GS-PKNN: An Efficient and High-Fidelity Mobility Prediction Method for Unmanned Ground Vehicles

Chen Hua[1,2], Runxin Niu[1], Chunmao Jiang[1,2], Biao Yu[1], Hui Zhu[1], and Bichun Li[1]

***Abstract*— To avoid unmanned ground vehicles being obstructed by deformed terrain in off-road, effective vehicle mobility analysis is required. However, the computational complexity of existing mobility analysis methods, such as discrete element analysis, poses significant challenges when applied to large-scale terrains. To address this problem, we propose an efficient and high-fidelity vehicle mobiliy prediction method for a large-scale terrain. Initially, precise terrain models are constructed employing Gaussian sampling (GS), thereby serving as optimal inputs for the mobility simulation. Subsequently, we introduce a co-simulation method based on a multi-body dynamics model and discrete element analysis to obtain high-fidelity vehicle mobility data on sampled terrains. Following that, the mobility data is utilized to train a PSO-kriging neural network (PKNN), enabling accurate predictions of the global mobility map. Through rigorous simulation experiments, the proposed method (GS-PKNN) demonstrates its remarkable effectiveness.**

***Index Terms*—Mapping, machine learning, vehicle mobility prediction, simulation.**

I. INTRODUCTION

With the continuous exploration and exploitation of Earth and extraterrestrial resources, unmanned ground vehicles (UGVs) are frequently deployed in diverse and challenging environments, ranging from beaches and deserts to celestial bodies like the Moon and Mars [1], [2]. However, the low shear strength of soil terrains poses a significant challenge as shear damage can impede the movement of UGVs in deformed terrains. Consequently, accurate mobility prediction becomes imperative prior to field missions to ensure successful navigation and mobility.

One of the prominent mobility prediction models in the field is the NATO Reference Mobility Model (NRMM) [3], which relies on empirical and semi-empirical techniques, including Bekker's derived terramechanics model (BDTM) [4]. These models, which rely on experimental results, suffer from limited generalizability. However, with advancements in computer simulation technologies, high-fidelity simulation techniques are now being employed for vehicle mobility prediction [5]. Currently, the prevailing high-fidelity simulation approach is discrete element (DE) analysis, which involves calibrating the contact parameters between particles through field test simulations (such as triaxial and shear displacement tests) to approximate the mechanical response of actual soil [6], [7]. Building upon this, Recuero *et al*. [8] proposed a co-simulation framework integrating DE terrain and a vehicle multibody dynamics (MBD) model to capture vehicle mobility. Similarly, for large-scale terrains, McCullough *et al*. [9] presented a DE-based method for simulating and predicting vehicle mobility. However, it is important to note that DE analysis is computationally intensive and time-consuming. Obtaining an accurate mobility map for a terrain spanning hundreds of square kilometers typically requires several months of effort [10]. To expedite the simulation process, various methods have been proposed. These include the moving soil patch approach [11] and the data-driven accelerated simulation method for computational mechanics [12].

The aforementioned accelerated methods still require the simulation to traverse all terrain rasters when dealing with large-scale terrains. To address this challenge. Quann *et al*. [13] predicted the vehicle power consumption on different terrains based on Gaussian process regression. Mechergui *et al*. [14] developed a neural network model to predict the mobility map of the vehicle model feeded by Latin hypercube sampling (LHS) soil cone index (CI) and slope values.

This paper presents a methodological framework for enhancing the prediction efficiency of global mobility under a realistic terrain using machine learning techniques. The main contributions of this work are outlined as follows:

- We propose a Gaussian sampling method that utilizes multidimensional terrain parameters. This method enables the calibration of real sampled soil macro parameters to obtain accurate micro parameters required for DE simulation by support vector machine (SVM).
- We propose an enhanced neural network model (PKNN) with the goal of enhancing the efficiency of global mobility prediction. This is achieved by incorporating a particle swarm optimization (PSO) algorithm to calibrate the network parameters. Also, we introduce kriging semi-variance coefficients to enhance the network's physical information constraint.

II. PROBLEM FORMULATION

In this section, we present the framework of purposed method for global mobility prediction for the UGV in the large-scale off-road terrain, Fig. 1 illustrates the framework.

This work was supported by Dreams Foundation of Jianghuai Advance Technology Center (NO.2023-ZM01G002) and Youth Innovation Promotion Association of CAS under Grant Y2021115. *(Corresponding author: Biao Yu.)*

[1]Chen Hua, Runxin Niu, Chunmao Jiang, Biao Yu, Hui Zhu and Bichun Li are with Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei, Anhui 230031 China ba20168187@mail.ustc.edu.cn; rxniu@iim.ac.cn; sa252@mail.ustc.edu.cn; byu@hfcas.ac.cn; hzhu@iim.ac.cn; bcli@iim.ac.cn

[2]Chen Hua and Chunmao Jiang are also with with Department of Automation, University of Science and Technology of China, Hefei, Anhui 230026 China

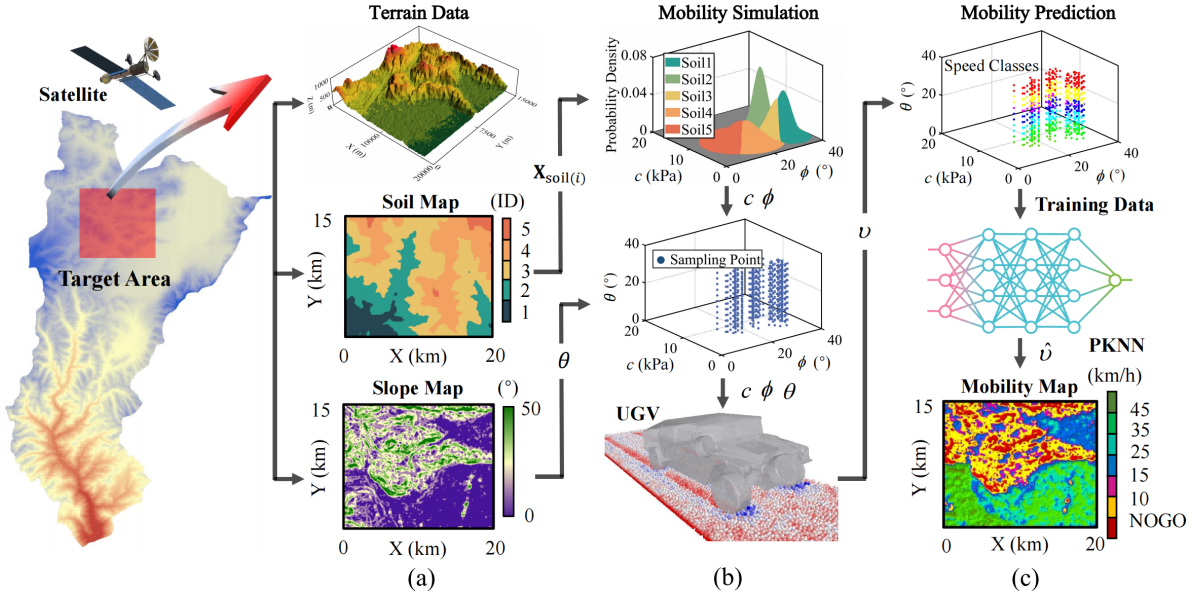

Fig. 1. Framework of global mobility prediction method for the UGV over large off-road terrain.

Fig. 1(a) shows the framework starts with the target satellite terrain map from ARCGIS/ENVI database [15]. The terrain rasters set $\mathbf{M_x}$ is selected as the target area, the attributes of each terrain raster $\mathbf{x}_{(i)} \in \mathbf{M_x}$ contain soil data $\mathbf{x}_{\text{soil}(i)}$, and slope data $\theta_{(i)}$, where $\mathbf{x}_{\text{soil}(i)}$ is a function of soil cohesion coefficient c and friction angle ϕ. $\theta_{(i)}$ is calculated by terrain elevation data [16], [17]. As shown in Fig. 1(b), in order to use small sample data to train an accurate mobility prediction model, c and ϕ are fitted to the Gaussian distribution function by statistical analysis. Based on the fitting results, c and ϕ with the high probability are selected and combined with different $\theta_{(i)}$ for terrain DE-based modeling. Afterwards, $\mathbf{x}_{\text{soil}(i)}$, $\theta_{(i)}$ and the UGV-related MBD parameters $\mathbf{H}$ are fed into the co-simulation method to predict the UGV's mobility at i as

$$v_{(i)} = \mathcal{J}\left(\mathbf{H}, \mathbf{x}_{\text{soil}(i)}, \theta_{(i)}\right) \tag{1}$$

where $v_{(i)}$ represents the vehicle mobility (maximum attainable speed). The MBD model used in the study is based on a HMMWV [18]. Finally, all sampled terrains are traversed by the simulation to obtain a dataset of terrain parameters corresponding to the mobility, which is fed to the PKNN algorithm to obtain the global mobility map, as shown in Fig. 1(c).

III. Mobility Simulation Based on Terrain Gaussian Sampling

A. Terrain data sampling based on Gaussian distributions

The open-source terrain digital elevation of the target area is provided by Geospatial Data Clouds [19], which has a terrain extent of 300 km^2 and contains 5490 terrain rasters. Data of the soil types are obtained from Soil Information System of China (SISChina) [20]. Table I shows the statistics of soil parameters corresponding to different soils, μ is mean value and σ is standard deviation.

TABLE I
Statistics of Parameters for Different Soils.

Soil ID	c (kPa)		ϕ (°)	
	μ	σ	μ	σ
Soil 1	6.8	1.2	33.9	5.2
Soil 2	11.2	2.1	31.0	4.5
Soil 3	7.6	1.5	32.8	6.5
Soil 4	7.8	2.0	20.0	4.1
Soil 5	8.1	1.9	18.4	3.5

Since two soil parameters are selected as factors affecting vehicle mobility, the soil parameters are fitted to a multi-dimensional mixed Gaussian model for weighted sampling, selecting the parameters with high probability can avoid the invalid data which has little effect on the prediction results.

According to the statistical results in Table 1, we can obtain $c_i \sim (\mu_{c(i)}, \sigma_{c(i)})$ and $\phi_i \sim (\mu_{\phi(i)}, \sigma_{\phi(i)})$. Thus to fit both variables, a Gaussian mixture model is used to fit each soil, and the different soil parameters $\mathbf{x}_{\text{soil}(i)}$ satisfy

$$\begin{cases} f^j\left(\mathbf{x}_{\text{soil}(i)}\right) \in (0,1) \\ \sum_1^n f^j\left(\mathbf{x}_{\text{soil}(i)}\right) = 1 \end{cases} \tag{2}$$

where $f^j\left(\mathbf{x}_{\text{soil}(i)}\right)$ is the probability corresponding to terrain raster i under a given soil ID j (j=1, 2, ..., 5). n is the number of all terrain rasters under soil ID j. Based on the multi-parameter Gaussian mixture model, we fit the soil parameters under different soil IDs to a Gaussian distribution function, and the fitting equation is

$$f^j\left(\mathbf{x}_{\mathbf{soil}(i)}\right) = \frac{1}{\left(\sqrt{2\pi}\right)^2 \sigma_{z(i)}} \mathrm{e}^{-\mathbf{u}_i^2/2} \tag{3}$$

where $\sigma_{z(i)} = \sigma_{c(i)}\sigma_{\phi(i)}$. $\mathbf{u}_i^2$ in equation (3) is

$$\mathbf{u}_i^2 = \mathbf{u}_i^{\mathrm{T}}\mathbf{u}_i = \mathbf{\Gamma}_i\mathbf{\Theta}_i^{-1}\mathbf{\Gamma}_i^{\mathrm{T}} \tag{4}$$

where $\mathbf{\Gamma}_i = [c_i - \mu_{c(i)}, \phi_i - \mu_{\phi(i)}]$ and $\mathbf{\Theta}_i$ is the covariance matrix of variables c_i and ϕ_i , the expression is

$$\mathbf{\Theta}_i = \begin{bmatrix} \sigma^2_{c(i)} & 0 \\ 0 & \sigma^2_{\phi(i)} \end{bmatrix} \tag{5}$$

according to equation (3), Fig. 2 presents the Gaussian distribution functions of different soil parameters.

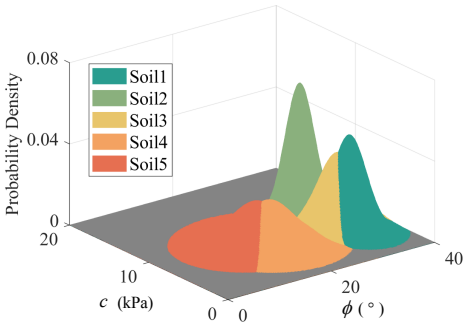

Fig. 2. Gaussian distribution of soil parameters for different soil IDs.

To improve the effectiveness of sampling, for each $\mathbf{x}_{\mathrm{soil}(i)} \in \mathcal{Q}^j$ ($\mathcal{Q}^j$ is the set of soil parameters under a certain soil ID), it is sampled according to its probability $f^j\left(\mathbf{x}_{\mathrm{soil}(i)}\right)$ as a weight. For terrain slope sampling, it is based on the kinematic limitations of the selected vehicle, so systematic sampling is used to sample the slope within 0-30°. To validate the proposed method, a final terrain sampling is performed, which involved a combination of soil parameters and slope values. Specifically, the selected sampling sizes are 220, 440, and 660.

B. Calibration of DE micro parameters

In order to ensure the accuracy of the simulated soil model in reproducing the mechanical response observed in real cases, it is necessary to calibrate the micro contact parameters $\{k_n, k_s, b_n, b_s, \mu\}$ of the DE soil model based on the real soil macro parameters $\{c, \phi\}$, here, $k_{n(s)}$ represents the normal elasticity coefficient and tangential elasticity coefficient of the DE, respectively, while $b_{n(s)}$ represents the normal damping coefficient and tangential damping coefficient of the DE, respectively. Additionally, μ represents the friction coefficient.

Based on the range of soil micro contact parameters summarized in [5], we sample 80 sets of micro parameters. These parameters are then used to simulate shear-displacement tests, by subjecting the shear box to various pressures (25 kPa, 50 kPa, and 100 kPa). We obtaine the corresponding shear stress-displacement curves under different pressure conditions.

The peak shear stresses of the soil at different pressures are selected as the vertical coordinate, while the vertical pressures are used as the horizontal coordinate. Fitted lines

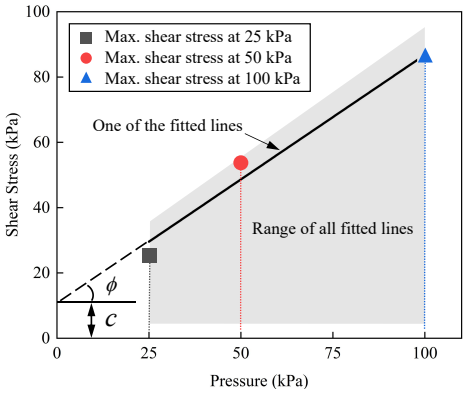

Fig. 3. Soil macro parameters fitted lines. The gray area represents the distribution interval of all fitted lines for 80 sets of experiments, and one set of results is shown for illustration, where the slope of the line is ϕ and the intercept with the vertical coordinate is c.

representing the linear relationship between shear stress and vertical pressure are then drawn, as shown in Fig. 3.

All the data are simulated to obtain training datasets for the machine learning model, including both soil micro and macro parameters. In this study, the SVM classification method is employed as the machine learning approach. The input data consisted of soil macro parameters $\{c, \phi\}$, while the output data consisted of soil micro parameters $\{k_n, k_s, b_n, b_s, \mu\}$. Out of the collected data, 20 sets are selected as test data to evaluate the model's performance. All parameters are normalized between [0,1].

The combination of hybrid kernel parameters for the best SVM performance are δ = 1.366, η = 0.934, r = 0.152, d = 2.231, h = 0.446. The mean absolute error (MAE) of SVM is 0.013, which indicates that the SVM fits the training data with high accuracy and the training can replace the DE traversal simulation process to achieve the calibration of soil micro parameters.

C. Co-simulation solution

In this section, we present a co-simulation solution to simulate the mobility of the UGV model on various terrains [21]. The process involves several steps. Firstly, the contact forces between the finite element (FE) tires and the DE terrain nodes are computed. These contact forces are then transmitted to the message passing interface (MPI) node of the tire to integrate the nonlinear FE in time. Subsequently, the calculated equivalent forces and torques at the center of the tire are sent from each FE tire node to the MPI node responsible for advancing the MBD of the UGV. This approach greatly enhances the simulation efficiency, as the MPI communication requires only a relatively small amount of time for data exchange. An overview of the co-simulation solution is illustrated in Fig. 4.

Based on the mentioned co-simulation solution, terrains are modeled with the desired DE soil contact parameters as well as the slope values. The vehicle is accelerated in a straight line to the steady-state maximum attainable speed at a certain acceleration, and the maximum attainable speed

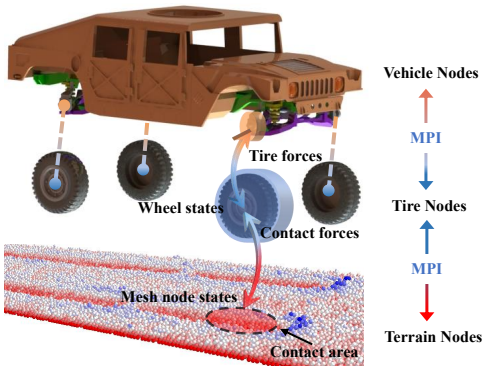

Fig. 4. Co-simulation solution for the mobility simulation.

of the vehicle in that state is recorded as the mobility. All sampled terrains are traversed in this manner. Fig. 5 presents the simulation run 660 times, and the results are classified into classes two, five, and seven.

IV. Mobility prediction model based on PKNN

In this section, the PKNN algorithm is proposed. This algorithm is improved by introducing the kriging semi-variance coefficient in the loss function, enabling the enhancement of physical information in the neural network.

A. Prediction model based on BPNN and PSO algorithm

The backpropagation neural network (BPNN) is a multilayer feedforward network trained using an error backpropagation algorithm [12]. We utilize the BPNN as the foundation for our improvement. The network model consists of an input layer, an output layer, and three hidden layers. Each hidden layer comprises 40 neurons, with each neuron having its weight $\mathcal{W}$ and bias $\mathcal{B}$. The functional function of each neuron is

$$\mathcal{U}_a^q = \mathcal{F}(\sum_{b=1}^{q-1} \mathcal{W}_{a,b}^{q,q-1} \mathcal{U}_b^{q-1} - \mathcal{B}_a) \tag{6}$$

where $\mathcal{U}_b^{q-1}$ is the output of the b-th node in the $q-1$ layer, $\sum_{b=1}^{q-1} \mathcal{W}_{a,b}^{q,q-1} \mathcal{U}_b^{q-1} - \mathcal{B}_a$ is the net input of the a-th node, $\mathcal{U}_a^q$ is the output of the a-th node in the q layer, $\mathcal{W}_{a,b}^{q,q-1}$ is the input weight of the b-th node $\mathcal{U}_b^{q-1}$ in the $q-1$ layer to the a-th node in the q layer, $\mathcal{B}_a$ is the bias of that node, and $\mathcal{F}(\cdot)$ is a nonlinear function. $\mathcal{F}(\cdot)$ is set as the hyperbolic tangent function.

In the PSO algorithm, each particle searches for food according to the initial position and velocity [6], the iterative process determines whether the current position is the optimal position according to the value of the pre-defined fitness function (the loss function of the neural network), each parameter consists of a species of n particles group, $\boldsymbol{p}_{\mathrm{g}} = [\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_n]^{\mathrm{T}}$ where the i-th particle is represented as a D-dimensional position vector $\boldsymbol{p}_i = [\boldsymbol{p}_{i1}, \boldsymbol{p}_{i2}, ..., \boldsymbol{p}_{iD}]^{\mathrm{T}}, i = 1, 2, .., n$, where $\boldsymbol{p}_{id} = [\mathcal{W}_{id}, \mathcal{B}_{id}]^{\mathrm{T}}, d = 1, 2, ..., D$, and its velocity vector is $\boldsymbol{v}_i = [\boldsymbol{v}_{i1}, \boldsymbol{v}_{i2}, ..., \boldsymbol{v}_{iD}]^{\mathrm{T}}$. The algorithm updates the velocity and position information of each particle as shown in equation (7).

$$\begin{cases} \boldsymbol{v}_{id}^{k+1} = s\boldsymbol{v}_{id}^{k} + \ell_1 \Re_1 \left(\tilde{\boldsymbol{p}}_{id}^{k} - \boldsymbol{p}_{id}^{k} \right) + \ell_2 \Re_2 \left(\tilde{\boldsymbol{p}}_{gd}^{k} - \boldsymbol{p}_{id}^{k} \right) \\ \boldsymbol{p}_{id}^{k+1} = \boldsymbol{p}_{id}^{k} + \boldsymbol{v}_{id}^{k+1} \end{cases} \tag{7}$$

where k is the number of iterations, $\tilde{\boldsymbol{p}}_{id}^{k}$ is the position of the optimal particle i for the k-th generation of individuals. $\tilde{\boldsymbol{p}}_{gd}^{k}$ is the position of the global optimal particle in the k-th generation. s is the inertia factor, $\Re_1$ and $\Re_2$ are random quantities taking values in the range [0,1], ℓ_1 and ℓ_2 are acceleration factors. The fitness value of the population is calculated based on the current position of the particle to determine if it is less than ε. If it is, the globally optimal particle position is outputted to obtain the weights and biases of the neural network.

B. Correction of loss function based on kriging variance function

The BPNN is a prediction model that relies on training data to make accurate predictions. However, it often exhibits limited generalization when applied to extrapolation scenarios [14]. The terrain properties of independent variables that influence mobility exhibit spatial correlation, as described by the semi-covariance coefficient in geostatistics [16], [17]. To address this, we incorporate this coefficient as a physical information constraint in the loss function during the training process. This constraint promotes the learning of a more generalizable model with a reduced need for extensive data samples. The loss function $L(\cdot)$ is determined by the semi-variance coefficient, the ground truth value, and the output limit, so the expansion is

$$\begin{aligned} L = & \frac{1}{2} \left\{ \frac{1}{n} \sum_{i=1}^{n} (\frac{\upsilon_i - \widehat{\upsilon}_i}{\bar{\upsilon}})^2 + \frac{1}{m} \sum_{k=1}^{m} (\frac{\gamma(h_k) - \gamma^*(h_k)}{\bar{\gamma}})^2 \right\} \\ & + \frac{1}{2} \left\{ \frac{1}{n_1} \sum_{i=1}^{n} [\max(\frac{\upsilon_{\max} - \widehat{\upsilon}_{\mathrm{i}}}{\upsilon_{\max} - \upsilon_{\min}}, 0)]^2 \right. \\ & \left. + \frac{1}{n_2} \sum_{i=1}^{n} [\max(\frac{\widehat{\upsilon}_{\mathrm{i}} - \upsilon_{\min}}{\upsilon_{\max} - \upsilon_{\min}}, 0)]^2 \right\} \end{aligned} \tag{8}$$

where $\gamma(h_k)$ and $\gamma^*(h_k)$ are the values of the variance functions obtained from the training data and the network output data, respectively. h_k is the interval distance of the k-th data (it is defined as the Euclidean distance between the input values). m is the maximum number of lags of data used to calculate the variance function (i.e., the number of data groupings at different h_k intervals), $\bar{\upsilon}$ and $\bar{\gamma}$ are the mean values of υ_i and $\gamma(h_k)$, respectively. $\upsilon_{\max}$ and $\upsilon_{\min}$ are the upper and lower limits of the predicted speeds, respectively. n_1 and n_2 are the number of data whose network output is larger than $\upsilon_{\max}$ or smaller than $\upsilon_{\min}$.

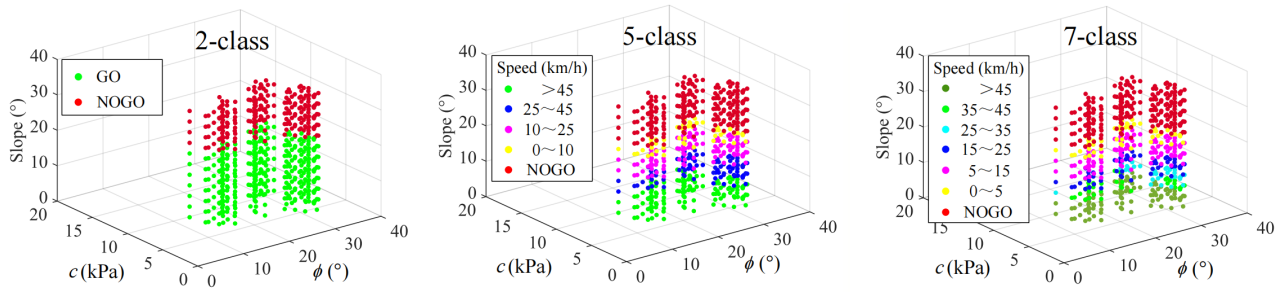

Fig. 5. Mobility simulation results with different number of classifications. The classification results are based on the maximum attainable speeds of the UGV.

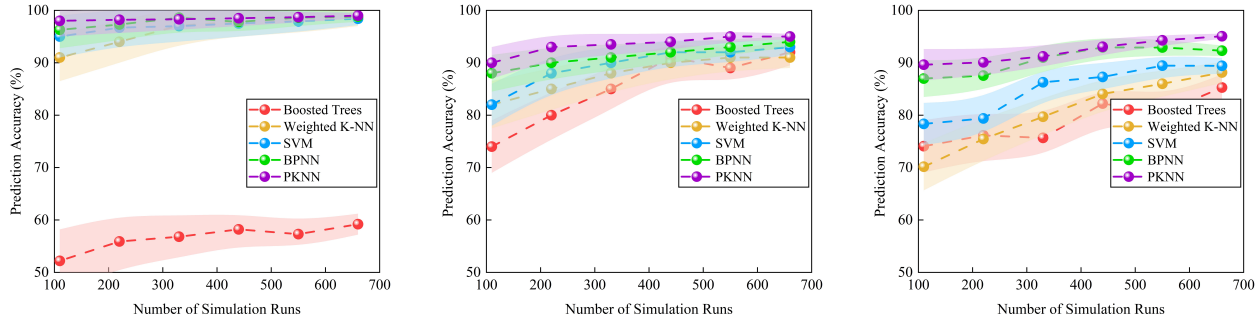

Fig. 6. Comparisons of the average prediction accuracy of different machine learning algorithms, the envelope region represents the result of fitting the standard deviation of the accuracy of multiple predictions. (a) 2-class. (b) 5-class. (c) 7-class.

V. Simulation Experiments and Analysis

A. Comparison of the training accuracy

The PKNN algorithm is compared with other machine learning algorithm such as booseted trees, weighted k-nearest-neighbor (K-NN), SVM and BPNN, in which 70% under different number of runs are used as the training data and 30% as the validation data, the results of the comparison of the average and standard deviation of the prediction accuracies are shown in Fig. 6. The prediction accuracy of various algorithms generally improves with increasing data volume. However, there is a general decrease in prediction accuracy as the speed classification increases. Based on the results, it can be concluded that both BPNN and PKNN exhibit significant advantages in terms of prediction accuracy. Furthermore, they show more stable prediction results based on the standard deviation. Notably, PKNN achieves a remarkable average prediction accuracy of 96% for seven speed categories within the dataset of 660 simulations (standard deviation less than 2%).

Based on the comparison results, the optimal hyperparameters of PSO in PKNN are the group number is 40, k =100, s = 0.729, ℓ_1 = ℓ_2 =1.495. Furthermore, PKNN and BPNN are separately compared for validation. The reliability evaluation indexes of the two algorithms are compared in Table II.

The results indicate that PKNN achieved a 5% decrease in MAPE compared to BPNN. Additionally, the smaller MSE value of PKNN demonstrates its better stability. Typically, when the R value exceeds 0.9, there is a strong correlation between the two datasets. It is evident that PKNN outperforms BPNN significantly in terms of the R value.

TABLE II
Comparison of prediction accuracy of BPNN and PKNN.

	R	MAPE	MSE	P
BPNN	0.89	8.96%	5.39	0.94
PKNN	**0.97**	**3.45%**	**2.92**	**0.98**

B. Ablation study

Various input parameters: The same sampling method (Gaussian sampling) is selected, along with the same prediction algorithm (PKNN). Following the approach described in [17], based on the data of Fig. 5(c) (defined as the data set $\mathcal{Q}$, 30% of $\mathcal{Q}$ as validation data), an equivalent transformation of the input parameters ($\{c, \phi, \theta\} \rightarrow \{\mathrm{CI}, \theta\}$) is performed. PKNN is applied to both datasets, and 10 sets of cross-validation experiments are conducted. As depicted in Table III, it is evident that considering additional terrain parameters leads to improved prediction accuracy using the same prediction algorithm.

TABLE III
Prediction accuracy for different input parameters.

	R	MAPE	MSE	P
$\{\mathrm{CI}, \theta\}$	0.91	6.40%	4.66	0.92
$\{c, \phi, \theta\}$	**0.97**	**3.44%**	**3.01**	**0.97**

Various sampling methods: The same input parameters ($\{c, \phi, \theta\}$) are selected, along with the same prediction algorithm (PKNN). In order to verify the effectiveness of the proposed Gaussian sampling, two other sampling methods, namely random sampling and LHS, are employed. Each sampling method generates the same amount of data as $\mathcal{Q}$. 10 sets of cross-validation experiments are conducted, and the average prediction accuracy results are presented in Table IV. The results clearly indicate that the experiment using Gaussian sampling achieved significantly higher prediction accuracy.

TABLE IV

PREDICTION ACCURACY FOR DIFFERENT SAMPLING METHODS.

	R	MAPE	MSE	*P*
Random sampling	0.82	10.40%	16.11	0.88
LHS	0.89	6.90%	7.92	0.93
Gaussian sampling	**0.97**	**3.45%**	**2.92**	**0.98**

C. Prediction accuracy comparison of extra sampling data

Based on $\mathcal{Q}$, an additional 20 sets of terrain data are randomly selected as validation data. By comparing the prediction accuracy for different speed classifications, the results are presented in Table V. It can be observed that PKNN exhibits high accuracy for both two and five classifications (no errors), and even for the seven classifications, PKNN outperforms BPNN in terms of prediction accuracy.

TABLE V

VALIDATION ERROR VS ALGORITHM AND CLASS NUMBER.

Algorithms	BPNN			PKNN		
Number of classes	Two	Five	Seven	Two	Five	Seven
Incorrect predictions	0	2	3	0	**0**	**1**
Validation error	0	10%	15%	0	**0**	**5%**

We generate the global mobility map, which required approximately 7 days of simulation time. It is found that the time for prediction using PKNN is less than 2 minutes. In Fig. 7, the mobility prediction map under seven speed classifications is presented. It is worth noting that this process consumes only 12% of the simulation time compared to fully traversing all 5490 terrain rasters, resulting in saving thousands of hours. Furthermore, compared to the LHS-based method, the total time is reduced by 11%, equivalent to a reduction of 20 hours.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a global mobility prediction method for UGVs. We sample high-probability data by fitting multi-dimensional terrain parameters to Gaussian distribution functions, avoiding the need to traverse all terrain rasters in the simulation. Additionally, we introduce the PKNN model, which incorporates semi-variance coefficients from the kriging model into the loss function of the PSO-BPNN, resulting in more accurate predictions. Our method is validated through various experiments, demonstrating significant reductions in simulation time while maintaining high prediction accuracy.

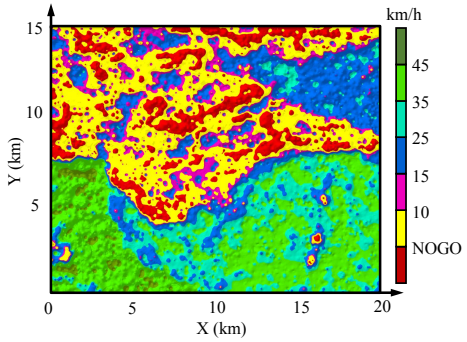

Fig. 7. Global mobility map predicted by PKNN for 7-class.

In the future work, the following aspects will be pursued. More factors for mobility prediction will be considered, such as soil moisture content, bulk density, and vegetation, etc. Data-driven computational mechanics will be introduced to further accelerate the mobility simulation process. Real-world vehicle mobility tests on the off-road terrain will be used to validate the reliability of the mobility prediction method.

REFERENCES

[1] D. Gonzalez, J. Perez, V. Milanes, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1135–1145, Apr. 2016.

[2] G. Hedrick, N. Ohi, and G. Yu, "Terrain-Aware Path Planning and Map Update for Mars Sample Return Mission," *IEEE Robot. Autom. Let.*, vol. 5, no. 4, pp. 5181-5188, Jun. 2020.

[3] J. Y. Wong, P. Jayakumar, E. Toma and J. Preston-Thomasc, "Comparison of simulation models NRMM and NTVPM for assessing military tracked vehicle cross-country performance," *J. Terramech.*, vol. 80, pp. 31–48, Dec. 2018.

[4] E. Karpman, W. Huang, J. Kovecses and M. Teichmann, "Speed-made-good: Mobility map generation for wheeled vehicles on soft terrain," *J. Terramech.*, vol. 101, pp. 11–22, Jun. 2022.

[5] J. Y. Wong, P. Jayakumar, E. Tomac and J. Preston-Thomasc, "A review of mobility metrics for next generation vehicle mobility models," *J. Terramech.*, vol. 87, pp. 11–20, Feb. 2020.

[6] C. H. Ma, J. Yang, G. Zenz, E. J. Staudacher and L. Cheng, "Calibration of the microparameters of the discrete element method using a relevance vector machine and its application to rockfill materials," *Adv. Eng. Softw.*, vol. 147, pp. 833–846, Sep. 2020.

[7] H. Zeng, W. Xu, M. Zang and P. Yang, "Calibration of DEM-FEM model parameters for traction performance analysis of an off-road tire on gravel terrain," *Powder. Technol.*, vol. 362, pp. 350–361, Feb. 2020.

[8] A. Recuero, R. Serban, B. Peterson, H. Sugiyama, P. Jayakumar and D. Negrut, "A high-fidelity approach for vehicle mobility simulation: Nonlinear finite element tires operating on granular material," *J. Terramech.*, vol. 72, pp. 39–54, Aug. 2017.

[9] M. McCullough, P. Jayakumar, J. Dasch, and D. Gorsich, "The next generation NATO reference mobility model development," *J. Terramech.*, vol. 73, pp. 49?60, Oct. 2017.

[10] G. R. Marple, D. Gorsich, P. Jayakumar and S. Veerapaneni, "An Active Learning Framework for Constructing High-Fidelity Mobility Maps," *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 9803?9813, Oct. 2021.

[11] T. M. Wasfy, P. Jayakumar, D. Mechergui and S. Sanikommu, "Prediction of vehicle mobility on large-scale soft-soil terrain maps using physics-based simulation," *Int. J. Vehicle Performance*, vol. 4, no. 4, pp. 347-381, Aug. 2018.

[12] G. Chen, H. Yamashita, Y. Ruan, P. Jayakumar, J. Knap, K. Leiter, X. Yang and H. Sugiyama, "Enhancing Hierarchical Multiscale Off-Road Mobility Model by Neural Network Surrogate Model," *J. Comput. Nonlinear Dyn.*, vol. 16, no. 8, Jun. 2021, Art. no. 081105.

[13] M. Quann, L. Ojeda, W. Smith, D.Rizzo, M. Castanier and K. Barton, "Off-road ground robot path energy cost prediction through probabilistic spatial mapping," *J. Field Robot.*, vol. 37, no. 3, pp. 421-439, Dec. 2020.

[14] D. Mechergui and P. Jayakumar, "Efficient generation of accurate mobility maps using machine learning algorithms," *J. Terramech.*, vol. 88, pp. 53?63, 2020.

[15] W. Huang, H. Liang, L. Lin, Z. Wang, S. Wang, B. Yu, and R. Niu, "A Fast Point Cloud Ground Segmentation Approach Based on Coarse-To-Fine Markov Random Field," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7841?7854, Apr. 2021.

[16] R. Gonzalez, P. Jayakumar and K. Iagnemma, "An efficient method for increasing the accuracy of mobility maps for ground vehicles," *J. Terramech.*, vol. 68, pp. 23-35, Sep. 2016.

[17] C. Hua, R. X. Niu, B. Yu, X. K. Zheng, R. G. Bai and S. Zhang"A Global Path Planning Method for Unmanned Ground Vehicles in Off-Road Environments Based on Mobility Prediction," *Machines*, vol. 10, no. 5, pp. 375-396, May. 2022.

[18] C. Jiang, Z. Hu, Z. P. Mourelatos, D. Gorsich and M. Majcher, "R2-RRT*: Reliability-Based Robust Mission Planning of Off-Road Autonomous Ground Vehicle Under Uncertain Terrain Environment," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 2, pp. 1030-1046, Apr. 2022.

[19] Geospatial Data Clouds. *DEM Database*. Accessed: Oct. 20, 2022. [Online]. Available: http://www.gscloud.cn

[20] SISChina. *Soil Database*. Accessed: Oct. 28, 2022. [Online]. Available: http://www.soilinfo.cn/map/index.aspx

[21] H. Yamashita, P. Jayakumar, M. Alsaleh and H. Sugiyama, "Physics-based deformable tire-soil interaction model for off-road mobility simulation and experimental validation," *J. Comput. Nonlinear Dyn.*, vol. 13, no. 2, p. 021002, Nov. 2018.