

Robust EMRAN-aided coupled controller for autonomous vehicles

Sauranil Debarshi^a, Suresh Sundaram^{a,*}, Narasimhan Sundararajan^b

^a Department of Aerospace Engineering, Indian Institute of Science, Bangalore, India

^b Technical Consultant, WIRIN Lab, Indian Institute of Science, Bangalore, India

ARTICLE INFO

Keywords:

Autonomous vehicles
Cruise control
Path-tracking
Inverse learning control
EMRAN neural network

ABSTRACT

This paper presents a coupled, neural network-aided longitudinal cruise and lateral path-tracking controller for an autonomous vehicle with model uncertainties and experiencing unknown external disturbances. Using a feedback error learning mechanism, an inverse vehicle dynamics learning scheme utilizing an adaptive Radial Basis Function (RBF) neural network, referred to as the Extended Minimal Resource Allocating Network (EMRAN) is employed. EMRAN uses an extended Kalman filter for online learning and weight updates, and also incorporates a growing/pruning strategy for maintaining a compact network for easier real-time implementation. The online learning algorithm handles the parametric uncertainties and eliminates the effect of unknown disturbances on the road. Combined with a self-regulating learning scheme for improving generalization performance, the proposed EMRAN-aided control architecture aids a basic PID cruise and Stanley path-tracking controllers in a coupled form. Its performance and robustness to various disturbances and uncertainties are compared with the conventional PID and Stanley controllers, along with a comparison with a fuzzy-based PID controller and an active disturbance rejection control (ADRC) scheme. Simulation results are presented for both slow and high speed scenarios. The root mean square (RMS) and maximum tracking errors clearly indicate the effectiveness of the proposed control scheme in achieving better tracking performance in autonomous vehicles under unknown environments.

1. Introduction

Recent technological developments in Advanced Driver Assistance Systems (ADAS) such as adaptive cruise control, lane keeping assistance, and automated parking have opened the doors for different prototypes of autonomous vehicles (AVs) to operate in real-life traffic conditions in public roads. With the number of vehicle fatalities on the rise due to traffic congestion, human error, and lack of safety features, AVs are increasingly gaining attention as solutions to the above problems and for improving general road safety (Eskandarian, 2012). Smart mobility features for perception, planning, control, and situational awareness on-board these vehicles (Schwartz et al., 2018) allow them to safely navigate without any human operator through complex and unstructured environments. One inherent attribute that makes an AV possess such a high level of intelligence is its ability to simultaneously control both the longitudinal and lateral dynamics while ensuring stability and ride comfort.

In autonomous driving, longitudinal control methods are designed to control the speed using the throttle and brake, while a lateral controller automatically steers the vehicle along a reference trajectory (Khodayari et al., 2010). The longitudinal controller maintains a constant speed and keeps a safe distance behind another vehicle, using methods like Adaptive Cruise Control (ACC) (Ntousakis et al.,

2015), emergency brake assist system (Lie et al., 2014), and car-following (Chen et al., 2016). Several control algorithms utilizing Model Predictive Control (MPC) (Shakouri and Ordys, 2014), extremum seeking approach (Dinçmen and Altunel, 2018), proportional-integral (PI) control (Aziziaghdam and Alankus, 2021), and model reference adaptive control (Raffin et al., 2017) have been proposed in the literature to control the longitudinal dynamics. On the contrary, the focus of the lateral control system is to avoid obstacles and assist in lane-keeping and lane-changing maneuvers. Arifin et al. (2019) surveyed some of the current state-of-the-art lateral control methods based on adaptive PID, fuzzy logic, and neural networks. Other steering algorithms present in the literature include a lateral H_∞ controller (Huang et al., 2014), linear quadratic regulator (LQR) (Piao et al., 2019), and a combination of backstepping and sliding mode controllers (SMCs) (Norouzi et al., 2019).

Recently, use of learning-based controllers is increasingly becoming popular in AVs because of their ability to self-optimize and adapt to the dynamics of the environment. Neural controllers, for instance, have been widely used in the literature for AV control. Taghavifar et al. (2020) proposed a type-2 fuzzy neural PID controller to minimize the heading and lateral errors during path-tracking. Additionally, they developed an extended Kalman filter-based adaptive observer to eliminate the effects of external disturbances and parametric uncertainties.

* Corresponding author.

E-mail addresses: sauranild@iisc.ac.in (S. Debarshi), vssuresh@iisc.ac.in (S. Sundaram), ensundara@gmail.com (N. Sundararajan).

Nomenclature**Vehicle Dynamics**

x, y :	Longitudinal/lateral displacement of the vehicle in global frame
\dot{x}, \dot{y} :	Longitudinal/lateral velocity of the vehicle in global frame
v_x, v_y :	Longitudinal/lateral velocity in vehicle frame
\dot{v}_x, \dot{v}_y :	Longitudinal/lateral acceleration of the vehicle
ψ, r, \dot{r} :	Yaw angle, yaw rate, and angular acceleration of the vehicle
i :	Denotes the front (f) and rear (r) wheels
F_{xi}, F_{yi} :	Longitudinal/lateral tire force
ω_{wi} :	Angular velocity of the wheel
I_{wi} :	Moment of inertia of the wheel
R_t, R_{ei} :	Static/effective wheel radius
F_{zi} :	Wheel load
κ_i, α_i :	Tire slip ratio/angle
C_{xi}, C_{yi} :	Longitudinal/lateral tire stiffness
$f_{ai}(), f_{si}()$:	Adhesion/sliding function
σ :	Adhesion potential rate
C_{ksi} :	Longitudinal stiffness in sliding mode
μ_{pxi}, μ_{pyi} :	Peak coefficient of friction in the longitudinal/lateral direction
μ_{sxi}, μ_{syi} :	Sliding coefficient of friction in the longitudinal/lateral direction
l_f, l_r :	Distance to the front/rear axle from the center of gravity (CoG) of the vehicle
I_z :	Vehicle's yaw moment of inertia
m :	Mass of the vehicle
T_d, T_b :	Drive/brake torque

EMRAN Parameters

\mathbf{v} :	Network inputs
μ_k, σ_k :	Center and width of the k th hidden neuron
h :	Number of hidden neurons
z^k :	Output of the k th hidden neuron with Gaussian activation function
α_{jk} :	Interconnection weight between the k th hidden neuron and the j th output
α_{0j} :	Bias at the j th output
u_{ad}^j :	Output of the neural network
p :	Total number of outputs of the network
$\epsilon_1, \epsilon_2, \epsilon_3$:	Thresholds for adding hidden neurons
y_e :	Error in the network output
w :	Represents the hidden neuron closest to \mathbf{v} , called the “winner neuron”
μ_w :	Center of the “winner neuron”
γ :	Decay constant between 0 and 1
S_w :	Sliding window's length
κ :	Overlap factor

\mathbf{NP}^w :	Connection weights, centers and widths of the “winner neuron”
\mathbf{K}^w :	Kalman gain matrix
\mathbf{B}^w :	Gradient matrix
\mathbf{R} :	Variance of the measurement noise
\mathbf{P}^w :	Error of the covariance matrix
δ :	Threshold for pruning a hidden neuron
N_w :	Number of consecutive input samples for which the output of a hidden neuron is less than δ
l :	Number of parameters of the “winner neuron”

Other variables

v_r, e_v :	Reference cruising speed/velocity error
y_r, e_y :	Reference lateral position in global frame/lateral error
ψ_r, e_ψ :	Reference yaw angle/heading error
\mathbf{y}_d :	Reference signal to be tracked by the vehicle
\mathbf{x}, \mathbf{y} :	State vector/output of the vehicle
\mathbf{u}_v :	Control input to the vehicle
u_{pid} :	Acceleration/deceleration commands from the PID controller
u_{nn} :	Acceleration/deceleration from the longitudinal EMRAN
u_i :	Total longitudinal control input
δ_s :	Steering angle from the Stanley controller
δ_{nn} :	Steering angle from the lateral EMRAN
δ_f :	Total steering input
\mathbf{x}_{ec} :	Input vector of the longitudinal EMRAN
\mathbf{x}_l :	Input vector of the lateral EMRAN

of the speed. The RBF network with its adaptive and universal function approximation ability reduces the large errors of the SMC and improves the robustness of the system. More recently, a comprehensive study on a wide range of longitudinal and lateral vehicle control methods based on deep learning was conducted by Kuutti et al. (2021). Various supervised and reinforcement learning strategies were compared in terms of network complexity, learning capabilities, and performance. It has been concluded that while deep learning has shown great promise in AV control applications, it also presents numerous challenges for actual deployment in vehicles. The need for large amount of data that captures all possible driving scenarios, computational effort, and selection of network architecture limits the use of deep learning-based control in AVs.

It should be noted that much of the existing research addresses the longitudinal and lateral controllers separately, assuming that there is no dynamic interaction (coupling) between them. However, it is not the case in an actual vehicle, and in many critical autonomous driving scenarios, a coupled control strategy is required (Amer et al., 2017). A simultaneous longitudinal and lateral control architecture was presented in Attia et al. (2014). A nonlinear MPC was employed as the steering controller and a Lyapunov-based control law considering the powertrain dynamics was used for the longitudinal speed-tracking. Another coupled method was shown in Devineau et al. (2018) by quantitatively comparing two different deep learning models, a Multi-layer Perceptron (MLP) and a Convolutional Neural Network (CNN). The CNN proved to be a better controller in terms of accuracy and smoothness of input commands. A recent study by Tork et al. (2021) describes an integrated control system using an adaptive multi-layer

A backstepping variable structure control (BVSC) coupled to an RBF neural network (RBFNN) was presented in Ji et al. (2018). The BVSC steers the vehicle, while the RBFNN eliminates the errors by acting as the estimator for the tire nonlinearities. In another study, Wang et al. (2019) proposed an SMC with an RBFNN for improving the tracking

neural network based on an interval type-2 fuzzy activation function. They performed Double Lane Change (DLC) maneuvers with and without parametric uncertainties, and their proposed controller outperformed other neural network-based controllers. However, effect of external disturbances such as slippery road conditions and different velocities were not taken into account in their study.

In this paper, we propose a novel neuro-aided coupled longitudinal and lateral control scheme for an AV by utilizing an inverse vehicle dynamics learning technique. We employ a previously developed RBF neural network referred to as the Extended Minimal Resource Allocating Network (EMRAN) (Li et al., 2000) and integrate a self-regulated learning mechanism (Suresh et al., 2010) as an extension for efficient training with fewer samples. EMRAN is a fast, adaptive, sequential learning algorithm that requires no a priori training and overcomes the challenges of deep learning-based controllers discussed earlier. It starts with zero hidden neurons and incorporates a growing/pruning strategy, making it a computationally inexpensive learning algorithm with a compact and efficient network suitable for real-time implementation in AVs. To the best of the authors' knowledge, no prior studies have examined an online inverse vehicle dynamics learning model for developing an integrated control system that is robust to unknown disturbances and uncertainties. Another major drawback of most of the learning-based approaches such as Deep Neural Networks (DNNs), is that they scale poorly and provide no generalization guarantee (Kuutti et al., 2021). The online inverse dynamics learning method addresses this issue by allowing a neural controller to be used in conjunction with any feedback controller for improving the tracking performance. The main contributions of the paper are as follows:

- A model-free, coupled EMRAN-aided controller is proposed for longitudinal cruise control and lateral path-tracking. A conventional PID aided by an EMRAN neural network (PID-EMRAN) provides the acceleration/deceleration commands for maintaining the desired velocity. Another EMRAN is used to aid a Stanley (Hoffmann et al., 2007) controller (Stanley-EMRAN) for steering the vehicle along a reference trajectory.
- The EMRAN-aided controller supports online adaptability and learns to approximate the inverse dynamics of the vehicle using a feedback error learning strategy. It ensures stability and provides robustness against various unknown disturbances and parametric uncertainties.
- The neuro-aided control architecture could be used as an add-on with any feedback controller for improving the tracking performance, thereby providing flexibility in the overall controller design.

The performance of the EMRAN-aided controller has been evaluated against conventional PID and Stanley methods, for both coupled and decoupled states. Additionally, the proposed controller is compared with a recently developed type-2 Fuzzy PID controller (Taghavifar et al., 2020) and an active disturbance rejection control scheme (Xia et al., 2016). The obtained results highlight the benefits of using EMRAN for real-time longitudinal and lateral control of AVs.

Rest of the paper is organized as follows: In Section 2, the problem statement related to AV control is formulated. Section 3 discusses the EMRAN-aided inverse dynamics learning controller in detail. Performance evaluation using simulation results is shown in Section 4. Finally, the conclusions based on the study are summarized in Section 5.

2. Problem formulation

Before presenting the mathematical model of the nonlinear vehicle used for the simulations, we first formulate the control problem in AVs to track a reference signal y_d as:

$$y_d = \{v_r, \{y_r, \psi_r\}\} \quad (1)$$

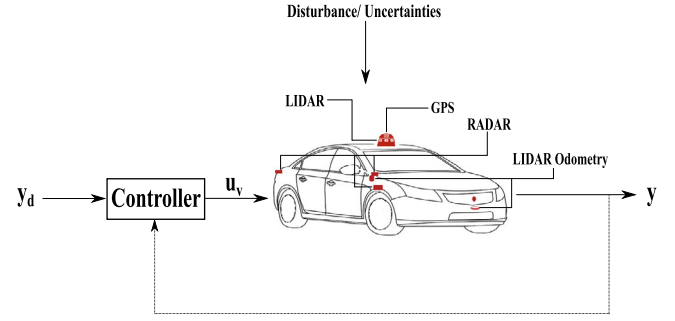


Fig. 1. Representation of a typical vehicle control architecture.

where v_r represents the reference cruise speed, and y_r and ψ_r respectively denote the reference lateral displacement and desired yaw angle of the vehicle. These reference signals are generated by a motion planning algorithm in an actual AV and a discussion on the same is outside the scope of the current study. This paper only investigates the longitudinal and lateral control problem in AVs through cruise control and DLC maneuvers.

The state-space model of the vehicle at time t is given by:

$$\dot{\mathbf{x}} = g(\mathbf{x}, \mathbf{u}_v, t) \quad (2)$$

where the dynamics of the vehicle is denoted by the nonlinear function $g()$. $\mathbf{u}_v \in \mathbb{R}^q$ is the control input and $\mathbf{x} \in \mathbb{R}^m$ is the state vector. Also, \mathbf{u}_v belongs to a class of permissible inputs, given by:

$$U := \{\mathbf{u}_v : \|\mathbf{u}_v(t)\| \leq \zeta, t > t_0\} \quad (3)$$

where ζ is a real positive number and it puts a constraint on the control input such that the vehicle follows the reference signals as closely as possible after a certain time instant, t_0 , and without losing stability. With the constraint on \mathbf{u}_v , we define the control objective as:

$$\|\mathbf{y}_d - \mathbf{y}\| \rightarrow 0 \quad (4)$$

where \mathbf{y}_d is the reference signal and \mathbf{y} is the output of the vehicle.

Fig. 1 shows the representation of an AV with multiple onboard sensors to replicate the human understanding of its location, perception, and navigation. The sensors provide critical information to the controller for generating the control input \mathbf{u}_v , such that the vehicle can track the reference signal \mathbf{y}_d with minimum errors. For the purpose of this study, a nonlinear single-track or “bicycle model” is used for the simulations, as depicted in Fig. 2. Due to the symmetry of the left and right side of a vehicle, the bicycle model provides a simplified, and at the same time, an accurate representation of the vehicle's dynamics.

Plessen et al. (2018) and Rajamani (2011) describe the dynamics of the bicycle model, with the longitudinal (\dot{x}) and lateral (\dot{y}) velocities in the global frame given by:

$$\begin{aligned} \dot{x} &= v_x \cos \psi - v_y \sin \psi \\ \dot{y} &= v_x \sin \psi + v_y \cos \psi \end{aligned} \quad (5)$$

where, v_x and v_y represent the longitudinal and lateral velocities in the vehicle frame respectively and ψ is the yaw angle. The angular rate ($\dot{\psi}$) and acceleration (\dot{r}) are defined as:

$$\begin{aligned} \dot{\psi} &= r \\ \dot{r} &= \frac{1}{I_z} (F_{yf} l_f - F_{yr} l_r) \end{aligned} \quad (6)$$

where, F_{yf} is the lateral tire force of the front wheel and F_{yr} is the lateral tire force of the rear wheel. l_f and l_r are respectively the distances to the front and rear axles from the center of gravity (CoG) of the vehicle, I_z is the yaw moment of inertia, and r is the yaw rate.

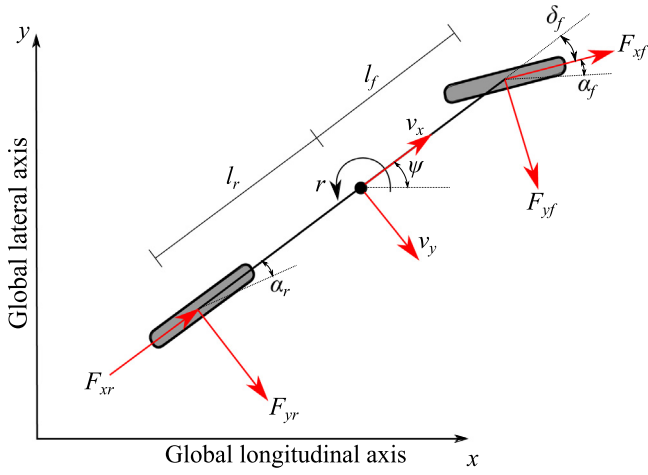


Fig. 2. Nonlinear single-track vehicle dynamics model.

The vehicle's longitudinal (\$\dot{v}_x\$) and lateral (\$\dot{v}_y\$) acceleration are obtained using:

$$\begin{aligned}\dot{v}_x &= \frac{1}{m}(F_{xf} + F_{xr}) + v_y r \\ \dot{v}_y &= \frac{1}{m}(F_{yf} + F_{yr}) - v_x r\end{aligned}\quad (7)$$

where, \$F_{xf}\$ and \$F_{xr}\$ denote the longitudinal tire forces of the front and rear wheels respectively and \$m\$ is the mass of the vehicle.

The angular velocities of the front (\$\omega_{wf}\$) and rear (\$\omega_{wr}\$) wheels are calculated from the drive (\$T_d\$) and brake (\$T_b\$) torques as:

$$\begin{aligned}\omega_{wf} &= \frac{1}{I_w}(T_d - R_t F_{xf} - T_b) \\ \omega_{wr} &= \frac{1}{I_w}(T_d - R_t F_{xr} - T_b)\end{aligned}\quad (8)$$

where, \$R_t\$ is the static radius of the wheel with moment of inertia \$I_w\$.

For estimating the forces on the wheels in the longitudinal (\$F_x\$) and lateral (\$F_y\$) directions, an analytical tire model developed by [Salaani \(2007\)](#) has been used. This nonlinear tire model captures the wheel-road dynamics accurately and has been validated from experimental data using four different tires. The longitudinal and lateral wheel forces are given as:

$$\begin{aligned}F_{xi} &= -\frac{\kappa_i F_{zi} C_{\kappa i} f_{ai}(\sigma)}{\sqrt{\left(\frac{C_{\kappa i} \kappa_i}{\mu_{pxi}}\right)^2 + \left(\frac{C_{ai} \tan(\alpha_i)}{\mu_{pyi}}\right)^2}} \\ &\quad - \frac{\kappa_i F_{zi} C_{\kappa si} f_{si}(\sigma)}{\sqrt{\left(\frac{C_{\kappa si} \kappa_i}{\mu_{sxi}}\right)^2 + \left(\frac{C_{ai} \tan(\alpha_i)}{\mu_{syi}}\right)^2}} \quad (i = f, r) \\ F_{yi} &= \frac{C_{ai} \tan(\alpha_i) F_{zi} f_{ai}(\sigma)}{\sqrt{\left(\frac{C_{\kappa i} \kappa_i}{\mu_{pxi}}\right)^2 + \left(\frac{C_{ai} \tan(\alpha_i)}{\mu_{pyi}}\right)^2}} \\ &\quad + \frac{C_{ai} \tan(\alpha_i) F_{zi} f_{si}(\sigma)}{\sqrt{\left(\frac{C_{\kappa si} \kappa_i}{\mu_{sxi}}\right)^2 + \left(\frac{C_{ai} \tan(\alpha_i)}{\mu_{syi}}\right)^2}} \quad (i = f, r)\end{aligned}\quad (9)$$

where \$F_{zi}\$, \$\kappa_i\$, and \$\alpha_i\$ are the wheel load, longitudinal slip ratio, and slip angle, respectively. \$C_{\kappa i}\$ and \$C_{ai}\$ denote the longitudinal and lateral tire stiffness, \$f_{ai}(\cdot)\$ and \$f_{si}(\cdot)\$ are the adhesion and sliding functions, \$\sigma\$ is the adhesion potential rate, and lastly \$C_{\kappa si}\$ represents the longitudinal stiffness in sliding mode. The peak/sliding coefficients of friction in the longitudinal and lateral directions are given by \$\mu_{pxi}\$, \$\mu_{sxi}\$, \$\mu_{pyi}\$, and \$\mu_{syi}\$, where the subscript \$i\$ denotes the front and rear tires.

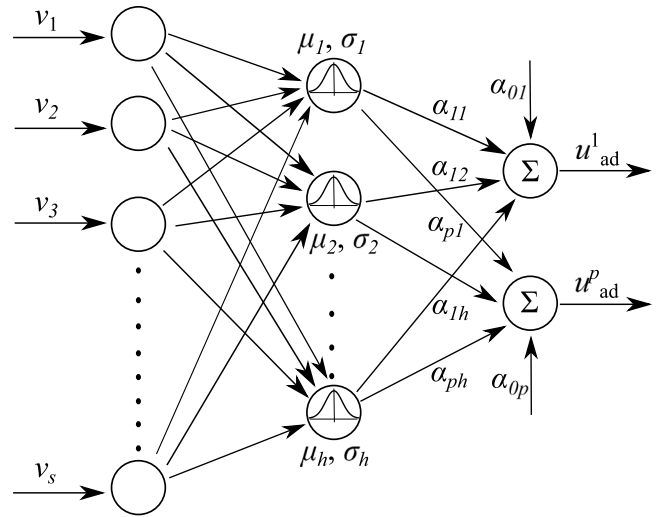


Fig. 3. EMRAN architecture with Gaussian activation functions.

Using small angle approximations, the tire slip angle in the front (\$\alpha_f\$) and rear (\$\alpha_r\$) can be linearized ([Liu et al., 2018](#)) as:

$$\begin{aligned}\alpha_f &= \arctan\left(\frac{\dot{y} + l_f \dot{\psi}}{\dot{x}}\right) - \delta_f \approx \frac{\dot{y} + l_f \dot{\psi}}{\dot{x}} - \delta_f \\ \alpha_r &= \arctan\left(\frac{\dot{y} - l_r \dot{\psi}}{\dot{x}}\right) \approx \frac{\dot{y} - l_r \dot{\psi}}{\dot{x}}\end{aligned}\quad (10)$$

where \$\delta_f\$ represents the steering angle. Additionally, the tire slip ratio (\$\kappa_i\$) ([Suzuki and Fujimoto, 2010](#)) is defined as:

$$\kappa_i = \frac{R_{ei} \omega_{wi} - v_x}{\max(R_{ei} \omega_{wi}, v_x, \epsilon)} \quad (i = f, r) \quad (11)$$

where \$R_{ei}\$ is the effective tire radius and \$\epsilon\$ is a small constant (\$\epsilon \ll 1\$) to avoid zero denominator.

Next, we describe the online inverse dynamics learning controller for AVs in detail.

3. Inverse dynamics learning EMRAN controller

In this section, the EMRAN-aided coupled longitudinal and lateral controller is presented. An inverse dynamics learning with self-regulation is proposed to achieve the tracking objective. The working of EMRAN is briefly discussed next.

3.1. Extended Minimal Resource Allocating Network (EMRAN)

EMRAN is a fast, online learning algorithm developed by [Li et al. \(2000\)](#), ideal for real-time application. It implements a compact RBF neural network by incorporating a fully adaptive learning strategy, with the capability to add and prune the hidden neurons based on the network inputs and the responses of the controlled object. It is an extension of the previously developed sequential learning RBF algorithm called the Minimal Resource Allocation Network (MRAN) ([Lu et al., 1997, 1998](#)). MRAN also utilizes a growing/pruning strategy for ensuring compactness. However, unlike EMRAN in which the parameters of only the nearest hidden neuron (in a norm sense) are updated, MRAN updates the centers, widths and weights of all the hidden neurons at every timestep. This causes the size of the matrices to be updated to become large as the hidden neurons increase and the RBF network structure becomes more complex computationally, thereby limiting the use of MRAN for real-time implementations.

Fig. 3 shows the EMRAN architecture consisting of a single hidden layer with \$h\$ hidden neurons. It initially starts with zero hidden neurons

and they are added/pruned based on a heuristic condition. The coefficients α_{jk} are the interconnection weights between the hidden and the output layer. The activation functions used in the hidden layer are Gaussian, whose outputs are given by:

$$z^k = \exp \frac{-\|\mathbf{v} - \boldsymbol{\mu}_k\|^2}{2(\sigma_k)^2} \quad (k = 1, 2, \dots, h) \quad (12)$$

where $\mathbf{v} \in \mathbb{R}^s$ is the network input, while $\boldsymbol{\mu}_k, \sigma_k \in \mathbb{R}$ are the center and width of the hidden neurons respectively. Gaussian activation functions have good local interpolation (each neuron responds only to a specified region of the input space) and global approximation ability (Lu et al., 1997; Li et al., 2000). EMRAN, thus, explicitly stores information regarding the input characteristics, instead of merely using the information for updating the network parameters. The outputs of the neural network are then given by:

$$u_{ad}^j = \sum_{k=1}^h \alpha_{jk} z^k + \alpha_{0j} \quad (j = 1, 2, \dots, p) \quad (13)$$

where α_{0j} are the biases at the output layer and p is the total number of outputs of the network.

EMRAN starts with no hidden neurons. A new neuron is added when the following criteria are satisfied at any time step, τ :

$$\begin{aligned} \|\mathbf{v}[\tau] - \boldsymbol{\mu}_w[\tau]\| &> \epsilon_1[\tau] \\ \|\mathbf{y}_e[\tau]\|^2 &\geq \epsilon_2 \\ J_{rmse} &= \sqrt{\frac{\sum_{\tau=S_w+1}^{\tau} \|\mathbf{y}_e[\tau]\|^2}{S_w}} \geq \epsilon_3 \end{aligned} \quad (14)$$

where $\epsilon_1[\tau] = \max[\epsilon_{max} \gamma^{\tau-1}, \epsilon_{min}]$. EMRAN starts with $\epsilon_1[\tau] = \epsilon_{max}$, the largest scale of interest, which is typically the size of the entire input space of nonzero probability density, and decays exponentially until it reaches ϵ_{min} . γ is the decay constant between 0 and 1 and represents the scale of resolution (Platt, 1991). $\mathbf{y}_e[\tau]$ is the error in the network output, $\boldsymbol{\mu}_w$ is the center of the hidden neuron closest to $\mathbf{v}[\tau]$, referred to as the “winner neuron” and represented by the subscript w . S_w is the sliding window’s length and ϵ_1 , ϵ_2 , and ϵ_3 are the thresholds that needs to be selected appropriately. The distance between the new observation and all the existing nodes is compared by the first criterion. The second criterion ascertains that the existing neurons are sufficient to produce a reasonable network output. The third criterion is based on the root mean square error for the window of samples S_w , which controls the noise from over-fitting the neurons. The parameters of the newly added hidden neuron are given by:

$$\begin{aligned} \alpha_{k+1}[\tau] &= \mathbf{y}_e[\tau - 1] \\ \boldsymbol{\mu}_{k+1}[\tau] &= \mathbf{v}[\tau] \\ \sigma_{k+1}[\tau] &= \kappa \|\mathbf{v}[\tau] - \boldsymbol{\mu}_w[\tau]\| \end{aligned} \quad (15)$$

where κ determines the overlap of the responses of the hidden neurons in the input space.

When the above criteria are not satisfied, an Extended Kalman Filter (EKF) updates the parameters of the neuron whose center is nearest to the network input data \mathbf{v} (“winner neuron”), whose parameters (\mathbf{NP}^w = connection weights, centers, and widths) at the τ th instant are updated as:

$$\mathbf{NP}^w[\tau] = \mathbf{NP}^w[\tau - 1] + \mathbf{K}^w[\tau] \mathbf{y}_e[\tau - 1] \quad (16)$$

where $\mathbf{K}^w[\tau]$ is the Kalman gain matrix given by:

$$\mathbf{K}^w[\tau] = \mathbf{P}^w[\tau - 1] \mathbf{B}^w[\tau] (\mathbf{R}[\tau] + \mathbf{B}^w[\tau]^T \mathbf{P}^w[\tau - 1] \mathbf{B}^w[\tau])^{-1} \quad (17)$$

where $\mathbf{B}^w[\tau] = \nabla_{\mathbf{w}} \mathbf{u}_{ad}$ is the gradient matrix of \mathbf{u}_{ad} with respect to the parameter vector $\mathbf{w}[\tau]$ evaluated at $\mathbf{w}[\tau - 1]$. $\mathbf{R}[\tau]$ is the variance of the measurement noise and $\mathbf{P}^w[\tau]$ is the error of the covariance matrix, which is updated by:

$$\mathbf{P}^w[\tau] = (\mathbf{I} - \mathbf{K}^w[\tau] \mathbf{B}^w[\tau]^T) \mathbf{P}^w[\tau - 1] + q \mathbf{I} \quad (18)$$

where q is a scalar quantity that determines the allowed random step in the direction of the gradient vector. When a new hidden unit is added to the network, the dimensionality of the covariance matrix increases to:

$$\mathbf{P}^w[\tau] = \begin{bmatrix} \mathbf{P}^w[\tau - 1] & 0 \\ 0 & P_0 \mathbf{I} \end{bmatrix} \quad (19)$$

where P_0 is a scalar value that represents the uncertainty in the initial parameters of the new hidden neuron.

EMRAN incorporates a pruning strategy for maintaining a compact network. It ensures that the neurons that have not been contributing significantly (based on a threshold parameter (δ)) to the network performance for a predefined period of time (N_w), are pruned from the network. In addition, two hidden neurons are combined into a single hidden neuron if they are found to be closer to one another, as defined by a threshold value. It results in a network that is computationally inexpensive and adapted to fast real-time online applications. Moreover, since only the parameters of the winner neuron are updated, the computations required to update its parameters is $O(l^3)$, where l is the number of parameters of the nearest neuron. Therefore, the total computational burden at each step is 8^3 floating point operations (FLOPS) ($1 \text{ FLOP} = 1e^{-06} \text{ s}$), which is relatively small. Control output calculation is $O(h)$, where h is the number of hidden neurons.

To further improve the online learning process and avoid over-training the EMRAN network, a self-regulated learning scheme (Suresh et al., 2010; Savitha et al., 2012) has been incorporated. The EKF, instead of using every training samples for updating the parameters of the winner neuron, utilizes only a subset of the training data based on certain thresholds of the tracking and residual errors. It achieves similar control performance to a conventionally trained network while using fewer samples, allowing the EMRAN network to learn faster, with reduced computational effort. Next, the EMRAN-aided control architecture and its functioning for AV control are described.

3.2. EMRAN-aided inverse vehicle dynamics learning

Consider the dynamics of the vehicle in Eqs. (5)–(7) to be in the form of:

$$\dot{\mathbf{x}} = g(\mathbf{x}, \mathbf{u}_v) \quad (20)$$

where $\mathbf{u}_v = [u_t, \delta_f]$ constitutes the control input vector to the AV, with u_t representing the acceleration/deceleration command for longitudinal cruise control and δ_f denoting the steering angle for lateral path-tracking. In this paper, we utilize the feedback error learning technique of Gomi and Kawato (1990) to learn the inverse dynamics of the vehicle using the EMRAN neural network. By learning the inverse dynamics, EMRAN can compensate for the nonlinearities of the vehicle such that it follows the desired response set closely. The inversion can be represented by the equation:

$$\mathbf{u}_v = g^{-1}(\dot{\mathbf{x}}, \mathbf{x}) \quad (21)$$

Further, if the function $g^{-1}()$ is changing with time due to external factors, the neural controllers can generate immediate corrective actions to compensate for such changes.

The architectures of the EMRAN-aided longitudinal and lateral control subsystems of the coupled controller are shown in Fig. 4, with their respective control input to the AV given by:

$$\begin{aligned} u_t &= u_{pid} + u_{nn} \\ \delta_f &= \delta_s + \delta_{nn} \end{aligned} \quad (22)$$

The outputs of the baseline PID and Stanley controllers are denoted by a vector, $\mathbf{u}_b = [u_{pid}, \delta_s]$ while their respective EMRAN outputs are represented by another vector as $\mathbf{u}_{nn} = [u_{nn}, \delta_{nn}]$. The outputs of the AV, namely, the velocity, lateral position and yaw rate are given as feedbacks to calculate their respective errors, which in turn form the inputs to the baseline controllers. In both the architectures, the

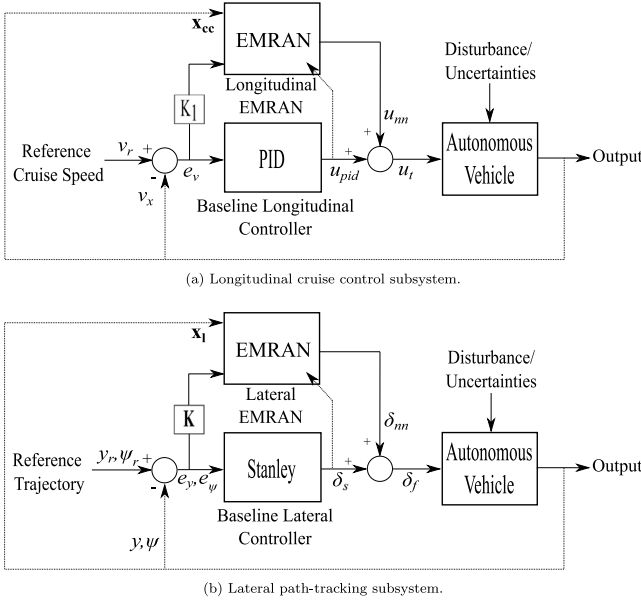


Fig. 4. Schematic of the EMRAN-aided longitudinal and lateral control architectures.

baseline controllers provide the basic stability requirements and also generate the signals to train their EMRAN networks online. The EMRANs compensate for the disturbances and parametric uncertainties of the vehicle, thereby aiding the baseline controllers to achieve a better tracking performance. The online learning process of the proposed controller is discussed next.

3.3. Online learning process

The inner loop of the longitudinal cruise control uses a fully tuned PID controller as the baseline controller, as shown in Fig. 4a. It generates acceleration and deceleration commands (u_{pid}) based on the differences between the reference (v_r) and actual (v_x) velocities. The input vector (x_{cc}) to the cruise control EMRAN consists of the longitudinal states of the vehicle, namely the position (x), velocity (v_x), and acceleration (\dot{v}_x). Sensor information from the GPS, LIDAR, and Vehicle speed sensor (VSS) onboard an AV could be used to calculate the states in real-time and given as inputs to the controllers, whose mathematical representation is given by:

$$\begin{aligned} e_v &= v_r - v_x \\ x_{cc} &= [x, v_x, \dot{v}_x] \end{aligned} \quad (23)$$

In practice, these states are susceptible to sensor noises, which can degrade the overall performance of the controllers. However, in this work, all the states of the vehicle are measured from the simulation environment and are assumed to be noise-free. Based on the postulate of Gomi and Kawato (1990), the output of the PID controller ($u_{pid} = u_t - u_{nm}$) is used as the signal for updating the weights and neurons of the network. EMRAN learns the total control signal ($u_{nm} \rightarrow u_t$) over time and eventually driving the PID output to zero. This means that EMRAN has generated the inverse longitudinal dynamics of the vehicle and is using it for control. However, if EMRAN learns only from the baseline controller, there is the possibility that it will not achieve any better results than the PID, since the conventional controllers are not robust against unknown disturbances and uncertainties. Hence, to get a better performance and eliminate the effects of unwanted nonlinearities, the learning signal is modified by adding the output of the PID controller with a scaled (K_1) velocity error signal. The drive torque (T_d) and brake torque (T_b) are ultimately computed from the acceleration/deceleration command (u_t) using dynamic equations.

Similarly, for the lateral path-tracking subsystem, a Stanley controller (Hoffmann et al., 2007) is used in the inner loop, as depicted in Fig. 4b. The Stanley controller is a geometrical path-tracking algorithm developed by the Stanford University's DARPA grand challenge team. It is a nonlinear feedback function of the cross-track/lateral (e_y) and heading (e_ψ) errors measured from the vehicle's front axle. The steering angle is given by:

$$\delta_s = e_\psi + \tan^{-1} \left(\frac{k_f e_y}{v_x} \right) \quad (24)$$

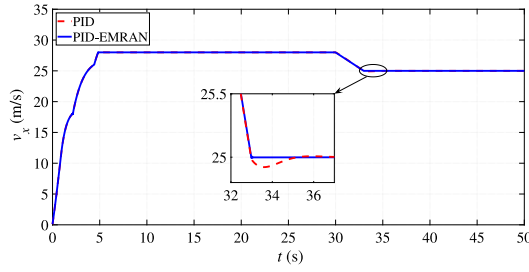
where k_f is the position gain in the forward motion, and v_x is the longitudinal velocity of the vehicle. Moreover, as it considers both the heading and the lateral errors, it has shown to perform well in previous studies (Paden et al., 2016; Amer et al., 2019). Similar to the longitudinal control architecture, the lateral states of the vehicle are given as inputs (x_l) to the path-tracking EMRAN. The output of the Stanley controller (δ_s) added to the scaled ($K = [K_2, K_3]$) error signal, is given as the excitation signal to the EMRAN network for learning the inverse lateral dynamics and improving the robustness.

4. Performance evaluation of EMRAN-aided controller for AVs

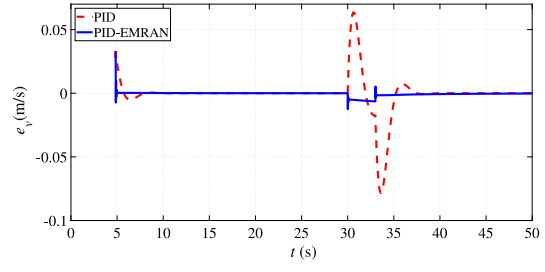
In this section, the performances of the proposed EMRAN-aided PID (PID-EMRAN) and Stanley (Stanley-EMRAN) controllers for a typical AV are presented for various test cases with and without external disturbances/uncertainties. As defined by the control objective in Eq. (4), the controllers have to minimize the tracking errors such that they rapidly converge to zero, and with minimum overshoots and steady-state errors. Moreover, the objective has to be achieved without degrading the stability of the vehicle during the maneuvers. As a first step, simulations were conducted for the longitudinal cruise control to evaluate the PID-EMRAN controller against a conventional PID method. Then, the Stanley-EMRAN controller is assessed through DLC maneuvers for lateral path-tracking at slow as well as high speed scenarios. Note that the effect of the longitudinal vehicle dynamics on the path-tracking performance was considered negligible during the simulations with the Stanley-EMRAN controller. However, in practice, it cannot be overlooked since the longitudinal dynamics can affect the performance of the lateral controller during maneuvers such as lane changing. To address this issue, an integrated PID-EMRAN and Stanley-EMRAN controller, referred to as the Coupled-EMRAN was employed, which took into account the dynamics of both the longitudinal and lateral directions of the vehicle. Finally, quantitative comparisons of the lateral Stanley-EMRAN controller with a fuzzy logic-based method and an active disturbance rejection control scheme are shown.

Simulations have been performed in MATLAB/Simulink-Unreal Engine (UE) 4 interface on a system with Ryzen 9 5900HX processor, 32 GB of memory and Nvidia RTX 3080 GPU. This configuration was chosen to support UE visualization. The physical parameters of the vehicle used for the simulation studies are given in Table 1, where C_f and C_r are the stiffness values of the front and rear tires, respectively. Moreover, the hyperparameters (ϵ_{max} , ϵ_{min} , γ , ϵ_2 , ϵ_3 , δ , N_w , S_w , κ , P_0 , q , and r) associated with EMRAN are problem-dependent and are determined offline through an optimization by a Genetic Algorithm (GA) for achieving the best results. Note that these hyperparameters remain constant throughout the simulations. More detailed descriptions of the parameters can be obtained from Kadirkamanathan and Niranjan (1993) and Lu et al. (1997). The GA parameters used in finding the best EMRAN parameters are as follows: crossover probability of 0.8, selection probability of 0.08, mutation probability of 0.15, maximum number of generations of 10, and a population size of 20. For a detailed description of optimization using a genetic algorithm, refer to Suresh et al. (2014).

For optimizing the hyperparameters associated with both the EMRAN networks, offline simulations involving speed tracking and DLC maneuvers were conducted. A reference cruise speed of 15 m/s was



(a) Cruise speed profile.



(b) Velocity tracking error.

Fig. 5. Velocity tracking without disturbances and uncertainties.

Table 1

Vehicle parameters.

Parameters	Values
m	1480 kg
I_z	2350 kg m ²
l_f	1.05 m
l_r	1.63 m
C_f	67 500 N/rad
C_r	47 500 N/rad

Table 2

GA optimized parameters of the longitudinal and lateral EMRAN controllers.

Parameters	Longitudinal	Lateral
e_{max}	7.455	4.003
e_{min}	3.938	3.086
γ	0.915	0.981
e_2	0.357	0.005
e_3	0.071	0.003
δ	0.091	0.073
N_w	12	9
S_w	10	14
κ	0.609	0.603
P_0	1.079	1.155
q	0.015	0.001
r	1.074	1.120

Table 3

RMS and maximum values of cruise speed tracking errors without disturbances.

Controller	$e_{v_{rms}}$ (m/s)	$e_{v_{max}}$ (m/s)
PID	0.0149	0.0784
PID-EMRAN	0.0017	0.0332

set for the PID aiding EMRAN to track without any disturbances and uncertainties. The optimized values were determined using GA by minimizing the objective of the fitness function, i.e., the RMS error between the reference and actual speeds. Similarly for the path-tracking EMRAN controller, a decoupled DLC maneuver at a constant velocity of 10 m/s was performed at ideal conditions, with the RMS lateral error as the objective of the fitness function to be reduced. The optimized hyperparameters of both the networks are given in Table 2.

4.1. Longitudinal cruise control

4.1.1. No disturbance

To evaluate the performance of the PID-EMRAN controller, ideal conditions without disturbances and uncertainties have been considered. The vehicle's cruising speed is changed from 28 m/s to 25 m/s at time $t = 30$ s, and it is assumed that the vehicle travels on a dry asphalt ($\mu = 1.0$), straight road. The proportional (K_p), integral (K_I), and derivative (K_D) gains associated with the baseline PID controller were tuned using GA and set as 1.841, 2.603, and 0.682, respectively.

The performances of the conventional PID and neuro-aided PID-EMRAN controllers to track the reference cruise speed are shown in

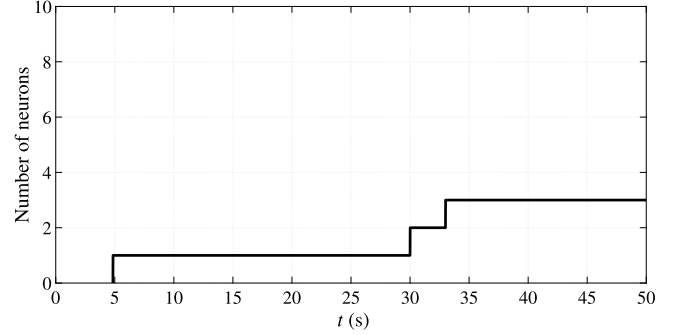


Fig. 6. Neuron history of PID aiding EMRAN for tracking the cruise speed without any disturbances.

Table 4

Cruise speed tracking errors in the presence of disturbance.

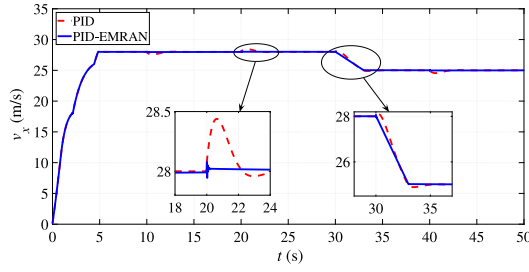
Controller	$e_{v_{rms}}$ (m/s)	$e_{v_{max}}$ (m/s)
PID	0.1368	0.5145
PID-EMRAN	0.0163	0.0817

Fig. 5. The PID-EMRAN controller outperforms the PID-based approach by lowering the overshoots and undershoots. Table 3 validates this claim, which shows the RMS ($e_{v_{rms}}$) and maximum ($e_{v_{max}}$) values of the errors. Significant improvements are observed, with the PID-EMRAN controller reducing both the $e_{v_{rms}}$ and $e_{v_{max}}$ by 88.59% and 57.65%, respectively. The neuron growth history in EMRAN is also shown in Fig. 6. The EMRAN controller adds hidden neurons to compensate for the errors during the change in cruise speed, thus tracking the reference signal more accurately during the transition.

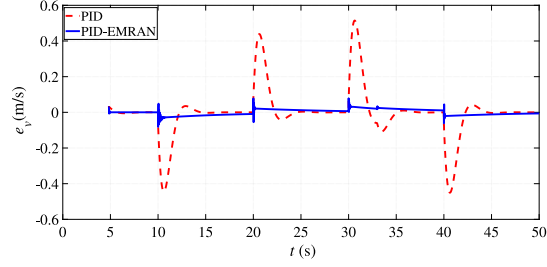
4.1.2. External disturbances and parametric uncertainties

To evaluate the disturbance rejection ability of the proposed PID-EMRAN controller under extreme conditions, the vehicle was subjected to road inclinations of $\pm 40^\circ$. The vehicle starts on a level road initially ($t < 10$ s) and encounters an uphill road with an inclination of 40° from $t = 10$ s to $t = 20$ s. At $t = 20$ s, the slope reduces to zero, and the vehicle starts moving on the level road for another 10 s. Similar test cases were performed for the subsequent duration of the simulation, in which the vehicle started descending on a steep road of slope -40° from $t = 30$ s to $t = 40$ s, after which it started moving on a level road again.

Fig. 7 shows the performances of both the conventional PID and EMRAN-aided PID controllers in the presence of the road disturbances. The proposed control architecture shows robustness against the varying road inclinations by minimizing the overshoot/undershoots and the steady-state errors. The vehicle also maintains smooth transitions in cruise speed. It has been verified quantitatively in Table 4, which shows the RMS and maximum errors. For the case with disturbance, PID-EMRAN reduces the tracking errors by 86.88% ($e_{v_{rms}}$) and 81.85% ($e_{v_{max}}$). The neuron history is also shown in Fig. 8. It can be inferred



(a) Cruise speed profile.



(b) Velocity tracking error.

Fig. 7. Velocity tracking with varying road inclinations.

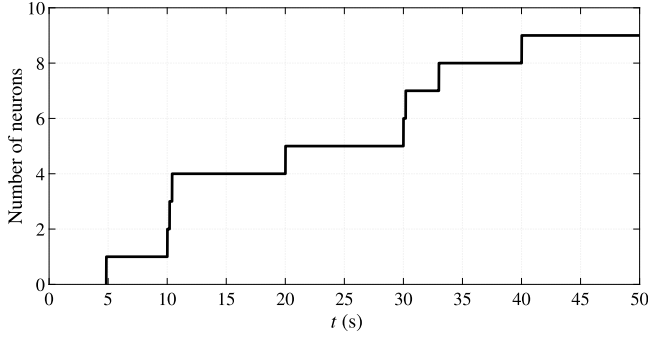


Fig. 8. Neuron history of longitudinal EMRAN for varying road inclinations.

Table 5

Tracking errors with parametric uncertainties.

Controller	$e_{v_{rms}}$ (m/s)	$e_{v_{max}}$ (m/s)
PID	0.1463	0.2678
PID-EMRAN	0.0076	0.0340

by comparing Figs. 6 and 8 that the EMRAN controller constantly adds new neurons to mitigate the errors due to the disturbances, thus showing the effectiveness in its online learning ability. Because of the various instances of road inclinations during the simulation, no pruning of hidden neurons is observed.

Similarly, uncertainties related to vehicle parameters, ($m = m + 0.15m \sin(t)$) and ($I_y = I_y + 0.2I_y \sin(t)$) were included, where I_y is the pitch moment of inertia. Additionally, environmental perturbations of the road-friction coefficient ($\mu = \mu + 0.5\mu \sin(t)$) and wind velocity ($V_w = 15 * \sin(t)$) in the longitudinal direction were also considered.

Table 5 shows that even with large parametric uncertainties and external perturbations, the proposed longitudinal controller can track the reference cruise speed with minimum errors. The PID-EMRAN scheme clearly improves the speed tracking performance with or without disturbances/uncertainties, thereby improving vehicle safety in terms of minimizing the risk of collisions with nearby vehicles.

4.2. Lateral path-tracking control

DLC maneuvers at constant longitudinal velocities were performed to assess the lateral path-tracking ability of the proposed Stanley-EMRAN controller. The reference trajectory in terms of lateral displacement (y_r) and yaw angle (ψ_r) are expressed as in Xia et al. (2016):

$$\begin{aligned}
 y_r &= \frac{4.05}{2}(1 + \tanh(a)) - \frac{5.7}{2}(1 + \tanh(b)) \\
 \psi_r &= \arctan \left(4.05 \left(\frac{1}{\cosh(a)} \right)^2 \left(\frac{1.2}{25} \right) \right. \\
 &\quad \left. - 5.7 \left(\frac{1}{\cosh(b)} \right)^2 \left(\frac{1.2}{21.95} \right) \right)
 \end{aligned} \quad (25)$$

Table 6

Error characteristics of the lateral controllers without disturbances and uncertainties.

Controller	$e_{y_{rms}}$ (m)	$e_{y_{max}}$ (m)	$e_{\psi_{rms}}$ (rad)	$e_{\psi_{max}}$ (rad)
Stanley	0.0683	0.2031	0.0160	0.0447
Stanley-EMRAN	0.0218	0.0462	0.0089	0.0256
Coupled-EMRAN	0.0274	0.0677	0.0083	0.0267

where $a = \frac{2.4(v_x t - 27.19)}{25} - 1.2$ and $b = \frac{2.4(v_x t - 56.46)}{21.95} - 1.2$. The results of the proposed lateral controller with and without disturbances/uncertainties are presented below for both slow and high speed scenarios.

4.2.1. Slow speed and no disturbance

This case examines the performance and online learning capability of the Stanley-EMRAN controller through a slow DLC maneuver at a constant velocity of 10 m/s, without any external disturbances and uncertainties. Fig. 9 shows the lateral position (y) and the corresponding lateral error (e_y) for both the conventional and EMRAN-aided Stanley controllers. Table 6 presents a quantitative comparison of their performances in terms of the RMS and maximum values of the lateral and heading angle errors. The proposed neuro-controller improves the trajectory tracking by reducing the peak lateral offset ($e_{y_{max}}$) by 77.25% during the DLC maneuver. The responsiveness of the vehicle to lane change has also improved compared to that of the conventional Stanley approach. Additionally, the vehicle's ability to follow the reference heading is achieved rapidly with a smaller maximum error (Fig. 10), which decreased by 42.72%. It is evident from the results that the proposed Stanley-EMRAN scheme improves the trajectory following ability of the vehicle, thereby reducing the chances of collision with another vehicle during sudden lane changes.

From Table 6, it is seen that the integrated Coupled-EMRAN controller is capable of maintaining the desired performance even under the effects of the longitudinal dynamics of the vehicle and the results of the coupled state are also comparable to that of the Stanley-EMRAN lateral control method. Minor deviations in errors are observed with the coupled architecture because of the small variations in the vehicle's velocity during the lane changes.

The yaw rate and the lateral velocity of the vehicle during the DLC maneuver are shown in Fig. 11. Both the conventional and EMRAN-aided Stanley controllers maintain the yaw rate and the lateral velocity within a reasonable range, thereby not degrading the stability of the AV. Fig. 12 shows the control input, i.e. the steering angle (δ_f) of the controllers for the DLC maneuver. The figure indicates that, at slow-speed, the vehicle is able to track the reference paths accurately, with reasonable control inputs.

The neuron growth history is shown in Fig. 13. It may be noted that EMRAN adds neurons when the AV is undergoing the lane changes and prunes when it is in steady-state operation. Neuron peaks are observed during $7s < t < 9s$ so as to minimize the large lateral and heading errors with the Stanley controller during that time period.

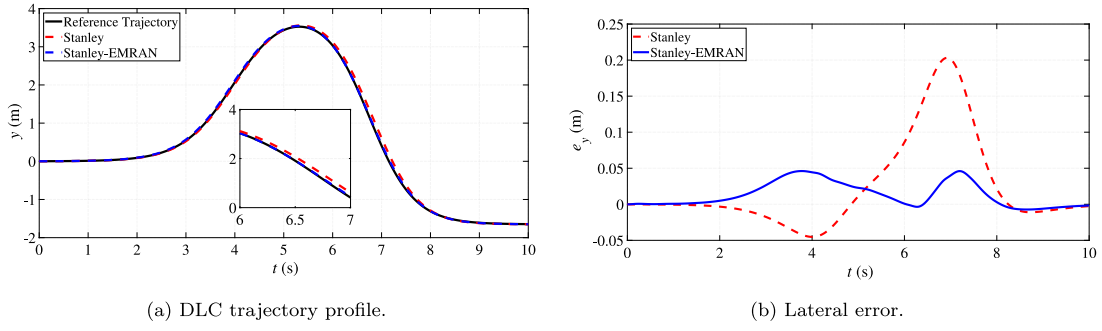


Fig. 9. Vehicle trajectory and corresponding lateral error without external disturbances and uncertainties.

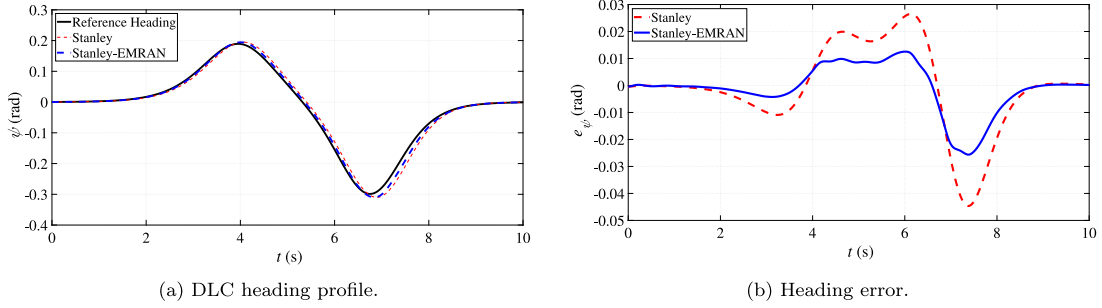


Fig. 10. Vehicle heading and error without external disturbances and uncertainties.

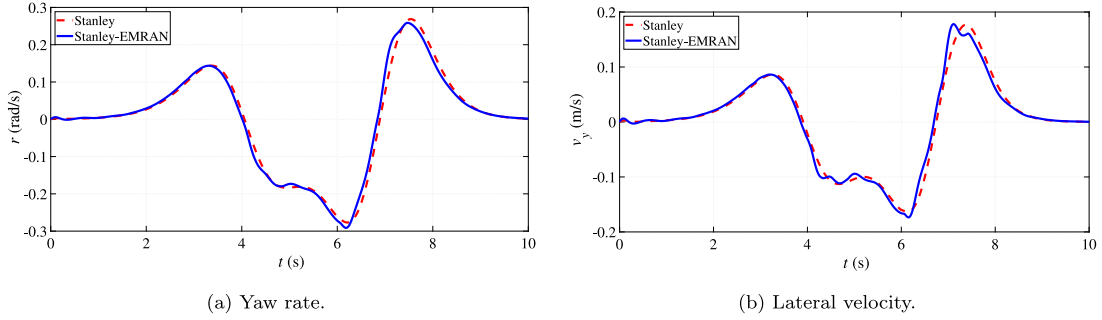


Fig. 11. Lateral dynamics during the slow DLC maneuver without disturbances and uncertainties.

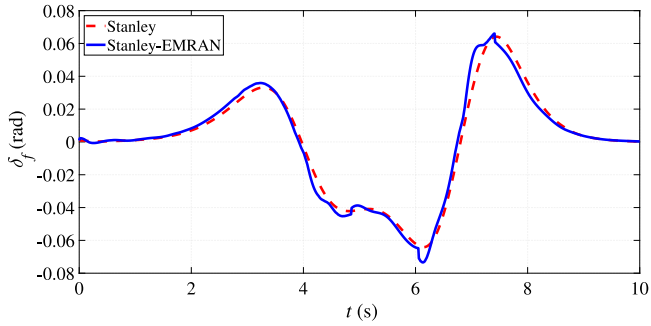


Fig. 12. Steering input for the DLC maneuver.

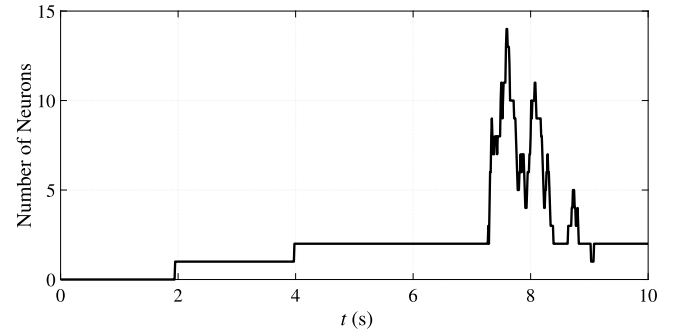


Fig. 13. Lateral EMRAN neuron profile during the DLC maneuver at slow speed and without disturbances.

4.2.2. Slow speed and disturbance

It is crucial to study the effects of external factors while designing lateral controllers for AVs. These factors can arise from wind gusts, external forces, and varying road friction in the form of disturbances and uncertainties and have the potential to degrade the desired path-tracking performance of the vehicle.

The effectiveness of the Stanley-EMRAN controller has been verified by applying a constant external lateral force of 1500 N on the vehicle. The tracking errors are shown in Fig. 14. Initially, EMRAN starts by learning to approximate the nonlinearities due to the disturbance. During this period, the lateral errors (e_y) are greater compared to

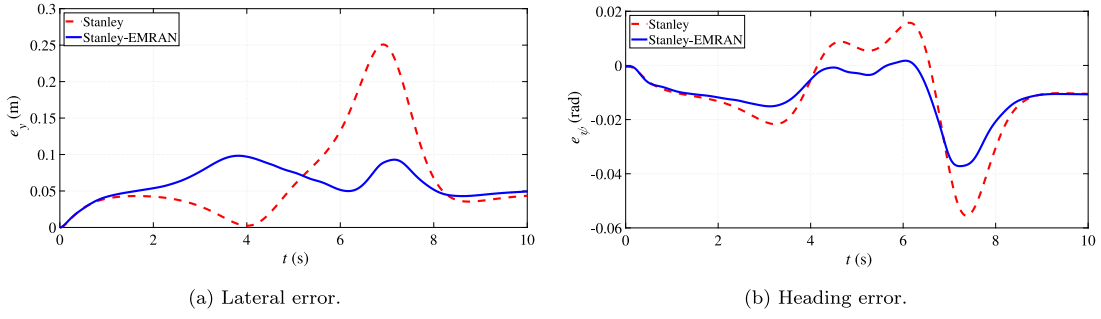


Fig. 14. Tracking errors at slow speed and in presence of a constant external force of 1500 N.

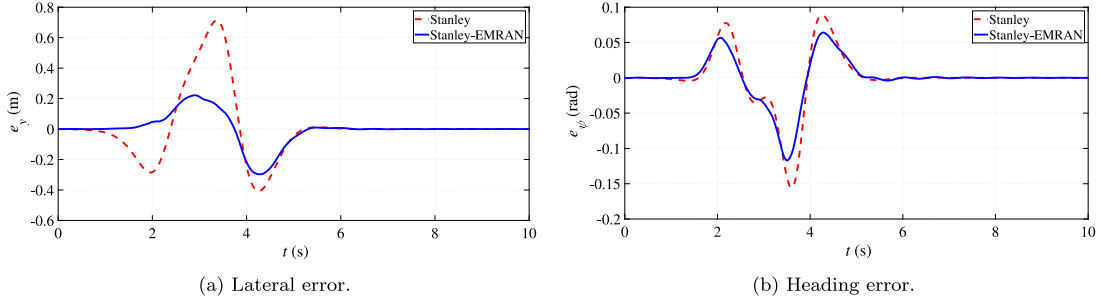


Fig. 15. Tracking errors at high speed and without disturbances.

Table 7

Path-tracking errors at slow speed and with external disturbance.

Controller	$e_{y_{rms}}$ (m)	$e_{y_{max}}$ (m)	$e_{\psi_{rms}}$ (rad)	$e_{\psi_{max}}$ (rad)
Stanley	0.0943	0.2508	0.0197	0.0555
Stanley-EMRAN	0.0647	0.0983	0.0145	0.0372
Coupled-EMRAN	0.0652	0.1188	0.0144	0.0373

Table 8

Tracking errors at high speed and without disturbances.

Controller	$e_{y_{rms}}$ (m)	$e_{y_{max}}$ (m)	$e_{\psi_{rms}}$ (rad)	$e_{\psi_{max}}$ (rad)
Stanley	0.2163	0.7131	0.0391	0.1558
Stanley-EMRAN	0.0981	0.2968	0.0305	0.1171
Coupled-EMRAN	0.0943	0.3041	0.0304	0.1168

Table 9

Tracking errors at high speed and with disturbances.

Controller	$e_{y_{rms}}$ (m)	$e_{y_{max}}$ (m)	$e_{\psi_{rms}}$ (rad)	$e_{\psi_{max}}$ (rad)
Stanley	75.3552	152.1152	1.3771	1.6698
Stanley-EMRAN	1.0765	2.4869	0.3893	0.5753
Coupled-EMRAN	1.0854	2.5523	0.3913	0.5928

Table 10

Range of internal vehicle parameters.

Nominal value	Minimum value	Maximum value
m	0.8 m	1.2 m
I_z	$0.8I_z$	$1.2I_z$
C_f	$0.85C_f$	$1.15C_f$
C_r	$0.85C_r$	$1.15C_r$

Stanley. It is indicative that once the learning and adaptation process is complete ($t > 4$ s), the EMRAN-aided Stanley controller outperforms the conventional Stanley method and significantly reduces both the peaks of lateral and heading errors. Due to the bounded nature of the control input as described in Eq. (3), the steady-state errors cannot be completely eliminated for all the cases. The bounded input ensures that the vehicle can follow the desired response set as closely as possible without losing its yaw stability. The quantitative results of the controllers are presented in Table 7. The robustness of Stanley-EMRAN to external disturbances is apparent, as it decreases $e_{y_{max}}$ by 60.8% and $e_{\psi_{max}}$ by 32.97%. Moreover, the results of the coupled system suggest that both the EMRAN-aided PID and Stanley controllers work coherently in rejecting the effects of the external force.

4.2.3. High speed and no disturbance

Next, we consider a high-speed DLC maneuver at constant velocity of 20 m/s and without disturbances/uncertainties. Note that it becomes more difficult for the controllers to control a fast-moving vehicle because of the shorter response time, leading to greater tracking errors.

It is observed from Table 8 and Fig. 15 that, compared to the conventional Stanley, the proposed Stanley-EMRAN controller improves the tracking performance by minimizing the overshoots and undershoots, and peak lateral errors. However, not much improvement is

Table 11

RMS and maximum path-tracking errors with parametric uncertainties.

Controller	$e_{y_{rms}}$ (m)	$e_{y_{max}}$ (m)	$e_{\psi_{rms}}$ (rad)	$e_{\psi_{max}}$ (rad)
Stanley	0.0750	0.2199	0.0166	0.0461
Stanley-EMRAN	0.0223	0.0554	0.0089	0.0263
Coupled-EMRAN	0.0272	0.0598	0.0085	0.0290

noticed in the heading performance since with a single control input, both the lateral and heading errors cannot be significantly reduced simultaneously (Ji et al., 2018). In this study, we have focused primarily on eliminating the lateral errors during the path-tracking maneuvers. It was found through experiments that mitigating the lateral errors while having reasonable errors in the heading, the vehicle is able to perform the harsh maneuvers without losing its stability. Aggressive heading correction also degrades the passenger comfort and increases the chance of a rollover.

Also, because of varying longitudinal dynamics of the vehicle during the lane changes, an increase in $e_{y_{max}}$ is noticed with the Coupled-EMRAN controller. This increase is also evident in all the test cases discussed earlier.

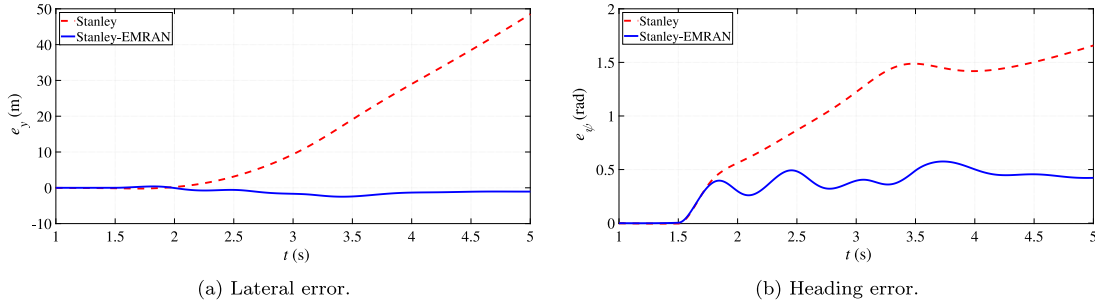


Fig. 16. Tracking errors at high speed and in the presence of wind gust.

Table 12

Quantitative comparison with existing lateral controllers.

Controller	$e_{y_{max}}$ (m)	$e_{y_{rms}}$ (m)	$e_{\psi_{rms}}$ (rad)	$e_{\psi_{max}}$ (rad)
EKF-T2FNN (Taghavifar et al., 2020)	0.0587	0.0685	0.0050	0.0089
ADRC (Taghavifar et al., 2020)	0.2207	0.5593	0.0178	0.0363
Stanley-EMRAN	0.0218	0.0462	0.0089	0.0256

4.2.4. High speed and disturbance

A disturbance tolerance study of the conventional Stanley and EMRAN-aided Stanley controllers is discussed in this case. DLC maneuver at a constant vehicle speed of 20 m/s was simulated in the occurrence of a wind gust ($V_w = 25$ m/s) at $t = 2$ s.

It is seen from Table 9 and Fig. 16 that the conventional Stanley controller has very poor tolerance to such an unexpected disturbance and completely diverges the vehicle from the reference path, leading to large tracking errors in both the lateral position and heading. On the other hand, the EMRAN-aided controllers were able to meet the lane change requirements and prevent the vehicle from veering off course.

Remark: This scenario was implemented to confirm the robustness of the proposed EMRAN-aided controllers against extreme unforeseen situations where other control approaches would fail, and should not be used as a metric for comparison.

4.2.5. Parametric uncertainties and no disturbance

Finally, uncertainties in tire cornering stiffness (C_f and C_r) and load (m and I_z) are considered at a constant speed of 10 m/s. Table 10 presents the minimum and maximum values of these parameters. The RMS and maximum tracking errors with the parametric uncertainties are shown in Table 11. Both the Coupled-EMRAN and Stanley-EMRAN neuro-controllers exhibit the capacity to withstand the perturbations of the internal vehicle parameters and have a better tracking accuracy compared to the conventional Stanley control.

4.2.6. Comparison with other existing methods: Type-2 fuzzy PID and active disturbance rejection schemes

The performance of the proposed EMRAN-aided path-tracking controller is further evaluated against the results of a recently developed type-2 fuzzy PID neural network coupled to an EKF-based neural observer (EKF-T2FNN) (Taghavifar et al., 2020) by using the same set of vehicle parameters and configurations. An active disturbance rejection control (ADRC) with differential flatness (Xia et al., 2016) is also compared with our approach.

Table 12 presents the comparison of the controllers for a DLC maneuver at a constant speed of 10 m/s. The Stanley-EMRAN controller outperforms the EKF-T2FNN method in terms of minimizing the lateral $e_{y_{max}}$ and $e_{y_{rms}}$ errors by 32.5% and 62.8% respectively. As already stated, the primary objective of this work is to reduce the lateral errors in path-tracking without degrading the yaw stability, particularly at high speeds. This constraint leads to greater heading errors than the EKF-T2FNN controller. However, in contrast to the ADRC, the EMRAN-aided lateral controller have a better overall tracking performance.

Based on the above results, it is evident that utilizing the EMRAN neural network as an aid to feedback controllers can significantly improve the cruise control and path-tracking capabilities of an AV. Its ability to learn and adapt online makes the proposed coupled controller indispensable in achieving accurate and reliable tracking response, even in harsh and extreme conditions.

5. Conclusions

In this paper, a novel coupled longitudinal and lateral controller based on the online learning EMRAN neural network was presented for improving the cruise control and path-tracking performances of AVs. A feedback error learning mechanism was employed for learning the inverse dynamics of the vehicle and eliminating the effects of external disturbances and uncertainties. The performance of the controller is compared with conventional PID and Stanley approaches, as well as a fuzzy-based PID method and an active disturbance rejection control system. Simulation results in terms of the RMS and maximum tracking errors confirm the significant enhancements in control performance with the proposed scheme. In addition to providing good robustness, it is demonstrated that the EMRAN-aided controller can adapt itself under extreme situations where other controllers could fail. A self-regulated scheme integrated with the fast online learning algorithm also improves the generalization ability of the controller and significantly reduces the computational burden on the AV. There are, however, certain aspects related to the presented work that can be further investigated such as adaptive fault tolerance, actuator delays, and study of tire-road friction estimation. Future studies will examine these aspects of AV control and also include benchmarking with other neural network-based controllers through actual hardware implementation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work has been funded by the Wipro-IISc Research Innovation Network (WIRIN), India. We would also like to thank the reviewers for their valuable comments which improved the quality of the paper.

References

- Amer, N.H., Hudha, K., Zamzuri, H., Aparow, V.R., Kadir, Z.A., Abidin, A.F.Z., 2019. Hardware-in-the-loop simulation of trajectory following control for a light armoured vehicle optimised with particle swarm optimisation. *Int. J. Heavy Veh. Syst.* 26 (5), 663–691.
- Amer, N.H., Zamzuri, H., Hudha, K., Kadir, Z.A., 2017. Modelling and control strategies in path tracking control for autonomous ground vehicles: a review of state of the art and challenges. *J. Intell. Robot. Syst.* 86 (2), 225–254.
- Arifin, B., Suprpto, B.Y., Prasetyowati, S.A.D., Nawawi, Z., 2019. The lateral control of autonomous vehicles: A review. In: 2019 International Conference on Electrical Engineering and Computer Science (ICECOS). pp. 277–282. <http://dx.doi.org/10.1109/ICECOS47637.2019.8984590>.
- Attia, R., Orjuela, R., Basset, M., 2014. Combined longitudinal and lateral control for automated vehicle guidance. *Veh. Syst. Dyn.* 52 (2), 261–279.
- Azziahdam, E.T., Alankus, O.B., 2021. Longitudinal control of autonomous vehicles consisting of power-train with non-linear characteristics. *IEEE Trans. Intell. Veh.* 1. <http://dx.doi.org/10.1109/TIV.2021.3115265>.
- Chen, C., Li, M., Sui, J., Wei, K., Pei, Q., 2016. A genetic algorithm-optimized fuzzy logic controller to avoid rear-end collisions. *J. Adv. Transp.* 50 (8), 1735–1753.
- Devineau, G., Polack, P., Alché, F., Moutarde, F., 2018. Coupled longitudinal and lateral control of a vehicle using deep learning. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC). pp. 642–649. <http://dx.doi.org/10.1109/ITSC.2018.8570020>.
- Dingmen, E., Altunel, T., 2018. An emergency braking controller based on extremum seeking with experimental implementation. *Int. J. Dyn. Control* 6 (1), 270–283.
- Eskandarian, A., 2012. *Handbook of Intelligent Vehicles*, Vol. 2. Springer.
- Gomi, H., Kawato, M., 1990. Learning control for a closed loop system using feedback-error-learning. In: 29th IEEE Conference on Decision and Control. IEEE, pp. 3289–3294.
- Hoffmann, G.M., Tomlin, C.J., Montemerlo, M., Thrun, S., 2007. Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing. In: 2007 American Control Conference. pp. 2296–2301. <http://dx.doi.org/10.1109/ACC.2007.4282788>.
- Huang, X., Zhang, H., Zhang, G., Wang, J., 2014. Robust weighted gain-scheduling hoo vehicle lateral motion control with considerations of steering system backlash-type hysteresis. *IEEE Trans. Control Syst. Technol.* 22 (5), 1740–1753.
- Ji, X., He, X., Lv, C., Liu, Y., Wu, J., 2018. Adaptive-neural-network-based robust lateral motion control for autonomous vehicle at driving limits. *Control Eng. Pract.* 76, 41–53.
- Kadirkamanathan, V., Niranjan, M., 1993. A function estimation approach to sequential learning with neural networks. *Neural Comput.* 5 (6), 954–975. <http://dx.doi.org/10.1162/neco.1993.5.6.954>.
- Khodayari, A., Ghaffari, A., Ameli, S., Flahatgar, J., 2010. A historical review on lateral and longitudinal control of autonomous vehicle motions. In: 2010 International Conference on Mechanical and Electrical Technology. pp. 421–429. <http://dx.doi.org/10.1109/ICMET.2010.5598396>.
- Kuutti, S., Bowden, R., Jin, Y., Barber, P., Fallah, S., 2021. A survey of deep learning applications to autonomous vehicle control. *IEEE Trans. Intell. Transp. Syst.* 22 (2), 712–733. <http://dx.doi.org/10.1109/TITS.2019.2962338>.
- Li, Y., Sundararajan, N., Saratchandran, P., 2000. Analysis of minimal radial basis function network algorithm for real-time identification of nonlinear dynamic systems. *IEE Proc.-Control Theory Appl.* 147 (4), 476–484.
- Lie, G., Zejian, R., Pingshu, G., Jing, C., 2014. Advanced emergency braking controller design for pedestrian protection oriented automotive collision avoidance system. *Sci. World J.* 2014.
- Liu, K., Gong, J., Kurt, A., Chen, H., Ozguner, U., 2018. Dynamic modeling and control of high-speed automated vehicles for lane change maneuver. *IEEE Trans. Intell. Veh.* 3 (3), 329–339. <http://dx.doi.org/10.1109/TIV.2018.2843177>.
- Lu, Y., Sundararajan, N., Saratchandran, P., 1997. A sequential learning scheme for function approximation using minimal radial basis function neural networks. *Neural Comput.* 9 (2), 461–478. <http://dx.doi.org/10.1162/neco.1997.9.2.461>.
- Lu, Y., Sundararajan, N., Saratchandran, P., 1998. Performance evaluation of a sequential minimal radial basis function (RBF) neural network learning algorithm. *IEEE Trans. Neural Netw.* 9 (2), 308–318. <http://dx.doi.org/10.1109/72.661125>.
- Norouzi, A., Masoumi, M., Barari, A., Sani, S.F., 2019. Lateral control of an autonomous vehicle using integrated backstepping and sliding mode controller. *Proc. Inst. Mech. Eng. K* 233 (1), 141–151.
- Ntousakis, I.A., Nikolos, I.K., Papageorgiou, M., 2015. On microscopic modelling of adaptive cruise control systems. *Transp. Res. Proc.* 6, 111–127.
- Paden, B., Čáp, M., Yong, S.Z., Yershov, D., Frazzoli, E., 2016. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Trans. Intell. Veh.* 1 (1), 33–55. <http://dx.doi.org/10.1109/TIV.2016.2578706>.
- Piao, C., Liu, X., Lu, C., 2019. Lateral control using parameter self-tuning LQR on autonomous vehicle. In: 2019 International Conference on Intelligent Computing, Automation and Systems (ICICAS). pp. 913–917.
- Platt, J., 1991. A resource-allocating network for function interpolation. *Neural Comput.* 3 (2), 213–225.
- Plessen, M.G., Bernardini, D., Esen, H., Bemporad, A., 2018. Spatial-based predictive control and geometric corridor planning for adaptive cruise control coupled with obstacle avoidance. *IEEE Trans. Control Syst. Technol.* 26 (1), 38–50. <http://dx.doi.org/10.1109/TCST.2017.2664722>.
- Raffin, A., Taragna, M., Giorrelli, M., 2017. Adaptive longitudinal control of an autonomous vehicle with an approximate knowledge of its parameters. In: 2017 11th International Workshop on Robot Motion and Control (RoMoCo). IEEE, pp. 1–6.
- Rajamani, R., 2011. *Vehicle Dynamics and Control*. Springer Science & Business Media.
- Salaani, M.K., 2007. Analytical tire forces and moments model with validated data. *SAE Trans.* 735–763.
- Savitha, R., Suresh, S., Sundararajan, N., 2012. A meta-cognitive learning algorithm for a fully complex-valued relaxation network. *Neural Netw.* 32, 209–218.
- Schwartz, W., Alonso-Mora, J., Rus, D., 2018. Planning and decision-making for autonomous vehicles. *Annu. Rev. Control Robot. Auton. Syst.*
- Shakouri, P., Ordys, A., 2014. Nonlinear model predictive control approach in design of adaptive cruise control with automated switching to cruise control. *Control Eng. Pract.* 26, 160–177.
- Suresh, S., Dong, K., Kim, H.J., 2010. A sequential learning algorithm for self-adaptive resource allocation network classifier. *Neurocomputing* 73 (16–18), 3012–3019.
- Suresh, S., Huang, H., Kim, H.J., 2014. Hybrid real-coded genetic algorithm for data partitioning in multi-round load distribution and scheduling in heterogeneous systems. *Appl. Soft Comput.* 24, 500–510.
- Suzuki, T., Fujimoto, H., 2010. Slip ratio estimation and regenerative brake control without detection of vehicle velocity and acceleration for electric vehicle at urgent brake-turning. In: 2010 11th IEEE International Workshop on Advanced Motion Control (AMC). pp. 273–278. <http://dx.doi.org/10.1109/AMC.2010.5464122>.
- Taghavifar, H., Hu, C., Qin, Y., Wei, C., 2020. EKF-neural network observer based type-2 fuzzy control of autonomous vehicles. *IEEE Trans. Intell. Transp. Syst.* 1–13. <http://dx.doi.org/10.1109/TITS.2020.2985124>.
- Tork, N., Amirkhani, A., Shokouhi, S.B., 2021. An adaptive modified neural lateral-longitudinal control system for path following of autonomous vehicles. *Eng. Sci. Technol. Int. J.* 24 (1), 126–137. <http://dx.doi.org/10.1016/j.jestech.2020.12.004>, URL: <https://www.sciencedirect.com/science/article/pii/S2215098620342658>.
- Wang, S., Hui, Y., Sun, X., Shi, D., 2019. Neural network sliding mode control of intelligent vehicle longitudinal dynamics. *IEEE Access* 7, 162333–162342. <http://dx.doi.org/10.1109/ACCESS.2019.2949992>.
- Xia, Y., Pu, F., Li, S., Gao, Y., 2016. Lateral path tracking control of autonomous land vehicle based on adrc and differential flatness. *IEEE Trans. Ind. Electron.* 63 (5), 3091–3099. <http://dx.doi.org/10.1109/TIE.2016.2531021>.