Autonomous Control Through the Level of Fatigue Applied to the Control of Autonomous Vehicles

Oscar A. Mayorga(✉) and Víctor H. Andaluz(✉)

Universidad de las Fuerzas Armadas ESPE, Sangolquí, Ecuador
{oamayorga,vhandaluzl}@espe.edu.ec

Abstract. In this article we present the detection of fatigue level of a vehicle driver and according to this level the autonomous driving assistance is implemented, the detection of fatigue level is based on facial recognition using deep learning (Deep Learning) developed in the Matlab software, applying neural networks previously trained and designed, this detection sends us a metric that comprises four levels, according to the metric the position and velocity control of a simulated Car-Like vehicle in the Unity3D software is performed which presents a user friendly environment with the use of haptic devices, the development of the control algorithm is based on path correction which calculates the shortest distance to reenter the desired path.

Keywords: Face recognition · Deep learning · Path tracking

1 Introduction

Traffic accidents are unforeseen events that cause harm to a person or thing, suddenly caused by an involuntary external agent [1, 2]. Over the years, the continuous increase in paths, and the number of vehicles in most countries, the velocity, lack of concentration, failure to respect regulatory traffic signals, driving under the influence of alcohol, driving in a state of drowsiness, they have produced an increase in the mortality rate and disability in people [2–6]. Some studies show that from 25% to 30% of traffic accidents are related to fatigue [7], so the way to reduce this type of accident is a major problem that must be resolved to reduce the threat of life, health and property of people.

Some researchers have developed some methods to improve driving safety based on: sensors and parking cameras [8, 9], cognitive information processing [10], threat assessment during semiautonomous driving [11]. If the driver's fatigue state can be detected, it will be possible to prevent drivers from driving in a sleepy state [12]. Some physiological characteristics, such as brain wave, blinking frequency, heart rate and blood pressure are aspects that identify fatigue in drivers [13–15].

The level of fatigue is detected through the recognition of characteristics defined in the human face and the positions it presents. For the detection of faces, the following aspects are considered: shape of the face, alignment of the face and face representation [16]. The main methods of face recognition can be classified into methods based on

S. S. Ge et al. (Eds.): ICSR 2018, LNAI 11357, pp. 115–126, 2018.
https://doi.org/10.1007/978-3-030-05204-1_12

geometric characteristics, sub-spatial analysis methods, model-based methods, local methods based on characteristics, techniques based on machine learning [17], among others. However, since a few years ago, the methods based on deep learning combine massive data training, this manages to recognize faces quickly and effectively [18].

Deep learning is a set of machine learning algorithms that attempts to model high-level abstractions in data using architectures composed of multiple non-linear transformations [19]. Several deep learning architectures, such as deep neural networks, deep convolutional neural networks, and deep belief networks, have been applied to fields such as computer vision, automatic speech recognition, and recognition of audio and music signals, this has generated results state of the art in various tasks [20].

For what has been described, the present work shows an alternative to this problem, considering levels of fatigue in the driver identified by means of deep learning, and with the assistance of a driver decreases the velocity of the vehicle to avoid traffic accidents. The driver's fatigue level detection is based on the gestures and facial positions that the driver makes when he shows a certain degree of drowsiness or fatigue. The present work presents three main stages: *(i) Fatigue detection* is the one that is responsible for determining the driver's level of tiredness that is based on the gestures and positions.

For the detection of fatigue level, a detection algorithm will be implemented through deep learning that classifies the images obtained by means of a webcam processed by the Matlab software, the algorithm contains a neural network trained by means of a database in which they are located images of the levels of fatigue. *(ii) Driving assistance,* in this stage is proposed a control algorithm that uses the variables of velocity, position and orientation that are received from the haptic devices, to perform the correction of these variables with respect to a pattern and according to the level of fatigue presented by the driver; *(iii) Simulation environment,* this contains user-friendly driving simulation software which obtains velocity, position and direction data once a control algorithm has been made according to the levels of fatigue to move the vehicle and feedback the value of the variables that are in that moment towards the control algorithm.

2 Fatigue Level Detection

The recognition of the level of fatigue presented by a driver is based on the detection of the facial features, the levels have been divided into four: (i) level zero, the driver does not present any change of facial features that he usually has; (ii) level one, there are constant yawns, (iii) level two, the eyes of the driver have been partially closed with the upper eyelid in half, (iv) level three, the driver has completely closed the eyes and then makes a tilt mild face that indicates total drowsiness.

To identify the aforementioned levels, object detection and recognition techniques are used; because they are similar techniques which vary when performing their execution, the processing based on Deep Learning takes it with a subset of characteristics where the level of fatigue that is presented at that moment is identified.

The deep learning approach is based on the transfer which involves a previously trained neural network with thousands of categories, the Deep Learning algorithm consist of 4 stages in which the structure of the neural network is included: (i) import of the convolutional neuronal network previously trains, (ii) modification of the layers final, (iii) new training of the network, (iv) prediction and evaluation of the accuracy of the network; between layer three and four there is a retro feed for the improvement of the network.

The first stage uses the Alexnet network [21], which has been trained with a database with more than 15 million tagged images corresponding to 22000 categories, the structure of this network consists of 13 layers of learning and 3 fully-connected layers for classification. The learning layers have three sublayers: (i) convolution, (ii) grouping, (iii) Rectified Rectilinear Unit (ReLU). Each of the layers takes the data from the previous layer, transforms them according to the parameters it needs to learn and finally transfers them to the next layer, *i.e,* each layer learns different properties of the image, activates with the characteristics of the image such as colors, shapes, contours, gray scale among others.

In the second stage corresponds to the reconfiguration of the last three layers of the Alexnet network [21], which must contain the necessary information to combine the characteristics that the network extracts in the classification and label probabilities; the modification of the antepenultimate layer corresponds to the fully connected layer that obtains the new weights of each of the previous layers to perform deep learning, the penultimate layer corresponds to the Softmax layer that contains the probabilities of the new trained objects and the last layer corresponds to the output layer where the classification of the input image is made.

The network training stage uses the new images corresponding to fatigue levels, grouped into four categories: (i) normal, (ii) yawning, (iii) medium and (iv) drowsiness, all of them contained in a database with 100 images in each one of the categories, Fig. 1, that contain the characteristics of the facial features that each person presents in both men and women. The greatest difference in traits can be identified in the category of yawning because the opening of the mouth of people is different, the likelihood that they close their eyes when yawning is high and the formation of other physical features such as expression lines changes from person to person. For network training a stochastic gradient descent is used with a lot size of 5 examples, an impulse of 0.9 and a weight decrease of 0.0005, this small amount of weight reduction is not a simple regulator, but that reduces the training error of the model.

To perform the training, the database contains images of size 227 $\times$ 227 pixels, the parameters that are defined in the training options, the database is divided into 70% of the images for the training and 30% for validation, the initial learning rate, frequency and validation estimation are defined. The final stage corresponds to the evaluation of

(a) Yawn (b) Medium (c) Normal (d) Somnolence

Fig. 1. Level of fatigue categories

the new trained network, where the percentage of operation of the new data is seen, for this the matrix of confusion of dimension is obtained *kxk*, *k* is the number of categories trained, the matrix contains the new trained categories. Each of the positions in the matrix indicates how many images have confusion in the same category as in the other categories; if the matrix diagonal contains the total number of images trained per layer, this indicates that the network had no errors when performing the training and has a 0% error. Figure 2 shows the confusion matrix of the facial features training where the diagonal contains 70% of the total network training, and in each of the positions of the matrix corresponds the number of images that is confused by identifying the four levels of fatigue that a driver represents.

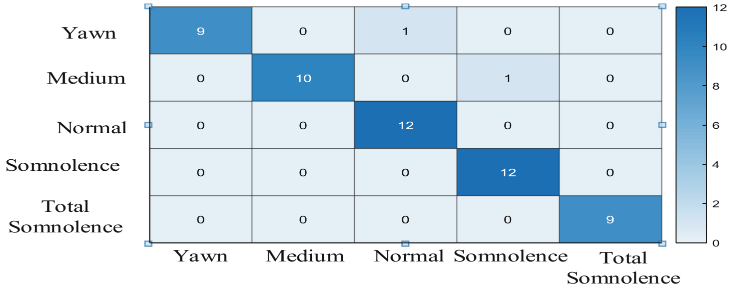


Fig. 2. Matrix of confusion of the training of the network for the detection of fatigue.

In the classification of images, the prediction is made according to the probabilities and the network continues training until the error decreases to 0% in the prediction of each of the images that are entered for the classification. To have a good workout the Accuracy must be greater than 65% and the error must be close to zero.

3 Control System Implemented

The proposed control algorithm is implemented in the Matlab software, which allows the programming of the control algorithms in an efficient way. The structure of the control is in closed loop with a force feedback that indicates the behavior of the terrestrial vehicle within the virtual environment that is developed in the Unity 3D graphics engine, with this behavior the controller sends the corresponding control actions to correct the error. In Fig. 3 we propose the control structure that is carried out depending on the level of fatigue that is detected in the driver, to execute the appropriate control in Matlab for path correction.

The control stage consists of three parts: *(i) Fatigue index*, which is given by the algorithm of Deep Learning according to the level of fatigue, which presents the driver of the vehicle as indicated in (1), levels of fatigue they have discrete values between 0 and 1, thus: normal is 0, yawn is 0.25, medium is 0.5, drowsiness and total drowsiness is 1 because they are levels that occur immediately in the driver.

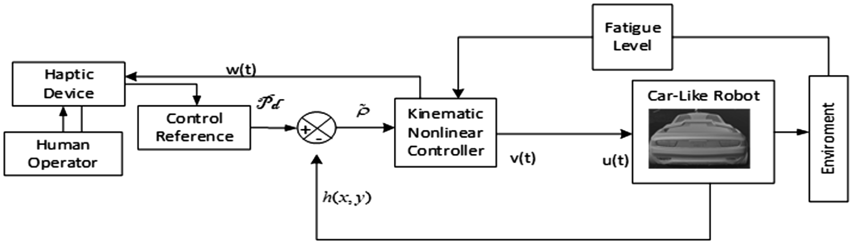


Fig. 3. Control scheme for driver assistant

$$I_s = 1 - L \tag{1}$$

(ii) Input devices, consisting of the Human Operator, the haptic devices, and the desired path that represents the control reference, the force fed back by the Logitech G29 haptic devices has a maximum force of 2.1 [N], this haptic device consists of two servomotors with position encoders that make possible an insertion of haptic experience.

(iii) Kinematic control, the movement of the vehicle is based on the kinematic model of the mobile robot type Car-Like with point of interest in $h(t) = f(x, y)$, which is represented by (2); where x, y are the coordinates of the center of the two rear wheels; θ indicates the direction of the mobile robot type Car-Like with respect to the axis X; $\varnothing$ is the steering angle of the front wheels and is the distance from the center of the robot to the control point of interest as indicated in Fig. 4.

$$\boldsymbol{h}(t) = \boldsymbol{J}(\theta)\boldsymbol{v}(t) \tag{2}$$

with:

$$\begin{bmatrix} \dot{h}_x \\ \dot{h}_y \end{bmatrix} = \begin{bmatrix} \cos\theta & -a \sin\theta \\ \sin\theta & a \cos\theta \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix}$$

$$\omega = \frac{u}{p} \tan\emptyset$$

where, ρ represents the distance between the front and rear wheels; u is the linear velocity of the vehicle; and ω represents the angular velocity.

The kinematic controller calculates the position errors in each sampling period, in order to be used to determine the linear velocity and angular velocity of reference, which will allow an assistance of automatic handling in the vehicle in order to reduce errors and with this avoid traffic accidents In Fig. 4 the denoted path tracking problem is indicated $P(s)$, where $P(s) = (x_p(s), y_p(s))$; P_d represents the current desired point of the Car-Like type robot which is considered as the closest point to $P(s)$ he traced path, this is defined as $P_d = (x_p(s_D), y_p(s_D))$, where s_D is the defined curvilinear abscissa of the point P_d; $\tilde{x} = x_p(s_D) - x$ represents the position error in the address X; and $\tilde{y} = y_p(s_D) - y$ it is the positional error in the address Y.

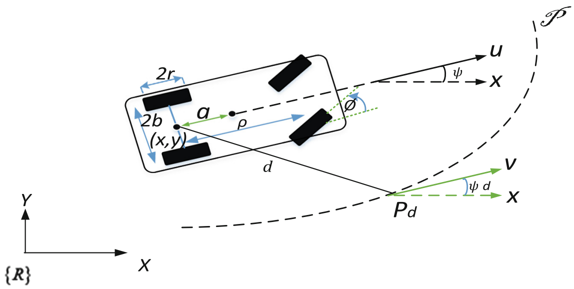


Fig. 4. Problem of the Car-Like for path tracking.

In this regard, the control errors for the human-car operator system are $\rho(t)$ that it represents the distance between the Car-Like vehicle position $h(x, y)$ and the desired point, P_d, where the position error in the direction $\boldsymbol{\rho}$ is $\tilde{\rho} = 0 - \rho$, *i.e.,* the distance between the position of the vehicle $h(x, y)$ and the desired point P_d is zero; is the orientation error of the vehicle that is defined as $\tilde{\psi} = \psi_d - \psi$, where ψ_d is the orientation of the unitary vector tangent to the trajectory of the point P_d with respect to the reference system.

The reference velocity is defined by the control errors and the angular velocity of the vehicle that it has as indicated in (3):

$$v_r = \frac{v_{d\max}}{1 + K_r\|\rho\| + K_f\|\omega\|} \tag{3}$$

where; v_d is the maximum desired velocity, K_r defines the positive constant that weighs the control error; K_f is the positive constant that weighs the change of angular velocity of vehicle type Car-Like.

As described above, the following control law is proposed as autonomous management assistance according to the fatigue levels presented:

$$\begin{bmatrix} u_c \\ \omega_c \end{bmatrix} = \underbrace{\mathbf{J^{-1}}\left(K_s\begin{bmatrix} \dot{x}_p \\ \dot{y}_p \end{bmatrix} + K_c\begin{bmatrix} \tanh(\tilde{x}) \\ \tanh(\tilde{y}) \end{bmatrix}\right)}_{V_1} + \underbrace{I_s\mathbf{v_m}}_{V_2} \tag{4}$$

where, $\dot{x}_p = |v_r|\cos(\psi_d)$ and $\dot{y}_p = |v_r|\sin(\psi_d)$ being u_c y ω_c the output velocities of the kinematic controller; $|v_r|$ represents the module of the input reference velocity for the vehicle controller; $\dot{x}_p$ is the projection of the vector v_r in the direction X, $\dot{y}_p$ is the projection of the vector v_r in the direction Y; $\mathbf{J^{-1}}$ represents the matrix of inverse kinematics of the robot described in (4), in addition $\tilde{x}$ and $\tilde{y}$ are the errors of position in the directions x, y respectively, with respect to the reference system, in order to saturate the vehicle velocities the function has been included $\tanh(.)$.

Remark 1: The control law described in (4) is comprised of two main terms: V_1 it represents the kinematic control that is responsible for correcting the control errors produced when the fatigue index is 1, with K_s which it is a gain with values $[0-1]$, that allows applying the desired velocity, V_2 represents the maneuverability of the driver according to the level of fatigue with linear and angular velocity entered from the haptic devices, *i.e,* when the fatigue index I_s indicated in (1) is greater, the incidence of the signals of the haptic devices on the vehicle is smaller because V_1, it begins to increase.

According to the velocities obtained from the proposed algorithm, this is used to generate feedback from the wheel through the relationship of lateral forces $F_{x,y}$, these forces are generated in the wheels when taking the curve, to generate a torque that is indicated in (5), applied to the steering wheel with which the orientation of the vehicle is corrected:

$$M_v = \frac{M_r}{rd(K_z)} \tag{5}$$

where M_v is the moment in the steering wheel, M_r defines the torque that exists in the steering wheels, rd he ratio of reduction in terms of the angle of rotation of the steering wheel and the angle of rotation of the wheels $\varnothing$ and K_z is the performance of the steering. According to Eq. (5) is defined: $M_r = F_{x,y}(r_w)$ with $F_{x,y} = \frac{m_T(\mathbf{v}_{ref})}{\Upsilon}$, where r_w is the radius of the steering wheels; m_T it is the mass of the vehicle; Υ, represents the radius of curvature of the vehicle.

For the analysis of stability is considered (4), when the fatigue index is zero because the level of fatigue is one thus canceling the velocities of haptic devices and starts by the relationship:

$$\tilde{x} = -\tilde{\rho}\ \sin(\psi_d)\ \text{and}\ \tilde{y} = \tilde{\rho}\ \cos(\psi_d) \tag{6}$$

in addition to:

$$\dot{\rho} = -\tilde{x}\sin(\psi_d) + \tilde{y}\cos(\psi_d) \tag{7}$$

Now it is considered that $\tilde{\rho} = -\rho$, from which the derivative is obtained as a function of the time obtained,

$$\dot{\tilde{\rho}} = -\dot{\rho} \tag{8}$$

If (7) are replaced in (8) it is obtained:

$$\dot{\tilde{\rho}} = \dot{x}\sin(\psi_d) - \dot{y}\cos(\psi_d) \tag{9}$$

Now if it enter $\dot{x}_p$, $\dot{y}_p$ and (4) in (9), result:

$$\dot{\tilde{\rho}} = (\tanh(\tilde{x})\sin(\psi_d) - \tanh(\tilde{y})\cos(\psi_d)) \tag{10}$$

To know the behavior of $\tilde{\rho}$ in the closed loop system of the vehicle, substitute (6) in (9) and obtain:

$$\dot{\tilde{\rho}} = (\tanh(-\tilde{\rho}\sin(\psi_d))\sin(\psi_d) - \tanh(\tilde{\rho}\cos(\psi_d))\cos(\psi_d)) \tag{11}$$

It is observed that from (11) it is concluded that the system has only one equilibrium point, *i.e.* $\tilde{\rho} = 0$. To analyze the stability of the car-like vehicle system, the candidate function to Lyapunov is proposed, $V(\tilde{\rho}) = \frac{1}{2}\tilde{\rho}^2 > 0$. While the derivative of the trajectory versus time is defined as $\dot{V}(\tilde{\rho}) = \tilde{\rho}\dot{\tilde{\rho}}$, a sufficient condition for the equilibrium stability of the closed loop system is that $\dot{V}(\tilde{\rho})$ it be defined as negative. In this regard, the closed loop system of (11) is introduced in $\dot{V}(\tilde{\rho})$, obtaining:

$$\dot{V}(\tilde{\rho}) = \tilde{\rho}\ \tanh(-\tilde{\rho}\sin(\psi_d))\sin(\psi_d) - \tilde{\rho}\ \tanh(\tilde{\rho}\cos(\psi_d))\cos(\psi_d) \tag{12}$$

that is $\dot{V}(\tilde{\rho}) < 0$, the stability of the closed loop system is guaranteed if the controller's profit constants that weigh the control error are: $K_s > 0$ and $K_c > 0$.

In context, from (12) it can be concluded that $\lim_{t\to\infty}\tilde{\rho}(t) \to 0$, *i.e.*, $\tilde{x}(t) \to 0$ and $\tilde{y}(t) \to 0$ with $t \to \infty$ asymptotically. Therefore, from (4) it is concluded that the final velocity of the point of interest when the driver is in a state of somnolence or total somnolence $V = \left|\mathbf{v}_p(s_D, h)\right|\cos\psi_d$, will therefore be $\tilde{\psi}(t) \to 0$ asymptotically.

4 Experimental Results

This section presents the performance of the proposed handling assistance control based on the level of fatigue of the vehicle driver. The experimental tests were implemented in a 3D virtual environment that interacts in real time with the Matlab software. Figure 5 shows the system operation which is implemented in an Intel Core i5 PC, with 8 GB of RAM, N-VIDIA GTX960 graphics card and Logitech G29 handwheel, for the detection of fatigue level by means of Deep Learning, which indicates the probability of each detected level and classifies it according to the percentage found.

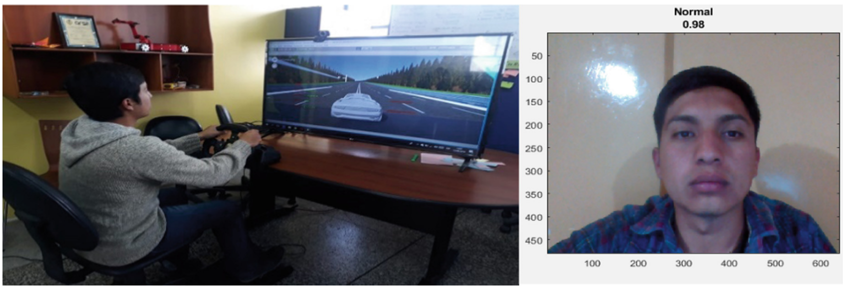
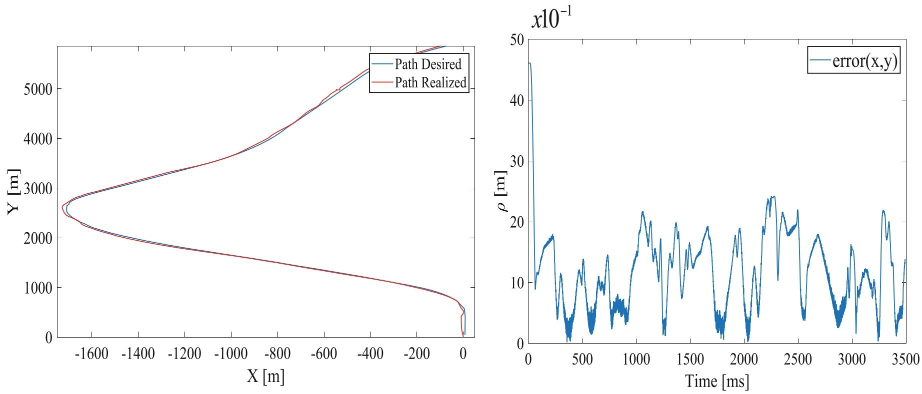


Fig. 5. Interaction between haptic devices and simulation software

To indicate the performance of the proposed controller, two experimental tests were carried out: *(i) Fatigue level detection without assistance,* in this test the Deep Learning algorithm has detected the level of fatigue and the driver for path assistance it is inactive, thus obtaining the desired path and the one made in Fig. 6(a). While the error that causes the level of fatigue is observed in Fig. 6(b), which is not greater than 2.5 [m] in virtue that protective barriers were incorporated in the simulation environment.



(a) Path desired (b) Control Error

Fig. 6. Driving without assistance.

(ii) Detection of the fatigue level of the driver with driving assistance, for this test the algorithm of Deep Learning and the active control algorithm were applied for all levels of fatigue recognized, thus obtaining the desired path and the path performed in Fig. 7, the path error presented by the vehicle is in Fig. 8, while the linear velocity applied by the controller is indicated in Fig. 9, the angle of rotation of the vehicle applied by the controller is indicated in Fig. 10.

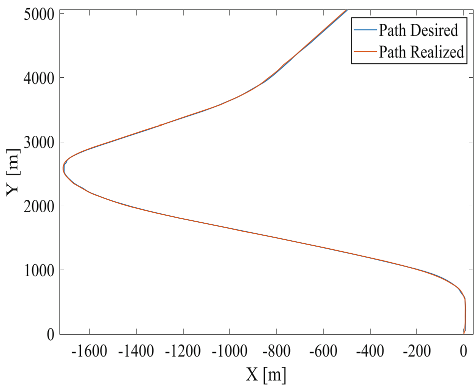


Fig. 7. Desired path and path realized by the proposed controller

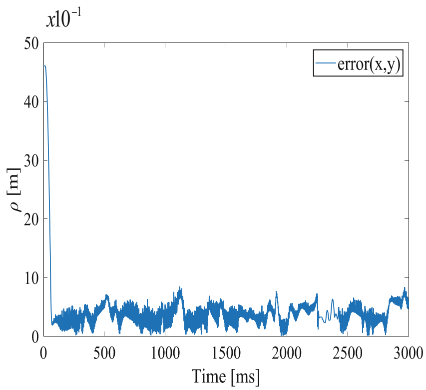


Fig. 8. Path error with active control algorithm

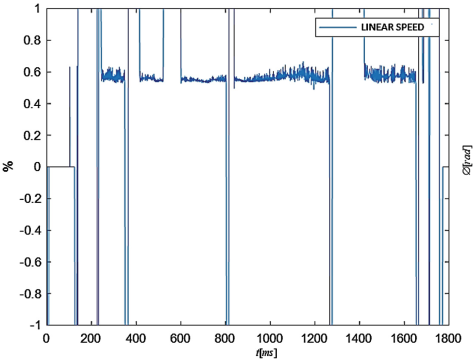


Fig. 9. Linear velocity applied by the controller

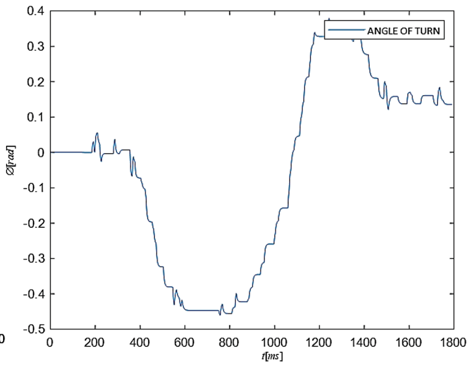


Fig. 10. Vehicle turning angle with active driving assistance.

According to the experimental tests *(i)* and *(ii)* it can be seen that the desired path with the path performed without the proposed controller are separated; while when the control algorithm acts the two paths are superimposed as indicated in Figs. 6(a) and 7 respectively, the maximum error presented when the controller is not current is 2.5 [m], but when the controller is current the error is reduced to a maximum of 1 [m] as seen in Figs. 6(b) and 8, the linear velocity applied when the controller is active is reduced to an average from 0.83 as shown in Fig. 9, according to the fatigue index, and finally the angle of rotation of the vehicle which has a linear variation when the driver is activated. according to the desired path as indicated in Fig. 10.

5 Conclusions

In this article, a fatigue level detection system of a terrestrial vehicle driver using Deep Learning algorithms is proposed to avoid accidents. The simulation environment is developed in Unity 3D Engine where the user can have an immersion in a real path driving, fatigue levels are based on the detection of facial features according to the established characteristics, simulation results shows a total recognition of the different levels according to the characteristics presented by the driver, according to this recognition the controller used for the driving assistance emits signals that correspond to the linear and angular velocity to the simulation software to perform a feedback to haptic devices according to the level of fatigue presented by the driver, thus reducing the distance between the desired path and the current path.

Acknowledgements. The authors would like to thanks to the Corporación Ecuatoriana para el Desarrollo de la Investigación y Academia –CEDIA for the financing given to research, development, and innovation, through the CEPRA projects, especially the project CEPRA-XI-2017-06; *Control Coordinado Multi-operador aplicado a un robot Manipulador Aéreo;* also to

Universidad de las Fuerzas Armadas ESPE, Universidad Técnica de Ambato, Escuela Superior Politécnica de Chimborazo, and Universidad Nacional de Chimborazo, and Grupo de Investigación en Automatización, Robótica y Sistemas Inteligentes, GI-ARSI, for the support to develop this work.

References

1. Wei, W., Hanbo, F.: Traffic accident automatic detection and remote alarm device. In: 2011 International Conference on Electric Information and Control Engineering, Wuhan, pp. 910–913 (2011)
2. Lia, C.: Intelligent traffic accident detection system based on mobile edge computing. In: 2017 3rd IEEE International Conference on Computer and Communications, China, pp. 2110–2115 (2017)
3. World Health Organization: Global status report on road safety 2015. World Health Organization (2015)
4. Agencia Nacional de Transito del Ecuador. Estadísticas de Siniestros. https://ant.gob.ec
5. Agencia Nacional de Transito del Ecuador. Estadísticas de mortalidad. https://ant.gob.ec
6. Traffic Incident Management: Federal Highway Administration. http://ops.fhwa.dot.gov/aboutus/one_pagers/tim.htm. Accessed 15 Oct 2013
7. The Royal Society for the Prevention of Accidents: Driver fatigue and road accidents: a literature review and position paper, Birmingham, U.K. (2001)
8. Eriksson, M., Papanikotopoulos, N.P.: Eye-tracking for detection of driver fatigue. In: 1997 IEEE Conference on Intelligent Transportation System. LTSC 1997. IEEE (1997)
9. Shu-ming, S., et al.: Driver mouth monitoring method based on machine vision. J. Jilin Univ. Technol. (Nat. Sci. Ed.) **2**, 014 (2004)
10. Tomlin, R., Villa, V.: Attention in Cognitive Science and Second Language Acquisition. Stud. Sec. Lang. Acquis. **16**, 183–203 (2008)
11. Shia, V., et al.: Semiautonomus vehicular control using driver modeling. IEEE Trans. Intell. Transp. Syst. **15**(6), 2696–2709 (2014)
12. Tomas-Gabarron, J.B., Lopez, E., García, J.: Vehicular trajectory optimization for cooperative collision avoidance at high speeds. IEEE Trans. Intell. Transp. Syst. **14**(4), 1930–1941 (2013)
13. Lin, C.-T., et al.: EEG-based drowsiness estimation for safety driving using independent component analysis. IEEE Trans. Circ. Syst. I: Regul. Pap. **52**(12), 2726–2738 (2005)
14. Swarnkar, V., Abeyratne, U., Hukins, C.: Objective measure of sleepiness and sleep latency via bispectrum analysis of EEG. Med. Biol. Eng. Comput. **48**(12), 1203–1213 (2010)
15. Roy, R.N., Charbonnier, S., Bonnet, S.: Eye blink characterization from frontal EEG electrodes using source separation and pattern recognition algorithms. Biomed. Signal Process. Control **14**, 256–264 (2014)
16. Tan, X., Triggs, B.: Enhanced local texture feature sets for face recognition under difficult lighting conditions. IEEE Trans. Image Process. **19**, 1635–1650 (2010)
17. Zeng, J.: Deep learning based forensic face verification in videos. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 77–80 (2017)
18. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In: Proceedings of the British Machine Vision Conference, pp. 41.1–41.12 (2015)
19. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: DeepFace: closing the gap to human-level performance in face verification. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1701–1708 (2014)

20. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: a unified embedding for face recognition and clustering. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 815–823 (2015)
21. Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet classification with deep convolutional neural networks. In: NIPs Proceedings of Neural Information Processing Systems Conference (2012)