the Fig.5(d) and Fig.5(e), it can be found that the amplitude and phase are increasing in this phase, which make the robot continue moving up.

Eventually all control parameters as well as velocity are stable. It shows that under this control strategy, the robot can adjust its parameters autonomously to adapt to the environment.

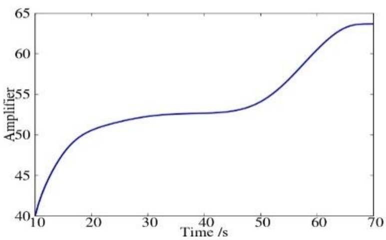
(a)t=39.8s



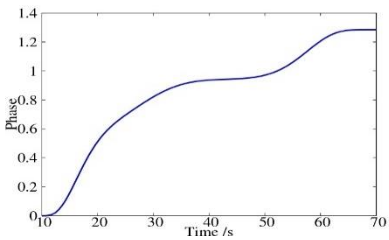
(b) t=60.0s



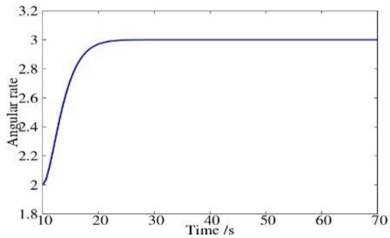
(c) t=64.5s



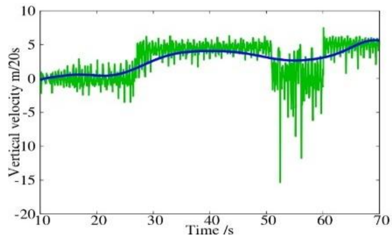
(d) Amplifier versus Time



(e) Phase versus Time



(f) Angular rate versus Time



(g) velocity versus Time

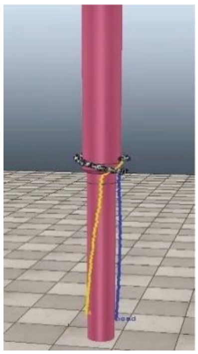
Fig. 5. The movement and the curves of parameters in the motion when the robot climb the pipe with 35cm lower diameter and 25cm upper diameter

2. Simulation about the pipe with 35cm lower diameter and 25cm upper diameter:

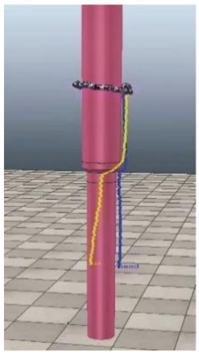
It is similar with the previous simulation, but the movement is different. The results are shown in Fig.6. Around 52 seconds, the robot comes across the interface where the pipe diameter shifts. At this time the robot autonomously reduce amplitude slightly (Fig.6(d)) and then significantly increase phase (Fig.6(e)). In this way, the control strategy makes the robot move itself from the 25cm part up to the 35cm part.

Both simulations show that our method is effective to adapt the robot to climb along the variable diameter pipe.

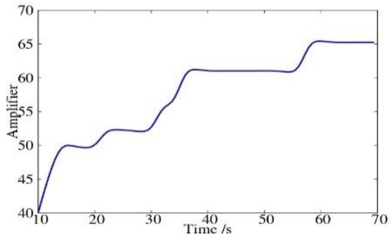
(a) t=40.5s



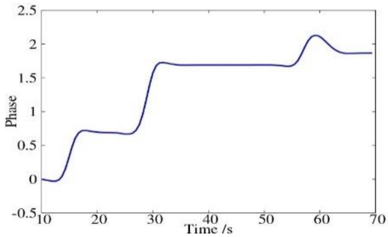
(b) t=58.9s



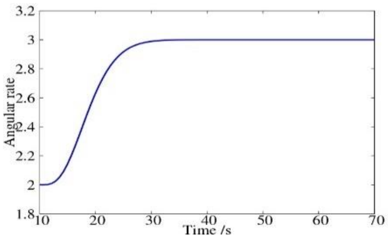
(c) t=66.0s



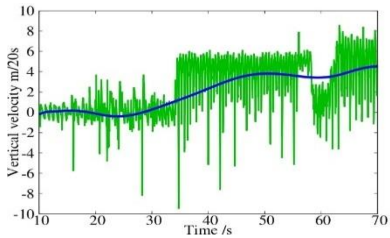
(d) Amplifier versus Time



(e) Phase versus Time



(f) Angular rate versus Time



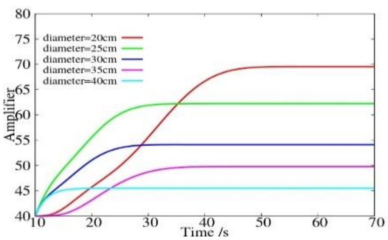
(g) velocity versus Time

Fig. 6. The movement and the curves of parameters in the motion when the robot climb the pipe with 25cm lower diameter and 35cm upper diameter

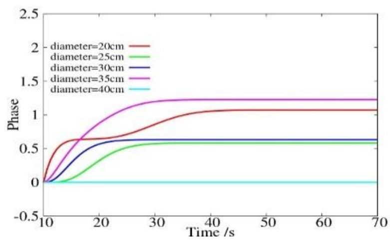
C. Contrast experiments on other straight pipes

As the unknown environment in the real world is complex and volatile, there may be differences between training environment and actual environment. Our training environment is vertical pipes with 25cm or 35cm diameter. To ascertain that this control approach can be employed in a variety of scenarios, we carry out multiple independent simulations on straight pipes with diameters of 20cm, 25cm, 30cm, 35cm, or 40cm. The result of these simulations is shown in Fig.7. Next is the detailed analysis.

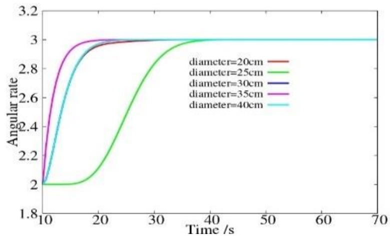
By observing the parameter curve we can find that the smaller diameter of the pipe is, the much more time the parameters need to spend in adjusting. All parameters finally reach a stable value and satisfy the requirements of movement. The robot cannot form a complete loop to coil around the 40cm diameter pipe due to its insufficient length. In this case, the influence made by phase Ɛ is very weak (Fig.7(b)). For the 20cm diameter pipe, it is difficult for the robot to climb along the pipe if we only increase the amplitude A, because when the amplitude is too large, it will cause the snake-like robot curling up to a high degree, which is not conducive to climb. Therefore, to let the robot climb small diameter pipe, we need to constantly adjust the phase Ɛ to meet the amplitude A's corresponding requirements. By observing the variation curve of the control parameters of the small diameter pipe, we can found that the phase Ɛ and the amplitude A of the robot are coordinated in the process of self-regulation (Fig.7(a)) (Fig.7(b)). The simulations show that the robot can use our control strategy to catch the unknown pipe and adjust itself to a suitable climbing state.



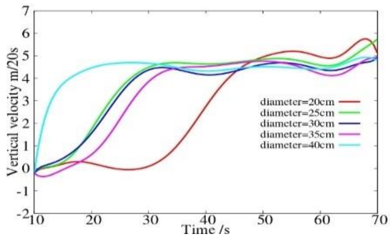
(a) Amplifier versus Time



(b) Phase versus Time



(c) Angular rate versus Time



(d) velocity versus Time

Fig. 7. The movement and the curves of parameters in the motion when the robot climb the pipe with different diameters

It is noteworthy that, in all testing environments, the robot's movement velocity ultimately oscillates within a consistent range (Fig.7(d)). The pipe diameter solely impacts the rate at which the velocity converges, and the control parameters of the robot ultimately attain stability (Fig.7). Hence, we see the external environment's effect on this control strategy is very weak. What's more, the robot will find the optimal parameter during its motion.

In conclusion, the simulations demonstrate that the adaptive control in robot's motion can be realized through our proposed framework.

5 CONCLUSION AND FUTURE WORK

Our robot-based adaptive control is based on the learning experience of robots. We take Z-axis velocity as feed-back signal and adopt regression to correct the robots' action. We simplify the runtime learning through clustering and transform the multiple regression into unit regression. Experiments show that the scheme is effective.

This approach is noteworthy for its potential applicability beyond pipe climbing, extending to other areas of robotics. We believe that the algorithm can adapt to the other corresponding scene, such as the unmanned vehicle's variable motion, the rugged ground motion of the serpentine robot and the simulated PID control as long as enough training data and clear moving purpose are given.

Moving forward, we intend to enhance the algorithm via the following means. The hierarchical clustering [27] [28] will be applied to the model to achieve uniform clustering to ensure that the data volume of each regression is consistent. Existing regression models will also be improved. What's more, we will carry out testing in much more complex climbing scene such as simulating trees in nature and bifurcate pipelines. We will develop much more complex rules for learning and running in much more complex environments.

REFERENCES

1. G. S. Chirikjian and J. W. Burdick, "The kinematics of hyper- redundant robot locomotion," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 6, pp. 781–793, Dec 1995.
2. A. Ferworn, C. Wright, J. Tran, C. Li, and H. Choset, "Dog and snakemarsupial cooperation for urban search and rescue deployment," in *2012 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Nov 2012, pp. 1–5.
3. S. Hirose and H. Yamada, "Snake-like robots [tutorial]," *IEEE Robotics Automation Magazine*, vol. 16, no. 1, pp. 88–98, March 2009.
4. A. Kuwada, Y. Adomi, K. Suzumori, and Takefumi, "Snake-likerobot negotiating three-dimensional pipelines," in *IEEE International Conference on Robotics and Biomimetics*, 2007, pp. 989–994.
5. T. Kamegawa, F. Matsuno, and R. Chatterjee, "Proposition of twistingmode of locomotion and ga based motion planning for transition of locomotion modes of 3-dimensional snake-like robot," in *Proceedings2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292), vol. 2, 2002, pp.1507-1512 vol. 2.*
6. H. Date and Y. Takita, "Control of 3d snake-like locomotive mech- anism based on continuum modeling," in *ASME 2005 International Design Engineering Technical Conferences and Computers and Infor-mation in Engineering Conference*, 2005, pp. 1351–1359.
7. D. Rollinson and H. Choset, "Gait-based compliant control for snake robots," in *2013 IEEE International Conference on Robotics andAutomation*, May 2013, pp. 5138–5143.
8. H. Ohno and S. Hirose, "Design of slim slime robot and its gait of locomotion," in *Proceedings 2001 IEEE/RSJ International Conferenceon Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, vol. 2, 2001, pp. 707–715 vol.2.
9. M. K. Naveen, M. Lokesh, K. Varma, and P. Teja, "A review on autonomous vehicles and its components," *Journal of PharmaceuticalNegative Results*, pp. 6916–6922, 2023.
10. W. Liu, M. Hua, Z. Deng, Y. Huang, C. Hu, S. Song, L. Gao, C. Liu, L. Xiong, and X. Xia, "A systematic survey of control techniques andapplications: From autonomous vehicles to connected and automated vehicles," *arXiv preprint arXiv:2303.05665*, 2023.
11. F. Zhu, Y. Feng, Y. Luo, and Q. Zhang, "A novel method of identifying moving status of bird based on waveform entropy variance," in *2008 International Conference on Communications, Circuits and Systems*, May 2008, pp. 922–926.
12. L. Yang, "Study on cumulative residual entropy and variance as risk measure," in *2012 Fifth International Conference on Business Intelligence and Financial Engineering*, Aug 2012, pp. 210–213.
13. N. Bagherzadeh and S. J. S. Mahdavi, "A new modified iwo algorithmby using entropy and variance," in *2015 International Congress on Technology, Communication and Knowledge (ICTCK)*, Nov 2015, pp.441–446.
14. M. Garcia-Sanz and J. Florez, "Adaptive optimum start-up and shut- down time controllers for heating systems based on a robust gradient method," *IEE Proceedings - Control Theory and Applications*, vol. 141, no. 5, pp. 323–328, Sep 1994.
15. L. D. Abreu and J. L. Romero, "Mse estimates for multitaper spectral estimation and off-grid compressive sensing," *IEEE Transactions on Information Theory*, vol. PP, no. 99, pp. 1–1, 2017.

16. A. Mizumoto, “Calculating the relative importance of multiple re- gression predictor variables using dominance analysis and random forests,” *Language Learning*, vol. 73, no. 1, pp. 161–196, 2023.
17. M. Tesch, K. Lipkin, I. Brown, R. L. Hatton, A. Peck, J. Rembisz,and H. Choset, “Parameterized and scripted gaits for modular snake robots,” vol. 23, pp. 1131–1158, 06 2009.
18. C. Tang, S. Ma, B. Li, and Y. Wang, “A self-tuning multi-phase cpg enabling the snake robot to adapt to environments,” in *2011 IEEE/RSJInternational Conference on Intelligent Robots and Systems*, Sept2011, pp. 1869–1874.
19. C. G. and H. Ranganathan, “Balancing and control of dual wheel swarm robots by using sensors and port forwarding router,” in *2017 Third International Conference on Advances in Electrical, Electronics,Information, Communication and Bio-Informatics (AEEICB)*, Feb 2017, pp. 532–536.
20. G. Gerboni, A. Diodato, G. Ciuti, M. Cianchetti, and A. Menciassi, “Feedback control of soft robot actuators via commercial flex bend sensors,” *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 4, pp. 1881–1888, Aug 2017.
21. G. Martius, R. Der, and N. Ay, “Information driven self-organization of complex robotic behaviors,” vol. 8, p. e63400, 05 2013.
22. R. Der and G. Martius, “Novel plasticity rule can explain the devel- opment of sensorimotor intelligence,” vol. 112, 05 2015.
23. J. P. Cai, L. Xing, M. Zhang, and L. Shen, “Adaptive neural network control for missile systems with unknown hysteresis input,” *IEEE Access*, vol. 5, pp. 15 839–15 847, 2017.
24. A. Vitiello, G. Acampora, M. Staffa, B. Siciliano, and S. Rossi, “A neuro-fuzzy-bayesian approach for the adaptive control of robot proxemics behavior,” in *2017 IEEE International Conference on FuzzySystems (FUZZ-IEEE)*, July 2017, pp. 1–6.
25. A. Widodo and I. Budi, “Clustering patent document in the field of ict(information amp; communication technology),” in *2011 InternationalConference on Semantic Technology and Information Retrieval*, June 2011, pp. 203–208.
26. M. Dundar, Q. Kou, B. Zhang, Y. He, and B. Rajwa, “Simplicity of kmeans versus deepness of deep learning: A case of unsupervised feature learning with limited data,” in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, Dec 2015, pp. 883–888.
27. P. Shi, J. Zhong, R. Huang, and J. Lin, “Automated quantitative imageanalysis of hematoxylin-eosin staining slides in lymphoma based on hierarchical kmeans clustering,” in *2016 8th International Conferenceon Information Technology in Medicine and Education (ITME)*, Dec 2016, pp. 99–104.
28. N. Salout, F. A. Awin, A. F. Alqawasmeh, and E. Abdel-Raheem, “Hierarchical cluster-based cooperative spectrum sensing in cognitive radio employing soft-hard combination,” in *2017 IEEE 30th CanadianConference on Electrical and Computer Engineering (CCECE)*, April2017, pp. 1–4.