Machine-vision-based Online Self-optimizing Control System for Line Marking Machines

Guanxu LONG[1], Lei SHI[2], Gongfeng XIN[1], Shenqi GAO[3]*, Wenliang ZHANG[1], Jicun XU[2]

[1] Shandong Gaosu Group Co., Ltd. Innovation Research Institute,
Longao North Road, Lixia District, Jinan, Shandong, 250101, China
longgx5@163.com, gfxin@163.com, wenliangzhang@163.com

[2] Transportation Infrastructure Construction Engineering Research Center, Shangdong Jiaotong University,
5001 Haitang Road, Jinan, Shandong, 250357, China
shil@sdjtu.edu.cn, xjctz@126.com

[3] School of Information Science and Electrical Engineering, Shangdong Jiaotong University,
5001 Haitang Road, Jinan, Shandong, 250357, China
feize0624@163.com (**Corresponding author*)

Abstract: In order to increase speed and efficiency of making line markings, a vision-navigation-based self-optimizing control system is proposed for an unmanned line marking machine (ULMM). A new Haar-like-feature based algorithm is used to detect a guide line (GL) for the ULMM, and to reduce the influence of complex road surfaces and light. For the problems of an inaccurate ULMM model and local navigation information, an online self-optimizing control algorithm is presented, and its self-learning rules are given. Results of simulations and real machine experiments reveal that the proposed navigation algorithm accurately detects the GL, and the precision of the control system satisfies the requirements of the line marking work.

Keywords: Unmanned Line Marking Machine, Vision navigation, Self-optimizing control, Model-free.

1. Introduction

With the development of unmanned driving technology and automation technology, engineering machines modified by the related technologies are widely applied in engineering. For a new built road, a long line marking task is needed. In the task, labor intensity is extremely high. In order to increase the efficiency of making a line and reduce labor intensity, an unmanned driving technology is applied to line marking machines. Before a line marking task, a white line, named GL, is draw on the road by workers, along which a line is marked by using a hand-push line marking machine. In this paper, a machine vision algorithm is used to detect the GL and obtain a relative position between the GL and the ULMM. Then a self-optimizing controller is designed to control the ULMM, which is shown in Figure 1.

Some advanced vision-based navigation algorithms are so complex that they hardly meet real-time requirements. Many algorithms only consider a value of an individual pixel. However, it is necessary for making full adventure of foreground and background information of the GL.

The mechanical structure of the ULMM is highly complex, and its accurate model is difficult to build. Thus, it is difficult to design a controller by using model-based methods. Moreover, it is challenging to obtain accurate global positions

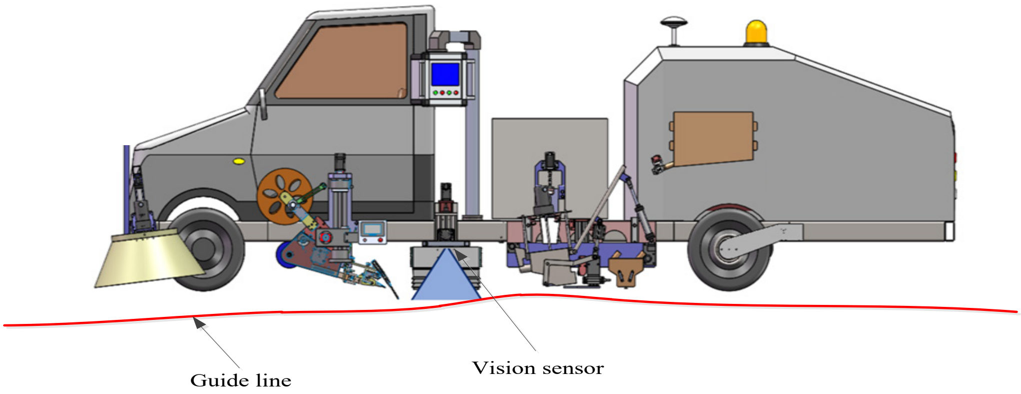


Figure 1. The ULMM

https://doi.org/10.24846/v32i2y202309

of the ULMM. For the above problems, a vision-navigation based control algorithm is proposed as follows:

1. A GL detection algorithm is presented based on Haar-like features. In the method, an improved Haar-like template is built to describe a GL straight-line characteristic. Then, based on a sigmoid function, a nonlinear map is given to stretch the characteristics, then the SVM is used to recognize the extracted Haar-like features;

2. An online self-optimizing non-linear controller is put forward. The controller consists of a predictive controller, a nonlinear PD controller, and online learning rules. The learning rules are used to optimize parameters of the nonlinear PD controller.

The remainder of this paper is structured as follows. The related works about vision navigation and control of vehicle are presented in section 2. Section 3 proposes the guide line detection algorithm, and section 4 proposes GL tracking control of the ULMM. The experiments are designed in section 5 and the conclusion is provided in section 6.

2. Related Works

Accurate navigation is an important factor which affects the straightness of a line marking. Based on high-precision navigation information, the ULMM meets the requirements of a line marking work. In recent years, as significant development of an on-board embedded controller, a navigation technology based on machine vision, has been widely used (Gruyer, Belaroussi & Revilloud, 2016; Ilci & Toth, 2020). In a road construction site, there is obvious distinction between the road surface and the GL. And other obvious navigation methods are scarce. Thus, a machine-vision based navigation strategy is used to detect the GL to provide accurate navigation information for the ULMM. However, vision-based navigation algorithms are easily affected by lighting, environment and other factors (Zhu, Meurer & Günther, 2022). Therefore, in order to improve accuracy of GL detection, many researchers have carried out related works. Jingmei et al. (2020) aimed at illumination influence in vehicle vision navigation, and optimized SURF feature points by using an image gradient. Thus, the matching accuracy was improved. Luo et al. (2019) applied Haar features and wavelet transforms in a navigation algorithm of unmanned aerial vehicles. Li, Wang & Liu (2019) proposed a method of a 3D autonomous navigation line extraction based on a binocular-vision system. It reinforced shadow identification and information fusion, and the road area was better distinguished from its surroundings. Jia, Chen & Zhang (2017) studied a detection method of a road with shadows. It was fused with an original image to obtain more details in dark and bright regions. It was concluded that environment effect elimination and target feature extraction were important research topics.

Another factor affecting line marking work is accuracy of the ULMM control system. Control algorithms of unmanned vehicles mainly include model-based control methods and model-free control methods, and many model-based control methods are used for unmanned driving, e.g., linear quadratic regulator (Park & Kang, 2021), model predictive control (Elsisi & Ebrahim, 2021), sliding mode control (Dai et al., 2021), robust control (Hang et al., 2021) and observer-based methods (Rodríguez et al., 2021), etc. However, an accurate vehicle model is needed if the model-based control algorithm is used for control of the ULMM. But the ULMM is a non-standard engineering machine, and its mechanical structure is quite different from common vehicles. It is difficult to build an accurate ULMM model. Therefore, model-free control methods, e.g., PID (Chu et al., 2022; Liu et al., 2022) and fuzzy control (Silva et al., 2022), were often used for special engineering vehicles. The advantages and disadvantages of these algorithms were very obvious. The PID method was simple, and it was widely used in various control systems (Carlucho et al., 2020; Onat, 2019). However, control performance was impacted by its parameters (Wang, Tan & Cui, 2021). The Fuzzy control method was a model-free and nonlinear control method, and it was used for the control of nonlinear, time-varying and time-delay systems (Liu et al., 2019). However, it was difficult to design fuzzy inference rules.

Considering tuning controller parameters, offline-optimization algorithms based on intelligent algorithms and online self-tuning methods were widely attended. Gao et al. (2021) proposed a parallel optimization algorithm based on the PSO and the GA to tune parameters of a PID controller. Although it successfully obtained a local optimal value and better parameters, its real-time performance needs to be improved. Chiou et al. (2012) combined the PSO with Q-learning to tune parameters and fuzzy set partition of a fuzzy PID controller. This method reduced the number of algorithm iterations, and avoided local optima. Naranjo, Serradilla & Nashashibi (2020) combined stable time, steady state errors and numbers of signal sign change to improve performance of intelligent optimization algorithms. Tran, Chiou & Dang (2020) used the PSO and a method of evolutionary programming to tune parameters of a fuzzy PID controller. The response speed of the control system was improved, and its overshoot was limited. Although intelligent optimization algorithms have been used to tune controller parameters and better control performance has been achieved, they often take a lot of time to optimize the parameters. It is difficult to apply them to real-time adjustment of controller parameters. Moreover, an accurate vehicle model will be needed if parameters of a controller are tuned. And if a vehicle status changes, the pre-set parameters are not suitable for the current vehicle control system.

Therefore, some researchers focused on studying online self-learning of controller parameters. Verma & Padhy (2019) considered the factors which affected the robustness of a control system, e.g., modeling errors and parameter uncertainty, and designed a new univariate tuning method to improve the performance of a controller. An online-tuning method was introduced for PID gains (Kimura et al., 2018), and it was used for speed control of a humanoid robot. For issues of inaccurate system models and poor PID control performance, Yu et al. (2021) proposed a method of adaptive model-free soft actor-critic PID control based on reinforcement learning. It was applied in automatic control of mobile robots, and the robustness of the control system was improved. Furthermore, deep reinforcement learning was used to tune PID parameters (Lawrence et al., 2022). The idea was improved by using a fuzzy logic method (Sarabakha & Kayacan, 2019) and the tracking performance of a nonlinear system was improved. The online self-optimizing controller has better adaptability than that with offline optimization parameters. It learns changes of a control system, and automatically adjusts controller parameters to achieve better control performance in different working conditions.

3. The Guide Line Detection Algorithm

This section depicts the GL detection algorithm based on Haar-like features. Its procedure is shown in Figure 2.

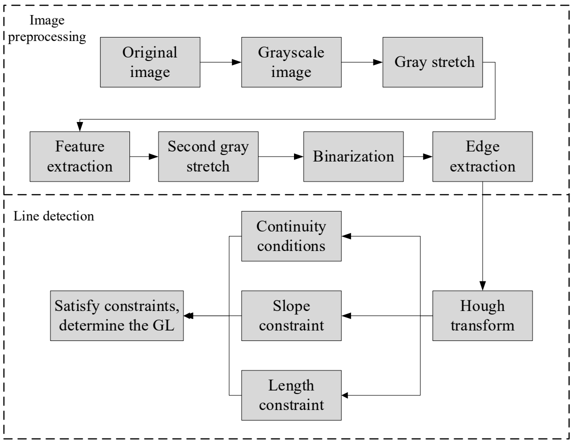


Figure 2. Detection procedure of the GL

Detection steps are introduced as follows. Based on the collected GL gray images, gray values of the image $I(m,n)$ are stretched by using equation (1), i.e.:

$$I(m,n)=255\cdot(\frac{I(m,n)}{I_{\max}})^{\lambda}, \tag{1}$$

where $\lambda>1$, $I_{\max}=\max_{m,n}\{I(m,n)\}$.

Improved Haar-like feature templates are given and shown in Figure 3. Based on the templates, a Haar-like feature is obtained for each pixel.

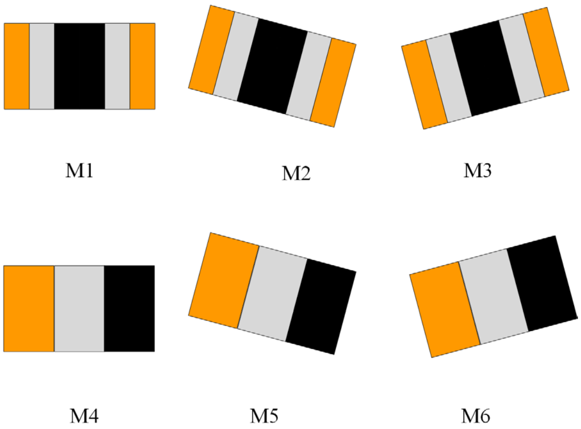


Figure 3. Improved Haar-like feature templates

In the templates $M1\sim M6$, the black part is a GL region, the yellow part stands for a road region, and the grey part represents a transition region. It is assumed that the GL width is $2\cdot w$ where w is a half of the GL width. In $M1\sim M3$, the widths of the yellow and gray regions are w, and black region widths are $2 \cdot w$. In templates $M4 \sim M6$, the width of each region is $2 \cdot w$. The height ranges of all templates are $4 \cdot w \sim 6 \cdot w$. Considering the rotation templates $M2, M3, M5, M6$, their rotation angles are 15 degrees. The six templates are used to highlight straight-line characteristics of the GL on the road, in order to improve the recognition accuracy of the GL.

Considering the i^{th} Haar-like template, a Haar-like feature of a pixel is obtained by using equation (2):

$$h_i = s_i^b - s_i^y, i = 1, \cdots, 6, \tag{2}$$

where s_i^b, s_i^y are pixel sums in black and yellow regions, respectively. Then, each feature is normalized by using equation (3):

$$h_i^n = h_i / \sigma, i = 1, \cdots, 6, \tag{3}$$

where h_i^n is a normalized Haar-like feature; $\sigma = \sqrt{sqmn - mn^2}$; mn is a mean of pixel values in the stretched image, and $sqmn$ is a mean of pixel value squares in the stretched image.

All features greater than 0 are stretched by using equation (4):

$$\overline{h}_i^n = \frac{h_{i,\max}^n}{1 + e^{-\lambda(h_i^n - h_c^i)}}, i = 1, \cdots, 6 \tag{4}$$

,

where $h_c^i = \alpha \cdot h_m^i + (1 - \alpha) \cdot h_t^i$. Suppose that $\Omega_1 = \{h_i^n \mid h_i^n > 0\}$, $\Omega_2 = \{h_i^n \mid h_t^i > h_i^n > 0\}$. h_m^i is a mean of features in Ω_1, and h_t^i is the feature which satisfies that the sum of all features in Ω_2 is 98% of the sum of all features in Ω_1. $h_{i,\max}^n = \max\{h_i^n\}$.

The stretching method is shown in Figure 4.

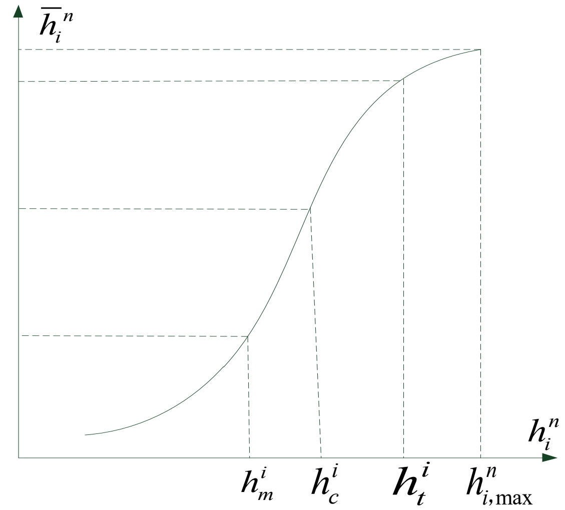


Figure 4. A method of feather stretching

Based on the above 6 normalized Haar-like features, a feature vector, describing straight-line characteristics of each pixel, is constructed and presented by using equation (5):

$$F(m, n, :) = \left[\overline{h}_1^n, \overline{h}_2^n, \cdots, \overline{h}_6^n\right]^T \tag{5}$$

In order to detect the GL, the SVM is used to recognize the constructed feature vector $F(m, n, :)$ and to determine whether a pixel with the feature vector $F(m, n, :)$ is on the GL or not. Thus, a problem of the GL detection becomes a pattern recognition question.

Consider the following nonlinear classifier:

$$f(F) = \omega\phi(F) + b, \tag{6}$$

where F is a feature vector, ω is a weight vector, and

$$\phi(F_i, F_j) = e^{-g\|F_i - F_j\|}.$$

An optimal problem is stated by using equations (7) and (8):

$$\min \ \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \phi(F_i)\phi(F_j) - \sum_{i=1}^{N}\alpha_i \tag{7}$$

$$\begin{aligned} s.t. \ & \sum_{i=1}^{N}\alpha_i y_i = 0 \\ & 0 \le \alpha_i \le C, \ i = 1, 2 \cdots, N \end{aligned}, \tag{8}$$

where α is a Lagrange multiplier, C is a punishment factor.

If a pixel is on the GL, it will be set as 1. Otherwise, it is set as 0. Then, a binary image B is obtained. Subsequently, methods of dilation and erosion are used to reduce noise in the image B, and a new binary image Bn is obtained. From the image Bn, an edge image E is obtained by using equation (9):

$$E(m, n) = Bn(m, n) - Bn(m, n - 1) \tag{9}$$

Then, the Hough transformation is used to detect the GL in the image E, i.e.:

$$x = \rho / \cos\theta - y\tan\theta, \tag{10}$$

where θ, ρ are an angle and an intercept of the GL, respectively.

In general, due to noise, many lines will be obtained from the image E. In this paper, the following method is put forward to eliminate the interference GLs:

Step 1: Select the 3~5 longest lines that meet a length constraint;

Step 2: Since the line is typically continuous, parameters of the real GLs, which are obtained from two adjacent images, satisfy the continuous condition:

$$\Delta^T \begin{bmatrix} \bar{\theta}^{-1} & 0 \\ 0 & \bar{\rho}^{-1} \end{bmatrix} \Delta \le \delta , \tag{11}$$

where $\Delta = \begin{bmatrix} \theta_k - \theta_{k-1} & \rho_k - \rho_{k-1} \end{bmatrix}^T$, $\bar{\theta}, \bar{\rho}$ are radiuses of θ, ρ, respectively. θ_k, ρ_k are GL parameters obtained from the k^{th} image. $\delta > 0$ is a threshold.

Step 3: If there are more than one line satisfying the conditions after steps 1 and 2, the line with the highest mean of pixel values will be selected as a GL. The reason is that the color of the GL is typically white, and the white color means a high mean of pixel values.

4. GL Tracking Control of the ULMM

Considering characteristics of the ULMM, an online self-optimization control system is put forward based on machine-vision navigation. The control system consists of 3 parts: predictive control, nonlinear PD control and online learning rules, as it is shown in Figure 5.

4.1 Tracking and Predictive Errors

It is assumed that a GL is detected, as shown in Figure 6(a).

In Figure 6(b), H, W are the height and the width of a captured image, respectively. $0.7 \cdot H$ is taken into equation (10) to obtain X_c, a current lateral position of the ULMM. A tracking error and its change are described by equation (12):

$$\begin{aligned} e(k) &= (0.5W - X_c) \cdot \kappa \\ ec(k) &= e(k) - e(k-1) \end{aligned}, \tag{12}$$

where $\kappa > 0$, and it represents an actual length between two pixels.

$0.3 \cdot H$ is taken into the (10) to obtain X_p, a predictive lateral position of the ULMM.

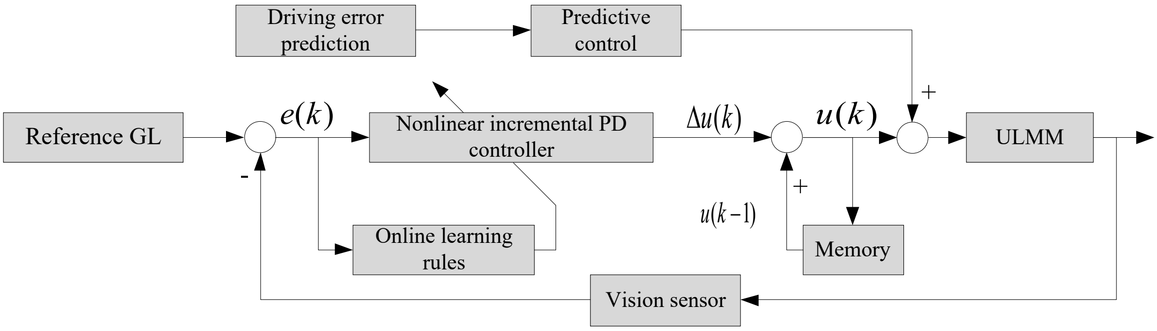
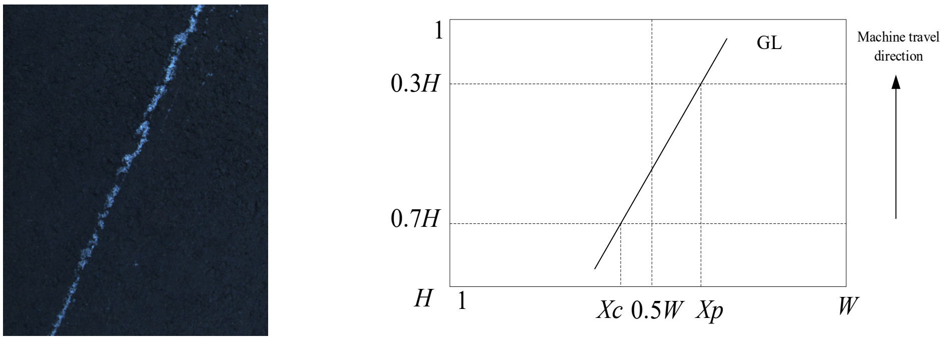


Figure 5. A control system of the ULMM



(a) Guide line

(b) Error calculation

Figure 6. Current and predictive errors

A predictive tracking error and its change are described by using equation (13):

$$\begin{aligned} e_p(k) &= (0.5W - X_p)\cdot\kappa \\ ec_p(k) &= e_p(k) - e_p(k-1) \end{aligned} \qquad (13)$$

4.2 The Online Self-optimizing Control Algorithm

From Figure 6, it becomes obvious that the current lateral position of the ULMM is on the left side of the image, and its predictive lateral position is on the right side. Therefore, the controller of the ULMM is not only related to the current position, but also to the predictive position.

Based on the predictive error, a predictive PD controller is designed and described by using equation:

$$u_p(k) = KP_p \cdot e_p(k) + KP_d \cdot ec_p(k) \qquad (14)$$

Generally, the e and ec of the UAMM satisfy a Gaussian distribution. So, a non-uniform division method based on the Gaussian function is used to divide e and ec, i.e.:

$$\begin{cases} \overline{x}_i = (\dfrac{1}{1-e^{-\tau}} - \dfrac{e^{-\tau\left(\frac{x_i}{X_{\max}}\right)^2}}{1-e^{-\tau}})\cdot X_{\max} & x_i \geq 0 \\ \overline{x}_i = -(\dfrac{1}{1-e^{-\tau}} - \dfrac{e^{-\tau\left(\frac{x_i}{X_{\max}}\right)^2}}{1-e^{-\tau}})\cdot X_{\max} & x_i < 0 \end{cases}, \qquad (15)$$

where $X_{\max}$ is $EC_{\max}$ or $E_{\max}$, $x_i \in \{x_1, x_2, \cdots, x_n\}$ is a uniform division point of e or ec, $\overline{x}_i$ is a non-uniform division point obtained by using equation (15). The method of non-uniform division is shown in Figure 7.

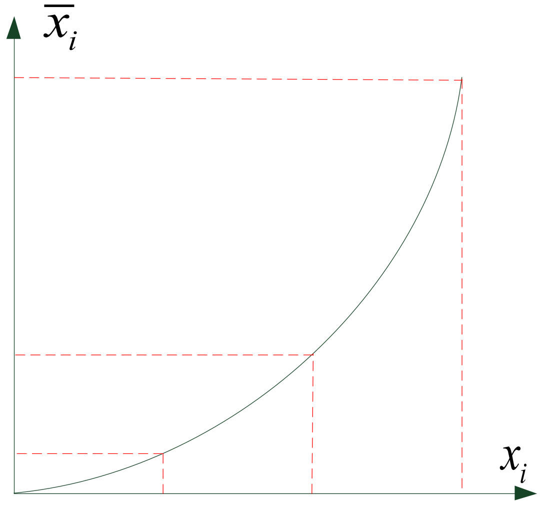


Figure 7. A non-uniform division method

A $e-ec$ plane is constructed and partitioned based on non-uniform division of e and ec, as shown in Figure 8. A set of divided regions is denoted by $\{R_{ij}\}$.

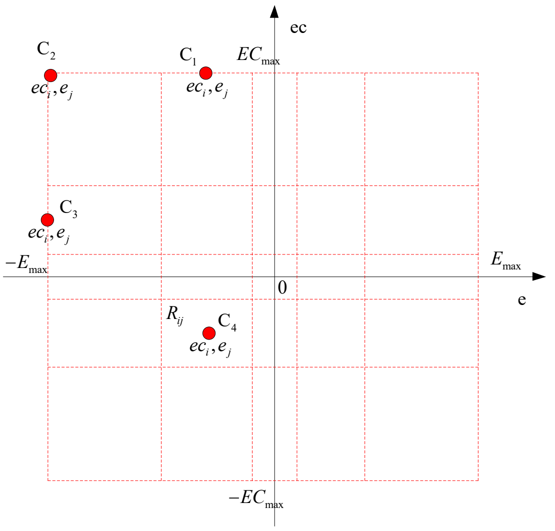


Figure 8. The $e-ec$ plane and its division

Then, an increment PD controller is designed in the divided region R_{ij}, i.e.:

$$\begin{aligned} \Delta u_{ij}(k) = K_{ij}^{p} ec(k) + \\ K_{ij}^{d}(ec(k) - ec(k-1)) \end{aligned}, \qquad (16)$$

where $\Delta u_{ij}(k)$ is an increment of the PD controller in the region R_{ij} at time k.

Based on the increments of all the regions, the increments of a non-linear PD controller are obtained by using equation (17):

$$\Delta u(k) = \sum_{i,j} \omega_{ij} \cdot \Delta u_{ij}(k), \qquad (17)$$

where $\omega_{ij} = \dfrac{w_{ij}}{\sum\limits_{i,j} w_{ij}}$, and

$$w_{ij} = \exp(-\begin{bmatrix} e(k) - e_j \\ ec(k) - ec_i \end{bmatrix}^T \begin{bmatrix} r_{ej}^2 & 0 \\ 0 & r_{eci}^2 \end{bmatrix}^{-1} \begin{bmatrix} e(k) - e_j \\ ec(k) - ec_i \end{bmatrix})$$

is a weight factor of $\Delta u_{ij}(k)$. The point (e_j, ec_i) is a weight center in the region R_{ij}. Considering the weight center (e_j, ec_i), there are four cases denoted by $C_1 \sim C_4$, as shown in Figure 8. r_{ej}, r_{eci} are the weight radiuses of e and ec in the region R_{ij}, respectively. Their values are shown in Table 1.

Table 1. Weight radius values

Weight Radius	C_1	C_2	C_3	C_4
r_{ej}	$0.5r_e$	r_e	r_e	$0.5r_e$
r_{eci}	r_{ec}	r_{ec}	$0.5r_{ec}$	$0.5r_{ec}$

Moreover, if $e(k) > E_{\max}$ (or $ec(k) > EC_{\max}$), let $e(k) = E_{\max}$ (or $ec(k) = EC_{\max}$). And if $e(k) < E_{\min}$ (or $ec(k) < EC_{\min}$), let $e(k) = E_{\min}$ (or $ec(k) = EC_{\min}$).

Based on equation (17), an incremental non-linear PD controller is designed, i.e.:

$$u(k) = u(k-1) + \xi(k)\Delta u(k)\,, \tag{18}$$

where

$$\xi(k) > 0\,,\ \xi(k) = \psi - \psi_0 e^{-(\upsilon \cdot ec^2(k) + \nu \cdot e^2(k))}\,.$$

ψ and ψ_0 are the coefficients used to adjust the learning rates, and υ and ν are the weight coefficients.

The parameters of the controller (18) are learned by using a supervised Hebb learning rule, i.e.:

$$\begin{aligned} K_{ij}^{p}(k) &= K_{ij}^{p}(k-1) + \eta\omega_{ij} e(k)u(k)ec(k) \\ K_{ij}^{d}(k) &= K_{ij}^{d}(k-1) + \\ &\quad \eta\omega_{ij} e(k)u(k)(ec(k) - ec(k-1)) \end{aligned}\,, \tag{19}$$

where η is a learning rate.

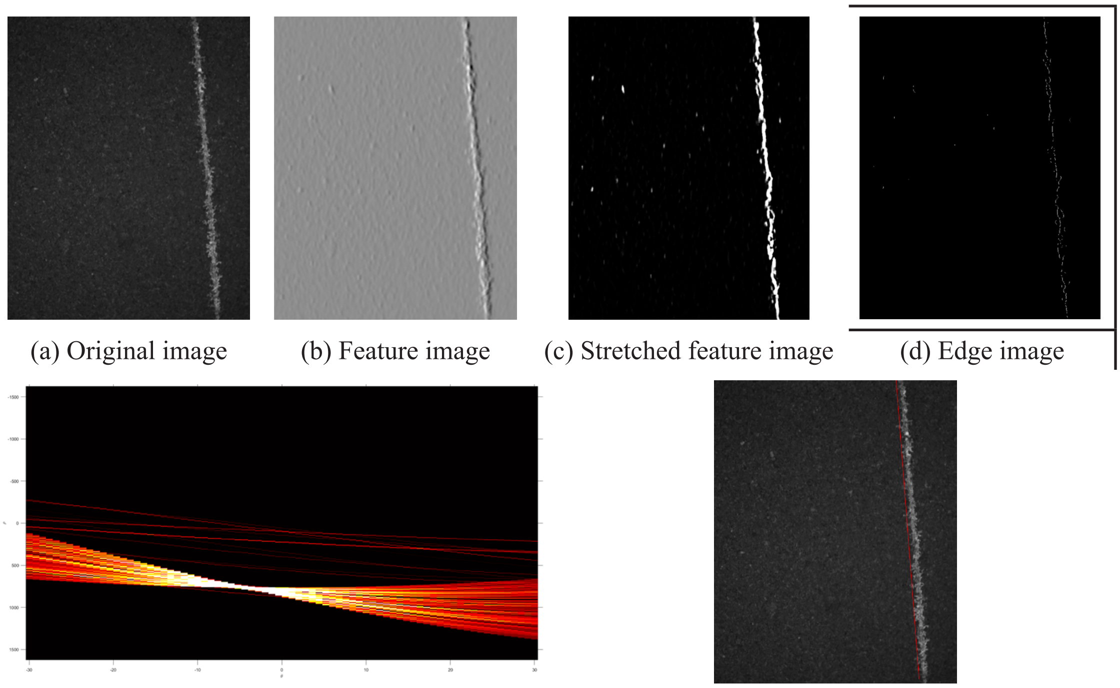
5. Experiments and Verification

5.1 Experiments of GL Detection

In order to verify the feasibility of the GL detection algorithm, different GL images are used for experiments in this section.

An original image of the GL is shown in Figure 9(a). From the original image, Haar-like features are extracted and normalized, as shown in Figure 9(b). It is concluded that straight-line characteristics of the GL are highlighted, and background noise is reduced. Then, the features are stretched by using equation (4), as shown in Figure 9(c). The difference between the GL and the road is significantly highlighted. By using equation (9), an edge image is obtained, as shown in Figure 9(d). The method of Hough transform is used to detect the GL, and a Hough space is obtained, as shown in Figures 9(e)-9(f). It can be seen that the GL is accurately detected.

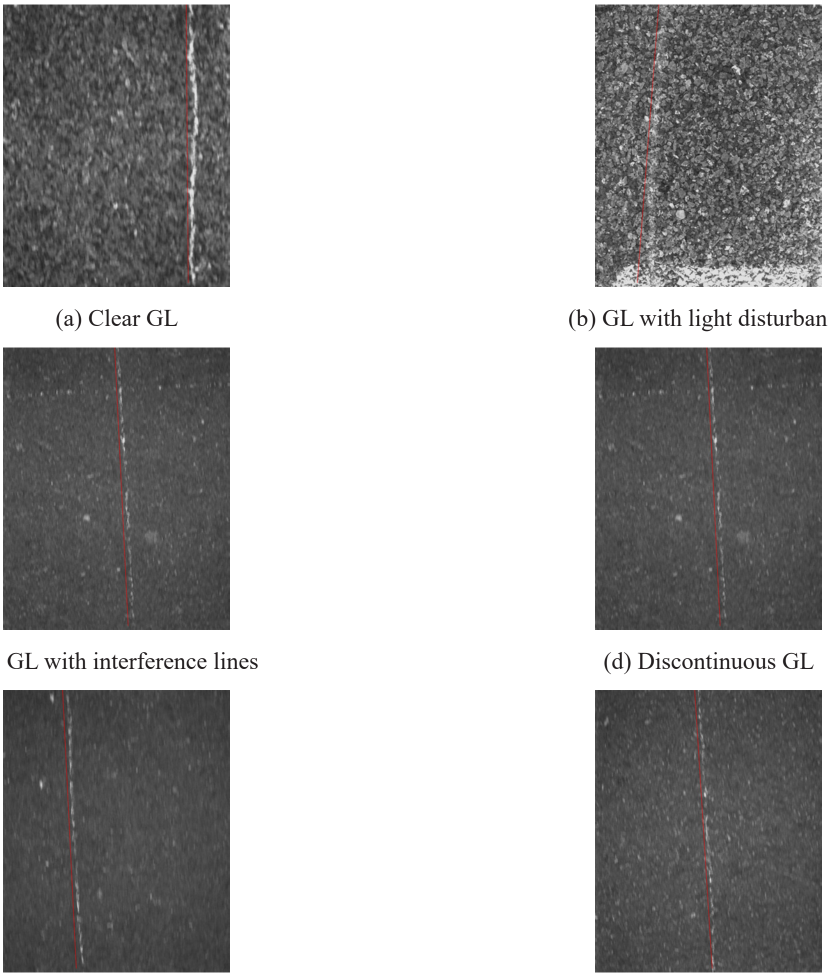
In order to verify the detection algorithm proposed in this paper, detection experiments are carried out under different conditions. Experimental results are shown in Figure 10. The clear GL shown in Figure 10(a) is accurately detected. As shown in Figure 10(b), the GL is blurred because of light



(a) Original image (b) Feature image (c) Stretched feature image (d) Edge image

(e) Hough space (f) GL detection result

Figure 9. Detection results of the GLs



(a) Clear GL

(b) GL with light disturbance

(c) GL with interference lines

(d) Discontinuous GL

(e) Blurred GL

(f) GL with noise

Figure 10. Experimental results

disturbance. The GL is still obtained successfully. Considering GLs shown in Figure 10(c)~(f), the proposed algorithm works well.

This system requires high real-time performance of GL detection. For the problem, comparative experiments are carried out on an industrial computer. And configure of the industrial computer is as follows, i.e. I9-10900 CPU, RTX3060 GPU and 64G RAM. Experimental results are shown in Table 2.

Table 2. Comparison of detection time

Algorithm	Detection time
The proposed algorithm	22ms
Yolov4	71ms

The YOLOv4 (Bochkovskiy, Wang & Liao, 2020), a deep learning method, is well-known because of its fast speed. It takes 71 ms to process an image (see Table 2). The algorithm proposed in this paper needs only 22 ms to complete the GL detection.

5.2 Simulation of Path Tracking

In this subsection, a path tracking experiment is made by using a vehicle model of two degree-of-freedom, in order to verify the effectiveness of the proposed control algorithm. The model is shown in Figure 11.

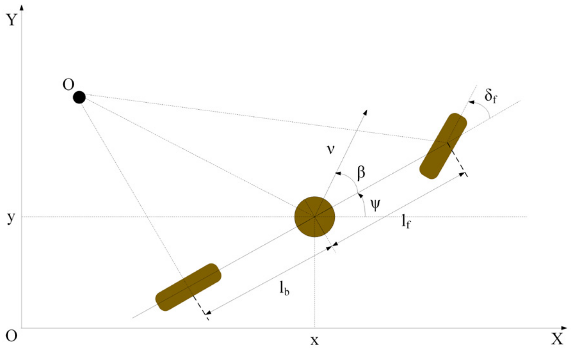


Figure 11. Vehicle model

The model is described by using equation (20), i.e.:

$$\begin{aligned} x_{k+1} &= x_k + v_k\cos(\psi_k+\beta)\cdot T \\ y_{k+1} &= y_k + v_k\sin(\psi_k+\beta)\cdot T \\ \psi_{k+1} &= \psi_k + (v_k/l_b)\cdot\sin(\beta)\cdot T \\ v_{k+1} &= v_k + a\cdot T \end{aligned}, \qquad (20)$$

where (x, y) is the position of the vehicle; ψ is its yaw angle; v is its pre-set speed; δ_f is an angle between its front wheels and its body, and it is also the output of the controller; a is the vehicle acceleration; l_f and l_b are the distances from the vehicle center to the front and rear wheels, respectively, and both of them are set to 2 m. The parameter β is obtained from equation (21):

$$\beta=\tan{}^{1}\{[l_b/(l_f+l_b)]\cdot\tan(\delta_f)\} \qquad (21)$$

The initial position of the vehicle is away from the GL, and the distance between them is 0.5 m. Simulation results are shown in Figure 12. As it can be seen, the proposed control system works well, and its response time is shorter than the one of the MFAC system (Wang et al., 2023). Regarding steady-state errors, the proposed system has no steady-state error, while the MFAC system has a steady-state error of 1 cm. Experiments show that the proposed control system has shorter tuning time and fewer tracking errors.

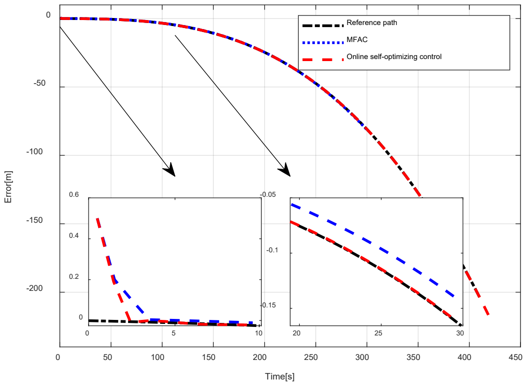


Figure 12. Simulation results of path tracking

5.3 Real Machine Experiments

An experiment is carried out based on an actual ULMM, as shown in Figure 13, where a rear-driven electric vehicle can be noticed. Its speed and steering wheel are controlled by an industrial computer.

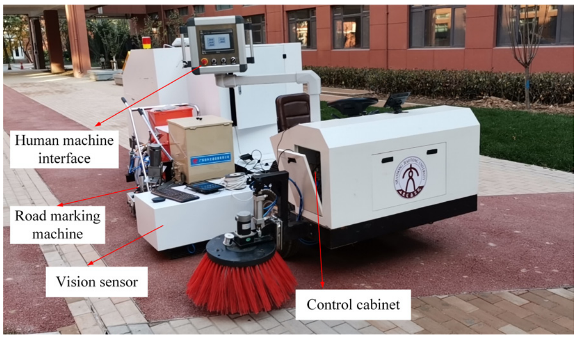
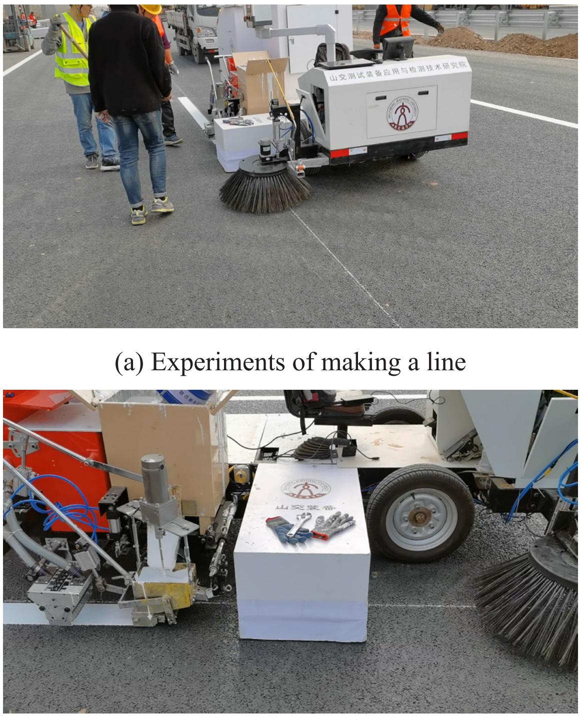


Figure 13. The ULMM

5.3.1 Experiments of Making Lines

This ULMM has been applied to make a line on an expressway of 10 kilometers. Its tracking error data have been collected, and the line making images are shown in Figure 14.



(a) Experiments of making a line

(b) Details of making a line

Figure 14. Line marking experiments

The speed of the ULMM has been set to 20 m/min and 40 m/min, and its recorded tracking errors can be seen in Figures 15 and 16.

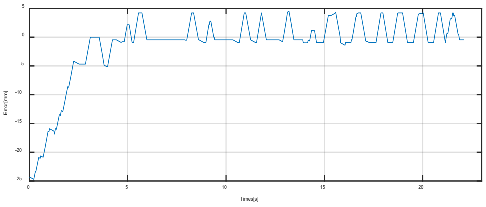


Figure 15. Tracking errors at the speed of 20 m/min

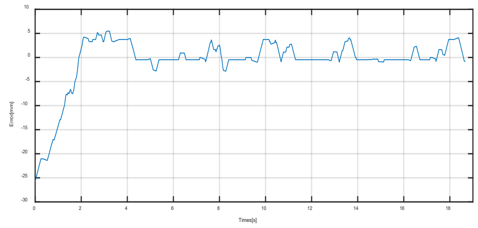


Figure 16. Tracking errors at the speed of 40 m/min

It becomes obvious that the maximum tracking error of ULMM is 0.5 mm, at the speed of 20 m/min, and, when its speed reaches 40 m/min, its tracking error is still within the range of 0-5 mm. Therefore, the performance of the proposed controller satisfies the demands of line marking work.

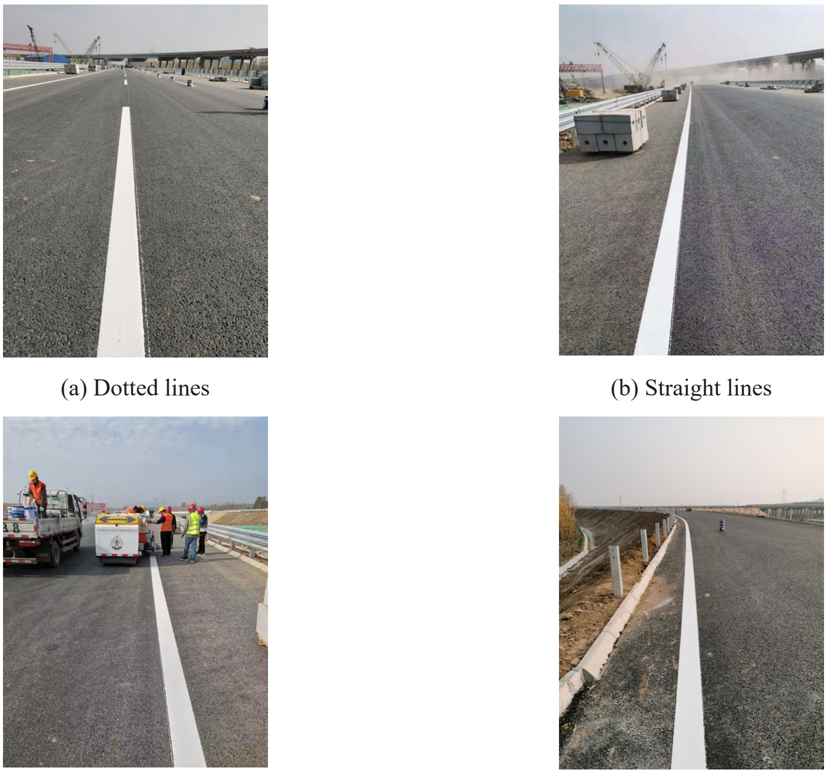
5.3.2 Experimental Results

The experimental results of making a line are shown in Figure 17. From Figures 17(a), (b) and (c), it can be noticed that the line markings made by ULMM are straight and neat. Even when the experiment is carried out on a ramp, as shown in Figure 17(d), the line marking is smooth and consistent with the arc-shaped ramp. So, the experiment results demonstrate that the proposed control system meets the demands of the line marking work.

6. Conclusion

In this paper, a vision-navigation-based self-optimizing control system is proposed for the ULMM. In order to increase detection accuracy of the GL, improved Haar-like feature templates are designed and used to extract characteristics of the GL. Then, the Haar-like features are stretched by using a non-linear mapping. From the detected GL, tracking and predictive errors are obtained. Subsequently, an online self-optimizing control algorithm is proposed, and it consists of a predictive controller and an online-learning non-linear PD controller. The parameters of the non-linear PD controller are learned by using a supervised Hebb learning rule.

Simulation experiments and real machine experiments demonstrate that a high-quality line can be obtained by using the ULMM. This research will effectively promote development of line marking machines.



(a) Dotted lines

(b) Straight lines

(c) Straight lines

(d) Curves

Figure 17. Construction of the ULMM

Acknowledgements

This research reported in this paper has been funded by the Shandong Transportation Science and Technology Program (Grant No. 2021B83) and Key Science and Technology Projects of Transportation from Ministry of Transport of the PRC (Grant No. 2021-MS6-146).

REFERENCES

Bochkovskiy, A., Wang, C. Y. & Liao, H. Y. M. (2020) Yolov4: Optimal speed and accuracy of object detection. *arXiv*. https://arxiv.org/abs/2004.10934. [Accessed 23rd April 2023].

Carlucho, I., De Paula, M. & Acosta, G. G. (2020) An adaptive deep reinforcement learning approach for MIMO PID control of mobile robots. *ISA Transactions*. 102, 280-294. doi: 10.1016/j.isatra.2020.02.017.

Chiou, J. S., Tsai, S. H. & Liu, M. T. (2012) A PSO-based adaptive fuzzy PID-controllers. *Simulation Modelling Practice and Theory*. 26, 49-59. doi: 10.1016/j.simpat.2012.04.001.

Chu, D., Li, H., Zhao, C. & Zhou, T. (2022) Trajectory Tracking of Autonomous Vehicle Based on Model Predictive Control with PID Feedback. *IEEE Transactions on Intelligent Transportation Systems*. 24(2), 2239-2250. doi: 10.1109/TITS.2022.3150365.

Dai, Y., Ni, S., Xu, D., Zhang, L. & Yan, X. G. (2021) Disturbance-observer based prescribed-performance fuzzy sliding mode control for PMSM in electric vehicles. *Engineering Applications of Artificial Intelligence*. 104, 104361. doi: 10.1016/j.engappai.2021.104361.

Elsisi, M. & Ebrahim, M. A. (2021) Optimal design of low computational burden model predictive control based on SSDA towards autonomous vehicle under vision dynamics. *International Journal of Intelligent Systems*. 36(11), 6968-6987. doi: 10.1002/int.22576.

Gao, S., Gao, S., Pan, W. & Wang, M. (2021) Design of Improved PID Controller Based on PSO-GA Hybrid Optimization Algorithm in Vehicle Lateral Control. *Studies in Informatics and Control*. 30(4), 55-65. doi: 10.24846/v30i4y202105.

Gruyer, D., Belaroussi, R. & Revilloud, M. (2016) Accurate lateral positioning from map data and line marking detection. *Expert Systems with Applications*. 43, 1-8. doi: 10.1016/j.eswa.2015.08.015.

Hang, P., Xia, X. & Chen, X. (2021) Handling stability advancement with 4WS and DYC coordinated control: a gain-scheduled robust control approach. *IEEE Transactions on Vehicular Technology*. 70(4), 3164-3174. doi: 10.1109/TVT.2021.3065106.

Ilci, V. & Toth, C. (2020) High definition 3D map creation using GNSS/IMU/LiDAR sensor integration to support autonomous vehicle navigation. *Sensors*. 20(3), 899. doi: 10.3390/s20030899.

Jia, B., Chen, J. & Zhang, K. (2017) Recursive drivable road detection with shadows based on two-camera systems. *Machine Vision and Applications*. 28(5), 509-523. doi: 10.1007/s00138-017-0858-y.

Jingmei, Z., Xin, C., Ruizhi, H. & Xiangmo, Z. (2020) Image Precise Matching with Illumination Robust in Vehicle Visual Navigation. *IEEE Access*. 8, 92503-92513. doi: 10.1109/ACCESS.2020.2994542.

Kimura, K., Nozawa, S., Mizohana, H., Okada, K. & Inaba, M. (2018) Riding and speed governing for parallel two-wheeled scooter based on sequential online learning control by humanoid robot. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 01-05 October 2018, Madrid, Spain.* IEEE. pp. 1-9. doi: 10.1109/IROS.2018.8593685.

Lawrence, N. P., Forbes, M. G., Loewen, P. D., McClement, D. G., Backström, J. U. & Gopaluni, R. B. (2022) Deep reinforcement learning with shallow controllers: An experimental application to PID tuning. *Control Engineering Practice*. 121(5), 105046. doi: 10.1016/j.conengprac.2021.105046.

Li, Y., Wang, X. & Liu, D. (2019) 3D autonomous navigation line extraction for field roads based on binocular vision. *Journal of Sensors*. 8, 1-16. doi: 10.1155/2019/6832109.

Liu, Z., Lu, K., Lai, G., Chen, C. P. & Zhang, Y. (2019) Indirect fuzzy control of nonlinear systems with unknown input and state hysteresis using an alternative adaptive inverse. *IEEE Transactions on Fuzzy Systems*. 29(3), 500-514. doi: 10.1109/TFUZZ.2019.2952783.

Liu, L., Zhang, L., Pan, G. & Zhang, S. (2022) Robust yaw control of autonomous underwater vehicle based on fractional-order PID controller. *Ocean Engineering*. 257, 111493. doi: 10.1016/j.oceaneng.2022.111493.

Luo, D., Shao, J., Xu, Y. & Zhang, J. (2019) Docking navigation method for UAV autonomous aerial refueling. *Science China Information Sciences*. 62(1), 1-16. doi: 10.1007/s11432-018-9578-9.

Naranjo, J. E., Serradilla, F. & Nashashibi, F. (2020) Speed control optimization for autonomous vehicles with metaheuristics. *Electronics*. 9(4), 551. doi: 10.3390/electronics9040551.

Onat, C. (2019) A new design method for PI–PD control of unstable processes with dead time. *ISA Transactions*. 84, 69-81. doi: 10.1016/j.isatra.2018.08.029.

Park, M. & Kang, Y. (2021) Experimental verification of a drift controller for autonomous vehicle tracking: A circular trajectory using LQR method. *International Journal of Control, Automation and Systems*. 19(1), 404-416. doi: 10.1007/s12555-019-0757-2.

Rodríguez, A. J., Sanjurjo, E., Pastorino, R. & Naya, M. Á. (2021) State, parameter and input observers based on multibody models and Kalman filters for vehicle dynamics. *Mechanical Systems and Signal Processing*. 155, 107544. doi: 10.1016/j.ymssp.2020.107544.

Sarabakha, A. & Kayacan, E. (2019) Online deep fuzzy learning for control of nonlinear systems using expert knowledge. *IEEE Transactions on Fuzzy Systems*. 28(7), 1492-1503. doi: 10.1109/TFUZZ.2019.2936787.

Silva, F. L., Silva, L. C., Eckert, J. J. & Lourenço, M. A. (2022) Robust fuzzy stability control optimization by multi-objective for modular vehicle. *Mechanism and Machine Theory*. 167, 104554. doi: 10.1016/j.mechmachtheory.2021.104554.

Tran, H. K., Chiou, J. S. & Dang, V. H. (2020) New Fusion Algorithm-Reinforced Pilot Control for an Agricultural Tricopter UAV. *Mathematics*. 8(9), 1499. doi: 10.3390/math8091499.

Verma, B. & Padhy, P. K. (2019) Robust fine tuning of optimal PID controller with guaranteed robustness. *IEEE Transactions on Industrial Electronics*. 67(6), 4911-4920. doi: 10.1109/TIE.2019.2924603.

Wang, C., Huo, X., Ma, K. & Ji, R. (2023) PID-Like Model Free Adaptive Control with Discrete Extended State Observer and Its Application on an Unmanned Helicopter. *IEEE Transactions on Industrial Informatics*. 1-10. doi: 10.1109/TII.2023.3245223.

Wang, Y., Tan, W. & Cui, W. (2021) Tuning of linear active disturbance rejection controllers for second-order underdamped systems with time delay. *ISA Transactions*. 118, 83-93. doi: 10.1016/j.isatra.2021.02.011.

Yu, X., Fan, Y., Xu, S. & Ou, L. (2021) A self-adaptive SAC-PID control approach based on reinforcement learning for mobile robots. *International Journal of Robust and Nonlinear Control*. 32(18), 9625-9643. doi: 10.1002/rnc.5662

Zhu, C., Meurer, M. & Günther, C. (2022) Integrity of Visual Navigation – Developments, Challenges, and Prospects. *NAVIGATION: Journal of the Institute of Navigation*. 69(2), 518. doi: 10.33012/navi.518.