SVM-based Gait Control Method For Hydraulic Quadruped Robots

Guo Zhang*, Wenliang Deng, and Ling Wei

Intelligent Manufacturing College, Chongqing Creation Vocational College, Chongqing 402160, China

Corresponding author. E-mail: zg52017@163.com

Received: Oct. 25, 2022; Accepted: Apr. 18, 2023

With the development of technology, there are more and more researches on quadruped robots in the field of artificial intelligence, but there is a lack of research on gait control methods of quadruped robots. To solve this problem, a GA-SVM algorithm model is proposed by combining Support vector machine (SVM) and Genetic Algorithm (GA). This model optimizes the important parameters of SVM through GA, and improves the performance of SVM. Then compare it with neural network (NN) model and traditional SVM model to verify its performance. The experimental results show that the value of GA-SVM model in the training set is 0.9215, and its performance is better than that of traditional NN model and traditional SVM model. In the simulation test of gait control system, GA-SVM model is 9.391, which is better than traditional NN model and traditional SVM model. The results show that the performance of GA-SVM model obtained by the combination of GA and SVM has been greatly improved compared with the traditional SVM, which can provide a new idea and method for the gait control of quadruped robot.

Keywords: GA; Gait control; Quadruped robot; SVM; Artificial intelligence

© The Author('s). This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are cited.

http://dx.doi.org/10.6180/jase.202402_27(2).0002

1. Introduction

Nowadays, with the development of society, the technology in the field of artificial intelligence is becoming more mature, and the application of intelligent robots is becoming more and more extensive [1]. Among them, the bionic research of quadrupedal robots has aroused the concern of many experts because of its great research and development value [2]. Although a wide variety of quadruped robots with different styles have been developed one after another, the overall performance of these quadruped robots failed to meet the expected goals [3]. So it is very important to propose a gait model that can accurately control quadruped robot. And SVM, as a common classification algorithm, is often used in various text recognition and classification because of its algorithmic simplicity as well as possessing good robustness [4]. GA is a kind of optimization algorithm with good optimization performance [5]. The GA-SVM model was obtained by optimizing the SVM with GA, and the GA-SVM model was embedded into the gait control system of quadruped robot. It is expected to further improve the accuracy of the gait control of quadruped robots and provide a new idea for the development of the field of gait control of quadruped robots.

2. Related works

Since the rapid development of SVMs in the 1990s, a large number of improved algorithms have been derived and thus are used in multiple areas.

Husada and Paramita proposed an improved analysis method of multi-class SVM to solve the problem that Twitter could not classify public satisfaction. The method was tested and the results showed a realistic accuracy of 84.37% [6]. Chen's team proposed an SVM-based classification model for retrospective data For the sake of reduce drilling costs and improve drilling operations by finding and generalizing popular mathematical models to derive a

multi-objective optimization model. By testing the model, the classification model improves the accuracy of field development data and provides a new idea for improving drilling operations [7]. Zhang et al. proposed a clustering algorithm combining SVM and LCA to solve the problem of poor results of existing clustering detection methods. Through testing the method, it is found that the clustering algorithm has higher accuracy and faster detection time than the traditional clustering algorithm [8]. Song's team proposed the least squares SVM prediction model LSSVM-PGV to address the problem of insufficient accuracy of field instrumented seismic warning models [9]. The results showed that the model on the training and test sets The standard deviation of the prediction error tends to be the same on the training and test sets, indicating that the model has generalization performance. Bose et al. designed an automatic plant disease detection system based on SVM to solve the problem that it is hard to accurately identify plants and their diseases. The test of the detection system shows that the accuracy of the two feature selection of SVM is 99%. The results indicate that the detection system can be used to accurately identify plants and their diseases [10].

And there are various approaches applied to robot control. The team of Farkh proposed an advanced controller based on deep neural networks for hospital service robots in response to the lack of medical facilities to cope with epidemics. The results showed that the controller significantly slowed down the pressure of hospital visits and provided access to more patients [11]. Togias et al. proposed the concept of teleoperation-based remote reprogramming of robots For the sake of achieve flexibility that can handle different tasks in production. The results showed that the concept, it can reduce the time and energy needed to readjust the operation purpose of the robot, so as to improve the operation flexibility of the robot [12]. Galushko's team proposed the development of a testbed to test underwater robotic systems in response to the dramatic increase in energy efficiency as well as noiselessness problems of underwater robots. The results showed that the testbed could properly test the main actuation systems of various underwater robots and reduce their noiseless problems [13]. For the sake of solve the problem of high complexity of highly adaptable articulated arm robots, Kunert et al. designed a system that combined control method, design method and reinforcement learning, and tested the designed system and found that the system could significantly reduce the time complexity of machine learning [14]. Mikada's team proposed a human cooperative controller with YOLOv3 and standard convolutional neural networks for the complex problem of surgical robot surgical task automation. The experimental results of the human cooperative control showed that the method is effective and can operate on patients more precisely and improve the success rate of surgery [15].

The above research shows that SVM model has been applied in many fields and applied to robot control in a variety of ways. However, it is rare to combine SVMs with robot gait control. So in the study, a GA-SVM model is obtained by optimizing the SVM model using GA, and the GA-SVM model is applied to the gait control of a hydraulic quadruped robot to obtain a system for controlling the gait of a hydraulic quadruped robot. It is used to improve the poor gait walking ability of today's hydraulic quadruped robots.

3. Research on svm-based control method for quadruped robots

3.1. SVM algorithm optimization

SVM algorithms are often used in the field of classification problems because of their excellent performance in solving small sample and nonlinear problems [16]. The biggest advantage of SVM algorithms is that they can map linearly indistinguishable problems to a high-dimensional space by introducing kernel functions, thus turning linear indistinguishability into linear differentiability [17]. The example of linear indivisibility of SVM algorithm is shown in Fig. 1.

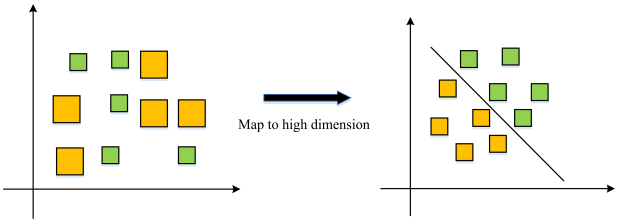

Fig. 1. Linear indivisible example of SVM algorithm.

SVM can be applied to regression problems in addition to solving classification problems, but the application of SVM in regression problems retains its main features in classification problems [18]. That is, the regression function is obtained from several special points in a high-dimensional space and the process is the problem that can be considered as a planning again, with the ultimate goal of obtaining the desired Lagrange multipliers by adjusting the insensitive loss coefficients and relaxation variables. Fig. 2 shows the structure of Support Vector Regression (SVR) machine.

The basic principle of linear regression is as follows. For general linearity the functional expression of regression is needed as shown in Eq. (1).

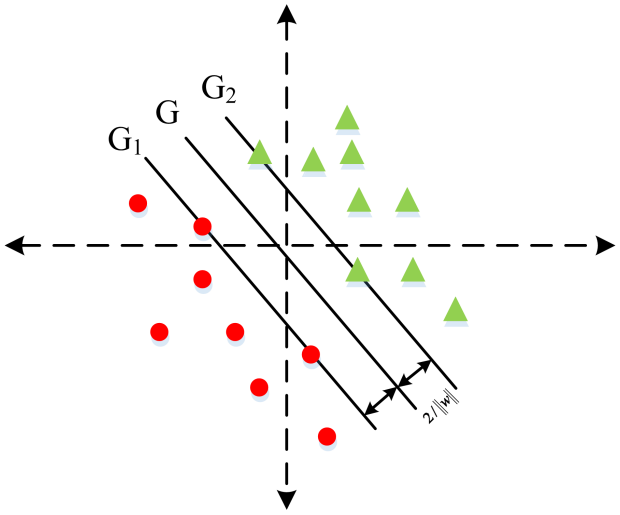

Fig. 2. Support vector regression machine.

$$f(x) = w^T x + c \tag{1}$$

The objective of this function is the minimum confidence range in Eq. (2).

$$\begin{cases} \min \frac{1}{2}\|w\|^2 \\ \text{s.t.} \quad y_k - wx_k - c \le \varepsilon \\ wx_k - y_k + c \le \varepsilon \quad k = 1, 2, 3, \ldots, N \end{cases} \tag{2}$$

In Eq. (2), $\min \frac{1}{2}\|w\|^2$ represents the minimization confidence range, which can be used as the precondition of the optimization problem. If the precondition cannot be achieved, the slack variables ξ_k and ξ_k^* need to be introduced to relax the precondition, and then the objective of the linear regression function is the minimum confidence range in Eq. (3).

$$\begin{cases} \min \quad \frac{1}{2}\|w\| + D\sum_{i=1}^{N}(\xi_k + \xi_k^*) \\ \text{s.t.} \quad \begin{array}{c} y_k - wx_k - c \le \varepsilon + \xi_k^* \\ wx_k + c - y_k \le \varepsilon + \xi_k^* \quad k = 1, 2, \ldots, N \\ \xi_k \ge 0 \\ \xi_k^* \ge 0 \end{array} \end{cases} \tag{3}$$

D in Eq. (3) is the punish coefficient. Then, the constraint problems are optimized by Lagrange algorithm. The Lagrange function expression is shown in Eq. (4).

$$\begin{aligned} L(w, c, \alpha_k, \alpha_k^*, \beta_k, \beta_k^*, \xi_k, \xi_k^*) &= \frac{1}{2}\|w\| - D\sum_{k=1}^{N}(\xi_k + \xi_k^*) \\ &- \sum_{k=1}^{N}\alpha_k(\varepsilon + \xi_k + wx_k + c - y_k) - \sum_{i=k}^{N}\alpha_k^*(\varepsilon + \xi_k^* + y_k - wx_k - c) \\ &- \sum_{i=k}^{N}(\beta_k\xi_k + \beta_k^*\xi_k^*) \end{aligned} \tag{4}$$

In Eq. (4), $\alpha_k, \alpha_k^*, \beta_k, \beta_k^*, \xi_k, \xi_k^* \ge 0, k = 1, 2, ..., N$ represents the optimization multiplier, and the derivative of w, c, ξ_k, ξ_k^* in $L(w, c, \alpha_k, \alpha_k^*, \beta_k, \beta_k^*, \xi_k, \xi_k^*)$ is derived by Lagrangian theory so that the derivative is zero to obtain Eq. (5).

$$\begin{cases} \frac{\partial L(w,c,\alpha_k,\alpha_k^*,\beta_k,\beta_k^*,\xi_k,\xi_k^*)}{\partial w} = w - \sum_{k=1}^{N}(\alpha_k - \alpha_k^*)x_k = 0 \\ \frac{\partial L(w,c,\alpha_k,\alpha_k^*,\beta_k,\beta_k^*,\xi_k,\xi_k^*)}{\partial c} = \sum_{k=1}^{N}(\alpha_k - \alpha_k^*) = 0 \\ \frac{\partial L(w,c,\alpha_k,\alpha_k^*,\beta_k,\beta_k^*,\xi_k,\xi_k^*)}{\partial \xi_k} = D - \alpha_k - \beta_k = 0 \\ \frac{\partial L(w,c,\alpha_k,\alpha_k^*,\beta_k,\beta_k^*,\xi_k,\xi_k^*)}{\partial \xi_k^*} = D - \alpha_k^* - \beta_k^* = 0 \end{cases} \tag{5}$$

Variation of Eq. (5) yields Eq. (6).

$$\begin{cases} w = \sum_{k=1}^{N}(\alpha_k - \alpha_k^*)x_k \\ \sum_{k=1}^{N}(\alpha_k - \alpha_k^*) = 0 \\ D = \alpha_k + \beta_k \\ D = \alpha_k^* + \beta_k^* \end{cases} \tag{6}$$

In addition to satisfying the above conditions, w, c, ξ_k, ξ_k^* should also satisfy the complementary condition in Eq. (7).

$$\begin{cases} \alpha_k\left[\varepsilon + \xi_k - y_k + \langle w - x_k\rangle + c\right] = 0 \\ \alpha_k^*\left[\varepsilon + \xi_k^* - y_k + \langle w - x_k\rangle - c\right] = 0 \\ \beta_k\xi_k = (D - \alpha_k)\xi_k = 0 \\ \beta_k\xi_k^* = (D - \alpha_k^*)\beta_k^* = 0 \quad k = 1, 2, \ldots, N \end{cases} \tag{7}$$

Combining Eqs. (6) and (7) yields Eq. (8).

$$\begin{cases} \max L(\alpha_k, \alpha_k^*) = \sum_{k=1}^{N} y_k(\alpha_k - \alpha_k^*) \\ -\frac{1}{2}\sum_{kj-1}^{N}(\alpha_k - \alpha_k^*)(\alpha_j - \alpha_j^*)(x_{k_k} - x_j) \\ \text{s.t} \quad \sum_{k=1}^{N}(\alpha_k - \alpha_k^*) = 0 \\ \alpha_k, \alpha_k^* \in [0, D] \end{cases} \tag{8}$$

Solving for this function, the $\alpha_k, \alpha_k^* \ne 0$ vector is obtained as the support vector, which leads to the expression of the linearly differentiable support vector regressor as shown in Eq. (9).

$$f(x) = \sum_{k=1}^{N}(\alpha_k - \alpha_k^*)\langle x_k \cdot x\rangle + c \tag{9}$$

The SVR machine is an insensitive loss function ε introduced into the SVM, and its architecture is shown in Fig. 3.

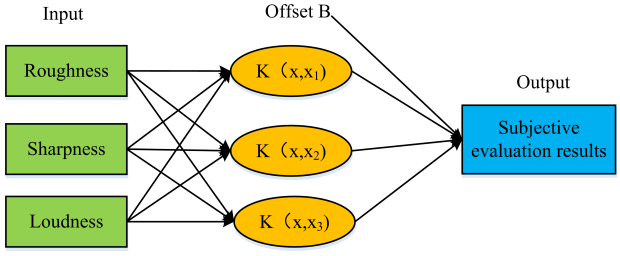

Fig. 3. Schematic diagram of SVR architecture.

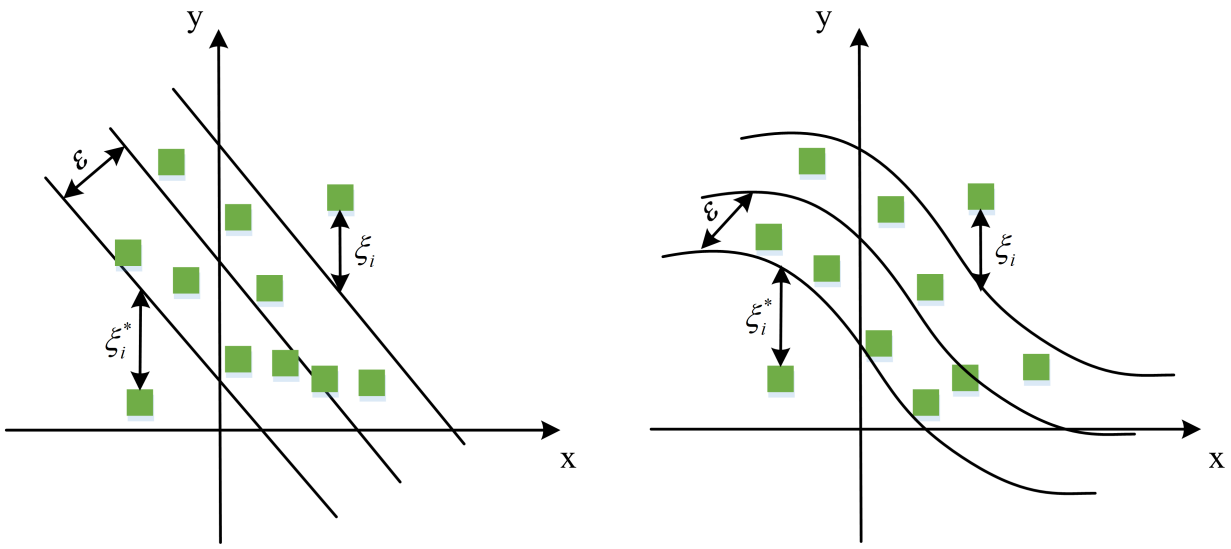

Fig. 4. Insensitivity band graph in SVR regression problem.

In Fig. 3, $K(x, x_{,k})(k = 1, 2, 3, ..., N)$ is the kernel function and c is the bias function, which are the most important components of the SVR learning machine. Assuming that the feature space has the training sample set T, the expression of T is shown in Eq. (10).

$$T = (x_1, y_1), (x_2, y_2), ..., (x_k, y_k) \tag{10}$$

In Eq. (10), x_k is the sample value of the k function, which is also the input quantity, and y_k indicates the output quantity. The fitting function is shown in Eq. (11).

$$f(\vec{x}) = \vec{w} \cdot \vec{x} + c \tag{11}$$

Define the expression of the insensitive loss function as shown in Eq. (12).

$$|y - f(x)|_z = \begin{cases} 0 & |y - f(x)| \le \varepsilon \\ |y - f(x)| - \varepsilon & |y - f(x)| \ge \varepsilon \end{cases} \tag{12}$$

In Eq. (12), ε is the coefficient of the insensitive loss function. This expression expresses that given a loss function, for any sample x corresponding to the true value y and the predicted value $f(x)$, if the difference between them is smaller than ε, then the predicted value of x has no loss and the loss function results in 0; if the difference between them is larger than ε, then the value of the loss function is the difference between the predicted and true values minus ε. So when ε is determined, the expression of the constraint of the regression function is shown in Eq. (13).

$$\begin{cases} \vec{w} \cdot \vec{x} + c - y_k \le \varepsilon \\ \vec{w} \cdot \vec{x} + c - y_k \le \varepsilon \end{cases} \tag{13}$$

Because of the existence of noisy samples in the sample set, regression errors will be generated, and it is necessary to introduce the slack variables ξ_k and ξ_k^*, and the SVR insensitive band diagram after the introduction of slack variables is shown in Fig. 4.

As shown in Fig. 4, the scope of the SVR algorithm can be extended by introducing the slack variables, at which point Eq. (13) can be derived as Eq. (14).

$$\begin{cases} \vec{w} \cdot \vec{x} + c - y_k \le \varepsilon + \xi_k^* \\ \vec{w} \cdot \vec{x} + c - y_k \le \varepsilon + \xi_k \end{cases} \tag{14}$$

In Eq. (14), ξ_k and ξ_k^* are relaxation variables. SVR solves nonlinear problems with the help of kernel functions, through which the nonlinear problems are mapped to higher dimensional space and converted to linear problems for solution. The expression of the kernel function is shown in Eq. (15).

$$K\left(\vec{x}, \overrightarrow{x_k}\right) = \overrightarrow{\phi(\vec{x})} \phi\phi \overrightarrow{\left(\overrightarrow{x_k}\right)} \tag{15}$$

In Eq. (15), $\vec{x}$ and $\overrightarrow{x_k}$ are vectors in low-dimensional space, and $\overrightarrow{\phi(\vec{x})}$ and $\overrightarrow{\phi(\vec{x_k})}$ are the corresponding vectors in high-dimensional space. In SVR learning, the kernel function is usually defined directly, and the mapping ϕ is obtained from the kernel function. Since the Gaussian radial kernel function has the advantage of being highly localized and applicable to a wide variety of sample sets, the Gaussian kernel function is chosen to map the vectors in the initial space. The expression of the Gaussian kernel function is shown in Eq. (16).

$$K(x, x_k) = \exp\left(-\frac{\|x - x_k\|^2}{2\sigma^2}\right) \tag{16}$$

The kernel function can be mapped to an infinite-dimensional high-dimensional space, and the samples in

the low-dimensional space mapped using the Gaussian kernel function are necessarily linearly divisible in the high-dimensional space, facilitating computation. Although SVMs are better in prediction, they can be optimized in terms of parameter selection [19]. Several important parameters in the SVM model are the penalty factor D, the insensitivity factor ε and the kernel function width δ^2 [20]. Among them, D may lead to regression failure when it is large, D may lead to regression error when it is small; ε is prone to overfitting when it is large and underfitting when it is small; and δ^2 indicates the closeness of the data, if δ^2 is smaller, it means that the data is closer, indicating that the selection of appropriate D, ε and δ^2 plays a very important role in the performance of the model [21]. Therefore, the model uses the global search property of the GA to optimize the three important parameters of the SVM regression to improve the approximation ability as well as the adaptiveness of the model to obtain the GA-SVM model.

3.2. GA-SVM based robot control system research

It is very difficult to implement completely accurate gait planning for different environments for a hydraulic quadruped robot, and there is a nonlinear motion relationship between the upper body position and the ZMP trajectory that satisfies the ZMP condition [3]. Therefore, the study incorporated the optimized GA-SVM model into the gait control system, and the flow chart of the gait control system is shown in Fig. 5.

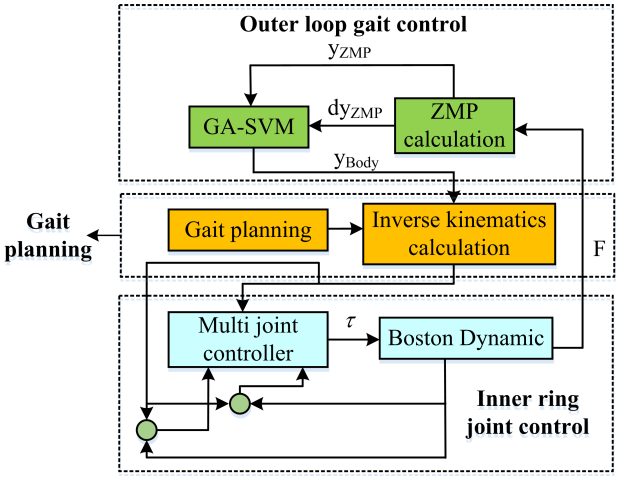

Fig. 5. Flow chart of gait control system.

Quadruped robots usually walk according to gait planning by controlling a multi-joint controller to drive each internal joint [22]. However, in practical situations, there are obstacles or other unforeseen circumstances, and if the robot still walks according to the gait planning, it will inevitably fall or deviate from its initial direction. And the above gait control system is designed on the basis of maintaining stability in human walking, which can evaluate whether the robot is balanced by the signal from the bottom of the robot's feet, and then adjust the robot's posture by the obtained signal to maintain the overall balance of the robot. The specific process is as follows: the first step is to process and filter the foot signals collected from the sensors, then input the filtered data into the GA-SVM regression model for regression modeling, and after modeling, obtain the dynamic relationship to determine whether the robot is stable or not. The final step is to place the obtained model in the gait control system designed in the study, and use this control system and human adjustment of the robot's upper limb position to enable the robot to walk stably. The gait samples of the nonlinear motion relationship between the upper limb and the ZMP trajectory of the robot satisfying the ZMP stability condition were collected as follows. First, the ZMP variation and ZMP values and the upper limb position sample data in the cycle of ZMP stability are collected, and the expression of the functional relationship between the three is shown in Eq. (17).

$$y_{trunk} = f(y_{ZMP}, dy_{ZMP}) \tag{17}$$

In Eq. (17), y_{trunk} represents the upper body position, y_{ZMP} represents the ZMP value, and dy_{ZMP} represents the ZMP change value. The obtained y_{ZMP} and dy_{ZMP} are fed into the GA-SVM model as input data, and the output y_{trunk}. In the process of model operation, due to its own sparsity, only a small number of sample coefficients (α_k, α_k^*) is not 0. At this time, the samples corresponding to (α_K, α_k^*) $(y_{ZMP}^{(k)}, dy_{ZMP}^{(k)}, y_{trunk}^{(k)})$ are support vectors. If the number of support vectors is g, then the gait regression function of this robot is shown in Eq. (18).

$$f(x) = \sum_{k=1}^{g} (\alpha_k^* - \alpha_k)\, l\left[\left(y_{ZMP}^{(k)}, dy_{ZMP}^{(k)}\right), \left(y_{ZMP}^{(k)}, dy_{ZMP}^{(k)}\right)^T\right] + c \tag{18}$$

In Eq. (18), α_k^* and α_k are the multipliers that need to continue to learn, and the minimum optimization algorithm is used to solve the multipliers in a loop iteratively, so that the multipliers all satisfy the KKT condition. The results are substituted into Eq. (17) to obtain the desired nonlinear model.

3.3. SVM parameter optimization based on GA

Genetic algorithm can utilize its strong search ability to optimize the parameters of SVM and improve the performance of the model. The optimization process of genetic

algorithms is actually to iterate over individuals in the algorithm, judge the fitness of the iterated individuals, and select the individuals with the best fitness for output. The specific process is shown in Fig. 6.

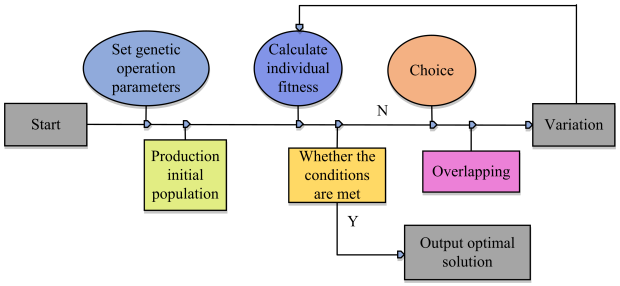

Fig. 6. Specific process diagram of genetic algorithm.

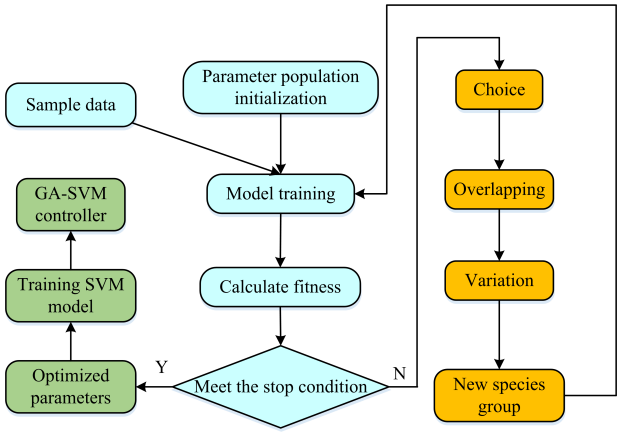

Fig. 7. Flow chart of model parameter optimization.

As shown in Fig. 6, the first step in genetic algorithms is to initialize the population, generate different individuals, and then use the fitness function to judge the fitness of individuals to determine their adaptability to the environment. Select individuals with good adaptability to the environment to enter the next generation for iteration, so that the algorithm gradually obtains the optimal solution during the iteration process. Select individuals from the population according to corresponding probabilities for crossover and mutation operations to generate new individuals for the next generation, and finally determine whether the output results meet the requirements. If it meets the requirements, the algorithm ends and outputs the optimal individual. If it does not, the algorithm is rerun from the calculation of individual fitness. Therefore, the model utilizes the global search characteristics of genetic algorithms to optimize the three important parameters of SVM regression, improve the approximation ability and adaptability of the model, and obtain a GA-SVM model. The GA-SVM model is a SVM in which a GA is introduced to optimize its parameters, as shown in the flowchart in Fig. 7.

After each round of training regression of the SVM, a GA is used to find more suitable values of C, ε, and δ^2. This process is called optimization of the population, and its criterion is the fitness function. Since the object of the regression is a nonlinear function, the MSE of the predicted and sampled values is used as the fitness function of the GA, and its expression is shown in Eq. (19).

$$MSE = \frac{\sum_{i=1}^{k}\left(y_{\text{thwik}}^{(l)} - f\left(C, \varepsilon, \sigma^2, \left(y_{\text{ZZMP}}^{(i)}, dy_{ZMP}^{(l)}\right)\right)\right)^2}{k} \tag{19}$$

In Eq. (19), k represents the number of test samples, $y_{trunk}^{'(i)}$ is the actual output value, and f is a function of the support vector regression. Although the optimization with this method population is simple, the possibility of overfitting in the process is higher, so the cross-validation method is also used to optimize the parameter values. The S samples were divided equally into N groups, and then the SVM regression function was built using the parameter set of the SVM and N-1 groups of samples, leaving one group of sample sets for testing. This operation is repeated several times, and the mean of the mean errors obtained is the desired population adaptation value. The expression of this procedure is shown in Eq. (20).

$$\begin{aligned} \min \quad \text{Fitness} &= cv\left(C, \sigma^2, \varepsilon, MSE\right) \\ &= \frac{1}{N}\sum_{i=1}^{N}\frac{1}{k}\sum_{j=1}^{k}\left(y_{trimk}^{((i)} - f\left(C, \sigma^2, \varepsilon, \left(y_{ZMP}^{(j)}, dy_{\text{ZZMP}}^{(j)}\right)\right)\right)^2 \end{aligned} \tag{20}$$

The k in Eq. (20) is $\frac{S}{N}$, and N denotes the number of training sets. The general optimization search steps of GAs are a series of operations such as encoding, fitness, and variation of parameters, and finally the optimal set of parameters is derived. However, for the sake of population diversity and to improve the model accuracy, the study uses both the crossover probability p_c and the variation probability p_m, and the expressions of the two probabilities are shown in Eq. (21).

$$\begin{aligned} p_c &= \frac{k_1}{f_{max} - f_{avg}} \\ p_m &= \frac{k_2}{f_{max} - f_{avg}} \end{aligned} \tag{21}$$

In Eq. (21), f_{max} represents the maximum adaptation value obtained in the last cycle, while f_{avg} represents the average adaptation value obtained in the last cycle. By applying the two probabilities to the crossover and variation

process in the GA, the diversity of its population is maintained and compared with the constant value probability of the GA in the simulation experiment session.

4. Ga-svm algorithm performance comparison and system result analysis

4.1. Analysis of GA-SVM algorithm test results

The determination of the SVM parameters has a large impact on its performance as well as accuracy, so it is necessary to find the best parameters before training the SVM model. The study uses a GA to find the best C and g parameters, and uses a combination of programming experiments and a 3-fold cross-validation method to validate the accuracy of the training set for the selected parameters. When using the GA to determine the optimal parameters, the penalty coefficient C is taken in the range of [0.1,1000], the kernel function parameter g is taken in the range of [0.01,100], and the crossover probability and variance probability are 0.7 and 0.1, respectively. The results of the GA adaptation change curve are shown in Fig. 8.

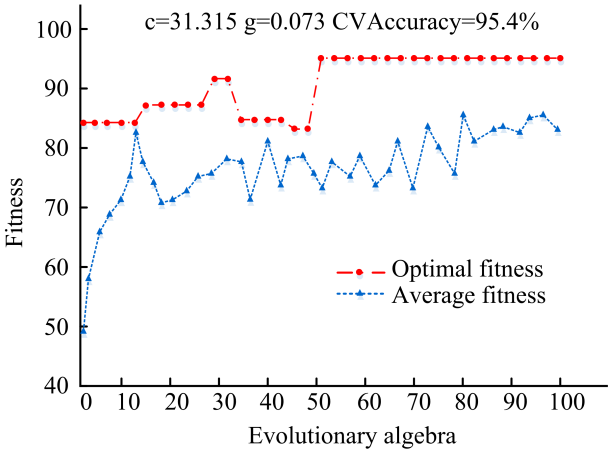

Fig. 8. Fitness curve of GA.

From Fig. 8, we can see that both the average fitness curve and the best fitness curve show an overall increasing trend with the increase of evolutionary generations, and finally the fitness of the best fitness curve is stabilized at 95.6. The optimal penalty coefficient of C is 31.315 and the optimal kernel function parameter of g is 0.073. The accuracy of the model classification reaches 95.4% at this time. Analyzing the optimal fitness curve in the figure, we can find that the curve tends to grow and stabilize rapidly when the population evolution generation is 50, which indicates that the model has good convergence in the process of evolution. The GA-SVM model is obtained by substituting the determined optimal parameters into the initial SVM. GA-SVM model was trained by using the training set, and a GA-SVM model was obtained after training. Then the accuracy of the model is verified by the test set until the prediction accuracy of the model reaches the expected target. The test results of GA-SVM model in the training set and test set are shown in Fig. 8.

The a-plot in Fig. 8 shows the prediction accuracy curve of the GA-SVM model in the training sample. From the a-plot, it can be observed that the value of the squared correlation coefficient of the model in the training set R^2 is 0.9215 and the value of the mean squared error MSE is 0.00632. The magnitude of these two values indicates that the GA-SVM model has a high accuracy in the training set. Since the value of the squared correlation coefficient is greater than 0.85, it also indicates that the GA-SVM model has a better generalization ability in the training set. The b-plot in Fig. 8 shows the prediction accuracy curve of the GA-SVM model in the test sample. From the b-plot, we can find that the value of the squared correlation coefficient of the GA-SVM model in the test sample R^2 is 0.8712, which is lower than the value of the squared correlation coefficient of the GA-SVM model in the test sample MSE; we can also find that the value of the mean squared error of the GA-SVM model in the test sample R^2 is 0.01312, which is higher than the value of the value of the mean squared error of the GA-SVM model in the test sample MSE is higher than that of the GA-SVM model in the training sample. Although the performance evaluation metric values of the GA-SVM model in the test sample are not as good as those in the training sample, the performance metric values of R^2 and MSE are also better than those of the conventional model. In addition, the research also compares the performance of GA-SVM algorithm with GA-BP algorithm, SVM algorithm, and BP algorithm to analyze the specific effects of GA-SVM algorithm. The error results of the four algorithms are shown in Fig. 10.

As shown in Fig. 10, the two graphs (a) and (b) are graphs of the average absolute error versus training times and the maximum absolute percentage error versus training times for the four algorithms. From Fig. 10(a), it can be seen that the GA-SVM algorithm has the fastest convergence speed among the three algorithms, with an average absolute error of 0.00152 after stabilizing; The average absolute error of the GA-BP algorithm after stabilizing is 0.00635; The average absolute error of the BP algorithm after stabilizing is 0.00817. From Fig. 10(b), it can be seen that the maximum absolute percentage error of the GA-BP algorithm tends to stabilize at 4.8% when the number of training times is 140; The convergence speed of APSO-BP algorithm is second only to GA-BP algorithm, and the max-

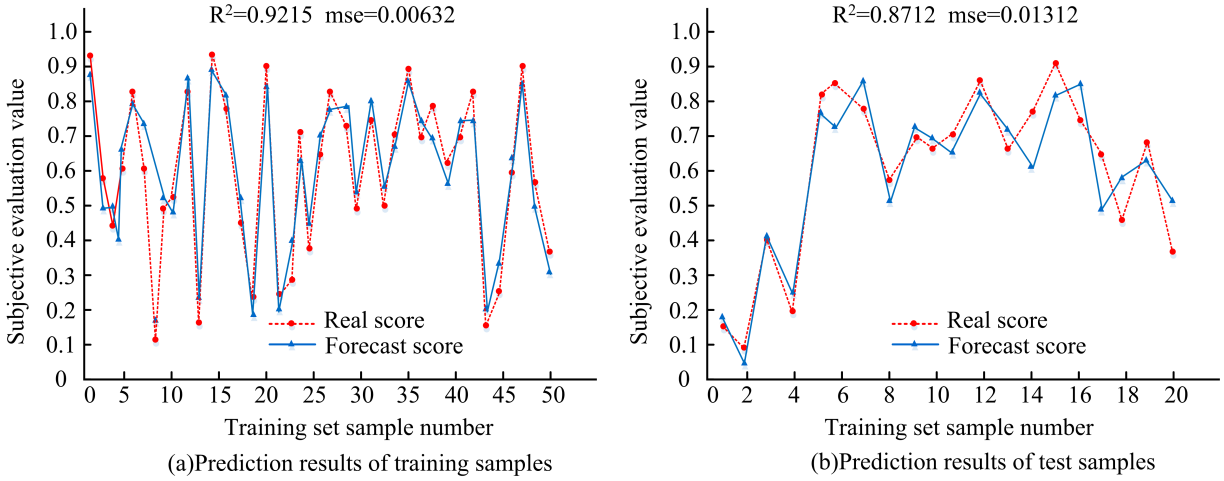

Fig. 9. Test results of training set and test set.

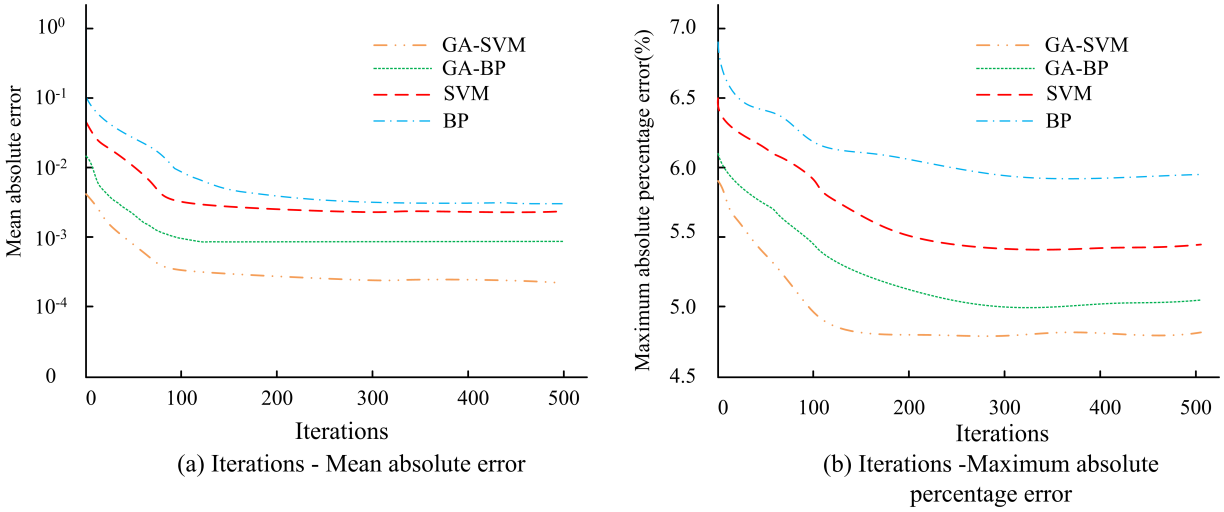

Fig. 10. Error Comparison Results of Four Algorithms.

imum absolute percentage error tends to stabilize at 5.3% when the number of training times is 210; The convergence speed of BP algorithm is the slowest, and the maximum absolute percentage error tends to stabilize at 6.1% when the number of training times is 310. The above results indicate that the GA-SVM model has high prediction accuracy in the training set and has strong generalization ability. Therefore, the GA-SVM model can be embedded in the gait control system of the quadruped robot to reasonably control the gait of the quadruped robot.

4.2. Analysis of experimental results of gait control system simulation

The gait control system designed by the study was applied to the common hip joint mutation of quadruped robots, and three groups with different gait parameters only and similar other parameters were set up for comparison. The gait parameters of the first group were 0.14 m and 0.875 cm; the gait parameters of the second group were 0.16 m and 1.037 cm; and the gait parameters of the third group were 0.18 m and 1.218 cm. For the sake of compare the performance of the gait control system, the neural network model, the simple SVM model and the GA optimized SVM model were used in the gait regression session. dy_{ZMP} For the sake of compare the performance of this gait control system, the neural network model, the simple SVM model and the GA optimized SVM model were used to regress the gait of three different sets of gait parameters. $ZMP(y_{ZMP})$, ZMP and y_{trunk} were sampled periodically in the forward direction of the quadruped robot. The comparative results of the regression effects of the three models are shown in Table 1.

Table 1. Comparison results of regression effects of three models.

Learning model		NN	SVM	GA-SVM
Step length 0.14 m, step height 0.875 cm	Training MSE (10^{-4})	11.235	12.672	9.391
	Test MSE (10^{-4})	1.284	0.481	0.273
Step length 0.16 m, step height 1.037 cm	Training MSE (10^{-4})	10.891	11.325	9.021
	Test MSE (10^{-4})	1.118	0.428	0.255
Step length 0.18 m, step height 1.218 cm	Training MSE (10^{-4})	10.521	10.983	8.934
	Test MSE (10^{-4})	1.028	0.385	0.237

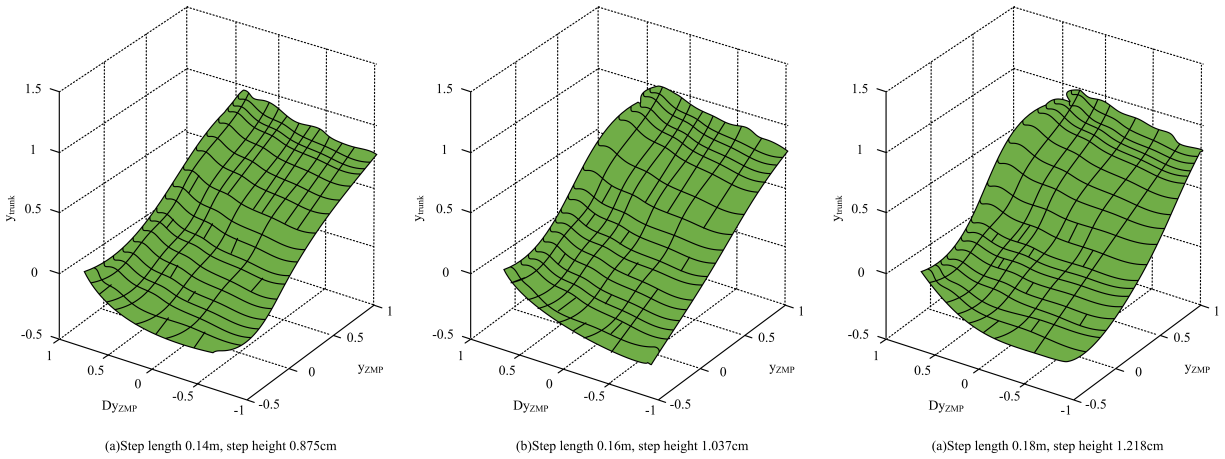

Fig. 11. Control model diagram after optimization under different parameters.

From Table 1, it can be seen that among the three learning methods, the MSE of the training set and the MSE of the test set of the SVM method optimized by the GA are lower than those of the traditional SVM method and the neural network training method. In the group with the gait parameters of 0.14 m and 0.875 cm, the training set MSE of the GA optimized SVM method is 9.391, which is lower than that of the traditional SVM method (12.672) and the neural network training method (11.235); the test set MSE of the GA optimized SVM method is 0.273, which is lower than that of the traditional SVM method. In the group with gait parameters of 0.16 m and 1.037 cm, the MSE of the training set of the GA optimized SVM method is 9.021, which is lower than that of the traditional SVM method (11.325) and the neural network training method (10.891). The MSE of the test set of the optimized SVM method is 0.255, which is lower than that of the traditional SVM method (0.428) and the neural network training method (1.118); in the group with the gait parameters of 0.18 m and 1.218 cm, the MSE of the training set of the optimized SVM method is 8.934, which is lower than that of the traditional SVM method (10.983). This result indicates that the MSE of the three learning methods decreases with the increase of step length and step height, which means the performance is better. In addition to this by comparing the MSE values of the three groups, it is observed that the GA optimized SVM method performs much better than the neural network and traditional SVM methods. For the sake of further test the performance of the method designed for the study, the SVM method optimized by GA was used to optimize the control system for three groups with different gait parameters, and the control model diagram obtained after optimization is shown in Fig. 11.

From the control model plots of the three gait parameters in Fig. 11, it can be seen that the values of $ZMP(y_{ZMP})$, ZMP changes $dy(ZMP)$ and the upper body motion trajectory y_{trunk} in the forward direction of the quadruped robot are in the smaller range for the three different sets of gait parameters. This result shows that after the optimization of the gait parameter control system by the improved SVM algorithm, the system is better controlled. In addition, it can be seen that the trend of the influence of different gait parameters on the gait control system is better as the step length and step height increase, but the overall difference is not significant. In addition, in order to analyze the practical application effects of the proposed quadruped robot, the study also conducted comparative experiments with traditional quadruped robots, using the walking route accuracy score and stability score as evaluation indicators for performance analysis. The comparison results of two dif-

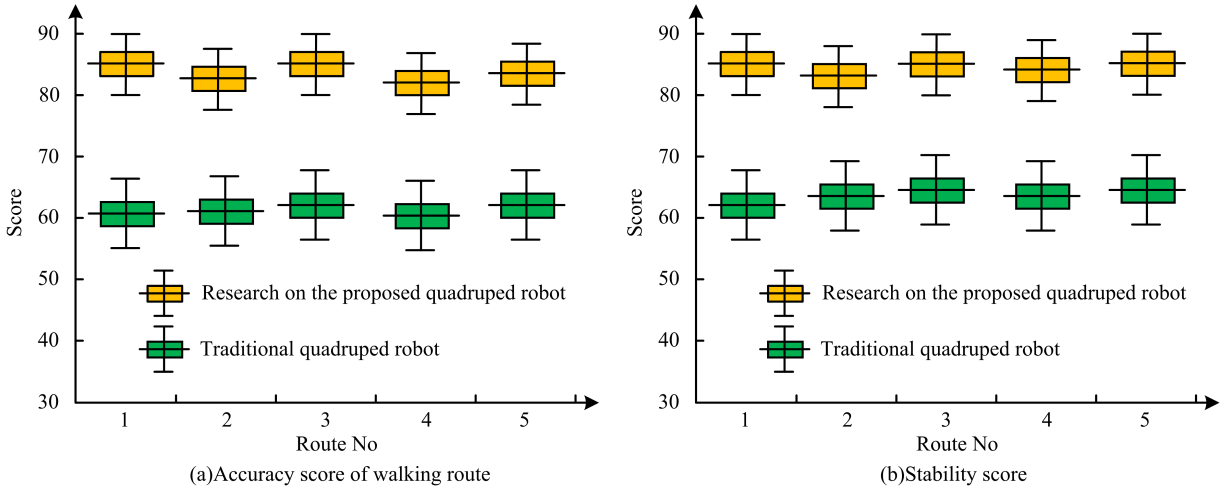

Fig. 12. Comparative results of two different quadruped robots in five different routes.

ferent quadruped robots in five different routes are shown in Fig. 12.

As shown in Fig. 12, the results can be seen from the comparison of the walking route accuracy scores and stability scores of two quadruped robots in five different routes. The accuracy score of the quadruped robot's walking path proposed in the study is generally higher than that of traditional face recognition models, and its average score is 83 points. In addition, it can be seen from Fig. 12(b) that the overall stability score of the quadruped robot proposed in the study is also higher than that of the traditional quadruped robot, and its average stability score is 84 points. From this result, it can be concluded that the performance of the proposed quadruped robot is superior to that of traditional quadruped robots. The above results show that the SVM algorithm optimized by GA is much better than the traditional model in terms of learning ability, and the control system trained by this algorithm has better control effect.

5. Conclusion

Although there are a number of quadruped robots designed in China today, there is a considerable gap between them and foreign designed quadruped robots in terms of their gait walking as well as their adaptability. A new GA-SVM algorithm model is designed by combining SVM and GA to improve the gait control of domestic quadruped robot.GA-SVM model was used to determine the optimal parameters of SVM, and the optimal penalty coefficient was 31.315 and the optimal kernel function parameter was 0.073. Then GA-SVM model was compared with NN model and traditional SVM model in the training and test sets to compare the model performance. The experimental results show that the square correlation coefficients of GA-SVM model in training set and test set are 0.9215 and 0.8712, respectively, and the MSE of GA-SVM model in training set and test set are 0.00632 and 0.01312, respectively, which are superior to the other two models. Through the simulation test of the control system, it was found that the MSE of the GA-SVM model was 9.391, 9.021, and 8.934 for the three different sets of gait parameters. Both outperformed the NN model and the traditional SVM model. The results indicate that the combination of GA and SVM algorithm can improve the performance of the SVM model, and the optimized SVM model can improve the control effect of the control system when applied to the gait control of a quadruped robot. Although the model method designed in the study can effectively improve the control effect of the gait control system, the research tests only studied the situation when the robot was walking straight, and did not consider the robot's lateral motion, etc. The subsequent research direction is to apply the model to the control of multi-directional motion of the quadruped robot.

Acknowledgement

There is no funding used for this research.

References

[1] Z. Jiang, B. Shi, F. Du, B. Xue, M. Lei, Z. Yang, and H. Sun, (2022) *"Intelligent Plant Cultivation Robot Based on Key Marker Algorithm Using Visual and Laser Sensors"* **IEEE Sensors Journal 22**(1): 879–889. DOI: 10.1109/JSEN.2021.3130607.

[2] G. Dwyer, F. Chadebecq, M. T. Amo, C. Bergeles, E. Maneas, V. Pawar, E. V. Poorten, J. Deprest, S. Ourselin, P. De Coppi, T. Vercauteren, and D. Stoyanov, (2017) *"A Continuum Robot and Control Interface for Surgical Assist in Fetoscopic Interventions"* **IEEE Robotics and Automation Letters** *2*(3): 1656–1663. DOI: 10.1109/LRA.2017.2679902.

[3] H. Huang, J. Zhang, B. Xu, G. Liu, Q. Luo, and X. Wang, (2021) *"Topology optimization design of a lightweight integrated manifold with low pressure loss in a hydraulic quadruped robot actuator"* **Mechanical Sciences** *12*(1): 249–257. DOI: 10.5194/ms-12-249-2021.

[4] J. Shang, D. Fang, Z. Luo, R. Wang, X. Li, and J. Yang, (2021) *"Design and analysis of a hydraulic drive downhole traction in-pipe robot based on flexible support structure"* **Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science** *235*(1): 18–27. DOI: 10.1177/0954406220914319.

[5] H. Dawid and M. Kopel, (1998) *"On economic applications of the genetic algorithm: A model of the cobweb type"* **Journal of Evolutionary Economics** *8*(3): 297–315. DOI: 10.1007/s001910050066.

[6] H. C. Husada and A. S. Paramita, (2021) *"Analisis Sentimen Pada Maskapai Penerbangan di Platform Twitter Menggunakan Algoritma Support Vector Machine (SVM)"* **Teknika** *10*(1): 18–26.

[7] H. Chen, J. Duan, R. Yin, V. V. Ponkratov, and J. W. G. Guerrero, (2021) *"Prediction of penetration rate by Coupled Simulated Annealing-Least Square Support Vector Machine (CSA_LSSVM) learning in a hydrocarbon formation based on drilling parameters"* **Energy Reports** *7*: 3971–3978. DOI: 10.1016/j.egyr.2021.06.080.

[8] J. Zhang, J. Sun, and H. He, (2022) *"Clustering Detection Method of Network Intrusion Feature Based on Support Vector Machine and LCA Block Algorithm"* **Wireless Personal Communications** *127*(1): 599–613. DOI: 10.1007/s11277-021-08353-y.

[9] J. Song, C. Yu, and S. Li, (2021) *"Continuous prediction of onsite PGV for earthquake early warning based on least squares support vector machine"* **Acta Geophysica Sinica** *64*(2): 555–568. DOI: 10.6038/cjg2021O0193.

[10] P. Bose, S. Dutta, V. Goyal, and S. Bandyopadhyay, (2021) *"Leaf Diseases Detection of Medicinal Plants Based on Support Vector Machine Classification Algorithm"* **Journal of Pharmaceutical Research International** *33*(42A): 111–119.

[11] R. Farkh, H. Marouani, K. Al Jaloud, S. Alhuwaimel, M. T. Quasim, and Y. Fouad, (2021) *"Intelligent autonomous-robot control for medical applications"* **Computers, Materials and Continua** *68*(2): 2189–2203. DOI: 10.32604/cmc.2021.015906.

[12] T. Togias, C. Gkournelos, P. Angelakis, G. Michalos, and S. Makris. "Virtual reality environment for industrial robot control and path design". In: ***100***. Cited by: 11; All Open Access, Gold Open Access. 2021, 133–138. DOI: 10.1016/j.procir.2021.05.021.

[13] I. Galushko, V. Salmina, and G. Makaryants, (2019) *"DEVELOPMENT OF A TEST BENCH FOR TESTING THE UNDERWATER ROBOT CONTROL SYSTEM WITH VARIABLE GEOMETRY OF THE BODY"* **Journal of Dynamics and Vibroacoustics** *5*(3): 6–13.

[14] G. Kunert, T. Pawletta, and S. Hartmann, (2020) *"Reduction of Complexity in Q-Learning a Robot Control for an Assembly Cell by using Multiple Agents"* **SNE Simulation Notes Europe** *30*(3): 117–124.

[15] T. Mikada, T. Kanno, T. Kawase, T. Miyazaki, and K. Kawashima, (2020) *"Suturing support by human cooperative robot control using deep learning"* **IEEE Access** *8*: 167739–167746. DOI: 10.1109/ACCESS.2020.3023786.

[16] Z. Feng, (2021) *"An image detection method based on parameter optimization of support vector machine"* **International Journal of Circuits, Systems and Signal Processing** *15*: 306–314. DOI: 10.46300/9106.2021.15.35.

[17] E. Samsudin, (2021) *"Modeling Student's Academic Performance During Covid-19 Based on Classification in Support Vector Machine"* **Turkish Journal of Computer and Mathematics Education (Turcomat)** *12*(5): 1798–1804.

[18] D. Gunawan, D. Riana, D. Ardiansyah, A. F., and S. Alfarizi, (2020) *"Komparasi algoritma support vector machine dan naïve bayes dengan algotima genetika pada analisis sentimen calon gubernur jabar 2018-2023"* **Jurnal Teknik Kompute** *6*(1): 121–129.

[19] L. Luthfiana, J. C. Young, and A. Rusli, (2020) *"Implementasi Algoritma Support Vector Machine dan Chi Square untuk Analisis Sentimen User Feedback Aplikasi"* **Jurnal Ultimatics** *12*(2): 125–126.

[20] R. Farkh, H. Marouani, K. A. Jaloud, S. Alhuwaimel, and Y. Fouad, (2021) *"Intelligent Autonomous-Robot Control for Medical Applications"* **Computers, Materials and Continua** *68*(2): 2189–2203.

[21] S. Lee and J. H. Kim, (2020) *"Improvement of the Support Vector Machine-based Monte Carlo Simulation for Calculating Failure Probability"* **Transactions of the Korean Society of Mechanical Engineers, A *44***(4): 269–279. DOI: 10.3795/KSME-A.2020.44.4.269.

[22] J. Liu, P. Wang, M. Li, W. Guo, F. Zha, L. Sun, and P. Zheng, (2020) *"A multiplicative noises and additive correlated noises cubature kalman filter and its application in quadruped robot"* **IEEE Access *8***: 162290–162301. DOI: 10.1109/ACCESS.2020.3021494.