Neural Network Based Modelling and Steering Control of an Intelligent Vehicle Under Dynamic Sensitive Conditions

K Zhang\textsuperscript{1,*}, S J Wang\textsuperscript{2}, L Ji\textsuperscript{3} and C Wang\textsuperscript{1}

\textsuperscript{1}E/E department, Brilliance Automobile Engineering Research Institute, shenyang 110141, China
\textsuperscript{2}School of Mechanical Engineering, Shenyang University of Technology, shenyang, 110023, China
\textsuperscript{3}School of Mechatronics Engineering, Shenyang Aerospace University, shenyang, 110136, China

*hit_zhangkun@126.com

Abstract. A neural network based modelling and steering control method of an intelligent vehicle under dynamic sensitive conditions is proposed in this paper. Four radius function neural networks are used to compensate the modelling error of a traditional dynamic vehicle steering model. The designed control law is consisted of four terms: a neural network term, a proportion term, an integral term and a robust term. Stability is analysed through Lyapunov function and the training law is generated at the same time. Test results show that the designed steering controller works quite well and is of great value in path tracking of an intelligent vehicle when the vehicle dynamics getting significant.

1. Introduction
With the development of the automobile, the problems of traffic safety, traffic congestion and traffic pollution have become more and more significant. Intelligent Transportation System (ITS) has proven to be an effective solution of these problems. As the core component of ITS, research on the connected intelligent vehicle has been highly valued and widely concerned by countries all over the world, including all major automobile manufacturers, automobile component suppliers, internet enterprises, universities and research institutes, which was also highly fuelled by the development in computer science and artificial intelligence recent years.

The DARPA Grand Challenge started in 2004 and DARPA Urban Challenge in 2007 is a major milestone of modern intelligent driving development. The raised abstracted system plan, layered software architecture by those successful entries are still well used by many most popular intelligent driving prototypes all around the world. To limit the scope of this paper, the perception, localization, route planning, behavioural decision and motion planner are omitted, only the motion controller is studied, whose role is to track the reference path or trajectory that was provided by the motion planner, with insurance of vehicle stability and a small tracking error.

There have been plenty of studies regrading motion control of intelligent driving vehicles. Some of these control methods exploit the geometric relationship between vehicle and the reference path, which are simple, robust, and can be easily implemented but can still achieve accurate path tracking in a broad set of driving scenarios\cite{1, 2}. As the vehicle kinematics become significant, simplifying the vehicle to a kinematic bicycle model is a common approximation. Some well-known control methods
are implemented on the kinematic model, and are widely used in platooning and motion control of vehicles with trailers[3-5]. When the vehicle speed, acceleration and side slip of the vehicle become significant, control methods based on the dynamic vehicle steering model and more advanced control methods, like Model Predictive Control[6], Sliding Model Control[7] and fuzzy control[8] are widely used, which are more complicated but can achieve desirable performance in these dynamic sensitive scenarios.

The commonly used dynamic model is built under some assumptions, which may fail in dynamic sensitive steering scenarios. Thus, a neural network compensated dynamic vehicle steering model is built, and the corresponding control method is proposed in this paper.

2. Dynamic vehicle steering model

The most commonly used dynamic vehicle steering model is called bicycle model, which in fact is two wheels that were rigidly linked. The same dynamic vehicle steering model built in [9] is used in this paper, detailed modelling process is not listed here due to limited space. The model in state space is

$$A\dot{q} + Bq + Cq - H = \delta$$

Where $q = \begin{bmatrix} e_{cg} & \theta_e \end{bmatrix}^T$, $A = \begin{bmatrix} \frac{m}{c_f} & 0 \\ 0 & \frac{l_c}{l_f c_f} \end{bmatrix}$, $B = \begin{bmatrix} c_f + c_r & 0 \\ \frac{l_f c_f}{c_f v_s} & \frac{l_f c_f}{c_f v_s} \end{bmatrix}$, $C = \begin{bmatrix} 0 & 0 \\ -\frac{l_f}{l_f c_f} & -\frac{l_f}{l_f c_f} \end{bmatrix}$, $H = \begin{bmatrix} \frac{m}{c_f} \frac{-(l_f c_f - l_r c_r)}{mv_s} \omega(s) \\ \frac{l_f^2 c_f + l_r^2 c_r}{l_f c_f v_s} \omega(s) - \frac{l_f}{l_f c_f} \omega(s) \end{bmatrix}$.

As is shown above, $e_{cg}$ is the cross track error, $\theta_e$ is the heading error, $m$ is the mass of the vehicle, $\omega$ is the yaw rate, $l_c$ is the inertial moment, $l_f$ and $l_r$ are the length of front and rear half axis, $c_f$ and $c_r$ is the corner stiffness of the front and rear wheels.

This model is built under some assumptions, so it is supposed to fail under some dynamic sensitive conditions such as when the vehicle speed and acceleration getting larger, the lateral slip getting significant. Thus, neural network is used to compensate these modelling errors in this paper.

3. Control algorithm design

3.1. Control law design

During the modeling process of the dynamic vehicle steering model, there exist many uncertainties and assumptions, thus, four radial basis function network are used to model $A$, $B$, $C$ and $H$:

$$\begin{cases} A = A_{NN} + E_A \\ B = B_{NN} + E_B \\ C = C_{NN} + E_C \\ H = H_{NN} + E_H \end{cases}$$

Where $A_{NN}$, $B_{NN}$, $C_{NN}$ and $H_{NN}$ are the output of the corresponding networks, $E_A$, $E_B$, $E_C$...
and $E_N$ are the corresponding network modelling errors.

Thus, $A\dot{q}_t + B\dot{q}_r + Cq_r - H$ can be represented as

$$A\dot{q}_t + B\dot{q}_r + Cq_r - H = A_{NN}\dot{q}_t + B_{NN}\dot{q}_r + C_{NN}q_r - H_{NN} + E$$

Where, $\omega_A$, $\omega_B$, $\omega_C$ and $\omega_H$ are the corresponding network weight vector, $h_A$, $h_B$, $h_C$ and $h_H$ are the corresponding network hidden layer outputs, $E$ is the error term, which is represented as $E = E_A\dot{q}_t + E_B\dot{q}_r + E_Cq_r - E_H$, $\dot{q}_t$ and $\dot{q}_r$ are the inputs with error terms, which is represented as

$$\dot{q}_r = r + \dot{q} = \dot{e} + \Lambda e + \dot{q}_d = \dot{q}_d + \Lambda e$$

Where, $\Lambda$ is a diagonal matrix, designed as $\Lambda = \text{eye}(5)$.

Represent the estimated value of $A_{NN}$, $B_{NN}$, $C_{NN}$ and $H_{NN}$ as $\hat{A}_{NN} = \hat{\omega}_A^T h_A$, $\hat{B}_{NN} = \hat{\omega}_B^T h_B$, $\hat{C}_{NN} = \hat{\omega}_C^T h_C$ and $\hat{H}_{NN} = \hat{\omega}_H^T h_H$. Where, $\hat{\omega}_A^T$, $\hat{\omega}_B^T$, $\hat{\omega}_C^T$ and $\hat{\omega}_H^T$ are the estimated weight vector of $A_{NN}$, $B_{NN}$, $C_{NN}$ and $H_{NN}$.

Thus, the original dynamic model can be represented as

$$\delta = \hat{A}\dot{q} + B\dot{q} + Cq - H = \hat{A}\dot{q}_t + B\dot{q}_r + Cq_r - H - \dot{A}r - Br$$

$$= \omega_A^T h_A\dot{q}_r + \omega_B^T h_B\dot{q}_r + \omega_C^T h_C q_r - \omega_H^T h_H - \dot{A}r - Br - C\int_0^t r \, dt + E$$

The steering angle is designed as

$$\delta = \delta_m + K_p r + K_i \int_0^t r \, dt + \delta_r$$

$\delta_m$ is the estimated steering angle of the network, which is $\delta_m = \hat{A}_{NN}\dot{q} + \hat{B}_{NN}\dot{q}_r + \hat{C}_{NN}q_r - \hat{H}_{NN}$. $K_p = \text{eye}(100)$ is the proportion term; $K_i = \text{eye}(100)$ is the integral term, $\delta_r$ is the robust term considering the network modelling error, designed as $\delta_r = K_r \text{sgn}(r)$, $K_r = \text{diag}[k_m], k_m \geq |E_t|$.

### 3.2. Neural network design

Substitute the estimated steering angle and the robust term into the control law (6), yields

$$\delta = \omega_A^T h_A\hat{\dot{q}}_r + \omega_B^T h_B\hat{\dot{q}}_r + \omega_C^T h_C q_r - \omega_H^T h_H + K_p r + K_i \int_0^t r \, dt + K_r \text{sgn}(r)$$

Substitute this control law (7) into the dynamic model (5), yields

$$\dot{A}r + Br + C\int_0^t r \, dt + K_p r + K_i \int_0^t r \, dt + K_r \text{sgn}(r)$$

$$= (\omega_A^T - \dot{\omega}_A^T) h_A\dot{q}_r + (\omega_B^T - \dot{\omega}_B^T) h_B\dot{q}_r + (\omega_C^T - \dot{\omega}_C^T) h_C q_r - (\omega_H^T - \dot{\omega}_H^T) h_H + E$$

$$= \dot{\omega}_A^T h_A\dot{q}_r + \dot{\omega}_B^T h_B\dot{q}_r + \dot{\omega}_C^T h_C q_r - \dot{\omega}_H^T h_H + E$$

Training law is designed as $\dot{\omega}_A = \Gamma_A h_A\dot{q}_r$, $\dot{\omega}_B = \Gamma_B h_B\dot{q}_r$, $\dot{\omega}_C = \Gamma_C h_C q_r$, and $\dot{\omega}_H = \Gamma_H h_H r$, where $\Gamma_A$, $\Gamma_B$, $\Gamma_C$ and $\Gamma_H$ are symmetric definite matrix, $h_A$, $h_B$, $h_C$ and $h_H$ are the corresponding network hidden layer outputs, and $k$ is number of inputs.
3.3. Stability analysis

Design Lyapunov function as

\[
V = \frac{1}{2} r^T Ar + \frac{1}{2} \left( \int_0^t r \, dt \right)^T K \int_0^t r \, dt + \frac{1}{2} \left( \int_0^t r \, dt \right)^T C \int_0^t r \, dt + \frac{1}{2} \sum_{k=1}^n \omega^T_{ik} \Gamma^{-1}_{ik} \omega_{ik} \\
+ \frac{1}{2} \sum_{k=1}^n \omega^T_{bk} \Gamma^{-1}_{bk} \omega_{bk} + \frac{1}{2} \sum_{k=1}^n \omega^T_{ck} \Gamma^{-1}_{ck} \omega_{ck} - \frac{1}{2} \sum_{k=1}^n \omega^T_{hk} \Gamma^{-1}_{hk} \omega_{hk}
\]

(9)

The derivation of Lyapunov function (9) is

\[
\dot{V} = r^T \left( A \dot{r} + \frac{1}{2} \dot{A} r + K \int_0^t r \, dt + C \int_0^t r \, dt \right) + \sum_{k=1}^n \omega^T_{ik} \Gamma^{-1}_{ik} \dot{\omega}_{ik} \\
+ \sum_{k=1}^n \omega^T_{bk} \Gamma^{-1}_{bk} \dot{\omega}_{bk} + \sum_{k=1}^n \omega^T_{ck} \Gamma^{-1}_{ck} \dot{\omega}_{ck} - \sum_{k=1}^n \omega^T_{hk} \Gamma^{-1}_{hk} \dot{\omega}_{hk}
\]

(10)

Considering the symmetric of \( \dot{A} - 2B \), yields

\[
\dot{V} = r^T \left( A \dot{r} + B r + C \int_0^t r \, dt + K \int_0^t r \, dt \right) + \sum_{k=1}^n \omega^T_{ik} \Gamma^{-1}_{ik} \dot{\omega}_{ik} \\
+ \sum_{k=1}^n \omega^T_{bk} \Gamma^{-1}_{bk} \dot{\omega}_{bk} + \sum_{k=1}^n \omega^T_{ck} \Gamma^{-1}_{ck} \dot{\omega}_{ck} - \sum_{k=1}^n \omega^T_{hk} \Gamma^{-1}_{hk} \dot{\omega}_{hk}
\]

(11)

Substitute (8) into (11), yields

\[
\dot{V} = -r^T K_p r + r^T E - r^T K_s \operatorname{sgn}(r) + r^T \omega^T_i \dot{h}_i \dot{q}_i + r^T \omega^T_b \dot{h}_b \dot{q}_b + r^T \omega^T_c \dot{h}_c \dot{q}_c - r^T \omega^T_h \dot{h}_h \\
+ \sum_{k=1}^n \omega^T_{ik} \Gamma^{-1}_{ik} \dot{\omega}_{ik} + \sum_{k=1}^n \omega^T_{bk} \Gamma^{-1}_{bk} \dot{\omega}_{bk} + \sum_{k=1}^n \omega^T_{ck} \Gamma^{-1}_{ck} \dot{\omega}_{ck} - \sum_{k=1}^n \omega^T_{hk} \Gamma^{-1}_{hk} \dot{\omega}_{hk}
\]

(12)

Let \( k_{\alpha \beta} \geq |E| \), it can induced that

\[
r^T E - r^T K_s \operatorname{sgn}(r) = r^T (E - K_s \operatorname{sgn}(r)) \leq 0
\]

(13)

Substitute (13) into (12), yields \( \dot{V} = -r^T K_p r + r^T (E - K_s \operatorname{sgn}(r)) \leq 0 \)

So this designed system is stable.

4. Simulation and test results

4.1. The Test Vehicle

The test vehicle is developed by brilliance automobile research institute, as is shown in figure 1. The software is mainly hierarchically structured into 4 layers: sensor interface layer, perception & localization layer, decision-making layer and vehicle interface layer, as is shown in figure 2. The decision-making layer is further abstracted into mission plan, behavioral decision, motion plan and motion control sub-layers.
Figure 1. Test vehicle

Figure 2. Test vehicle software architecture

Sensor interface layer is the hardware driver of different kinds of sensors, including laser scanner, camera, radar, global navigation satellite system (GNSS) and inertial navigation system (INS), which can provide the raw sensor data.

Perception & localization layer is responsible for the process of the raw sensor data and prior knowledges like high resolution map data, which can provide the observation of the surroundings and the movement state estimation of the host vehicle.

Mission planner is responsible for the producing of the global route that is a sequence of waypoints across the road network according to the prior map data and user specified information. The behavioral decision layer is responsible for selecting a specific and appropriate behavioral based on the perception information and ego vehicle movement information. The selected driving behavioral is then translated into a dynamically feasible, safe and comfort path or trajectory, which is accomplished by the motion plan module.

The planned local reference path or trajectory is then tracked by the motion planner, and the corresponding control signal is sent to vehicle interface through CAN bus.

4.2. Test Result

The steering controller proposed in this paper is implemented as a part of the motion controller. The output front road wheel steer angle is transformed to the steering wheel angle according a pre-calibrated steer ratio table. During the test, the vehicle speed is separately controlled by a self-adaptive longitudinal motion controller, which is out the scope of this paper.

The test was conducted on a field test road of brilliance automobile research institute. The reference path and the footprint of the test vehicle is illustrated in figure 3, and a more clear and intuitive cross tracking error is shown in figure 4.

Figure 3. field test reference path and vehicle path

Figure 4. field test lateral tracking error

As is shown in figure 3 and figure 4, the designed steering controller tracked the reference path quite well, with a maximum lateral tracking error of -0.3m, this was acceptable but still some improvement is needed in developing of level 3+ intelligent vehicles.

Double Lane Change (DLC) tests under 10m/s, 15m/s, 20m/s and 25m/s are conducted to test the dynamic sensitive scenario tracking effect of the designed controller, the reference path and the footprint of the test vehicle is illustrated in figure 5, and the cross tracking error is shown in figure 6.
Figure 5. DLC reference path and vehicle path

The maximum lateral tracking error of DLC test is 0.27m, prove the effectiveness of the designed controller under dynamic sensitive conditions.

5. Conclusion
To handle dynamic sensitive steering scenarios, a neural network based modelling and steering control method of an intelligent vehicle is proposed in this paper. The modelling error of a traditional dynamic vehicle steering model is compensated by four radius function neural networks. A corresponding four term control law is designed, and the stability is analysed through Lyapunov function. Test results show that the designed steering controller works quite well especially when the vehicle dynamics getting significant. A coupled lateral and longitudinal motion controller is expected to achieve better results.

Acknowledgments
This work is supported by the China National key research and development plan key special project of intergovernmental international science and technology innovation cooperation development project “automotive passive and active collaborative protection technology for pedestrian safety” (No. 2018YFE0192900).

References
[1] Sebastian T, Montemerlo M and Palatucci M 2009 Stanley: The Robot that Won the DARPA Grand Challenge Journal of Field Robotics vol 23(9) pp 661-92.
[2] Chirs U, Joshua A, Drew B, et al 2008 Autonomous driving in urban environments : Boss and the urban challenge Journal of Field Robotics vol 25(8) pp 425-66.
[3] Luca A D, Oriolo G and Samson C 1998 Feedback control of a nonholonomic car-like robot In Robot Motion Planning and Control pp 171–249.
[4] Samson C 1993 Mobile robot control Part 2 : control of chained systems and application to path following and time-varying point-stabilization of wheeled vehicles HAL - INRIA.
[5] Campbell S F 2007 Steering control of an autonomous ground vehicle with application to the DARPA Urban Challenge Massachusetts Institute of Technology.
[6] Falcone P, Borrelli F, Asgari J, Tseng H E and Hrovat D 2007 Predictive active steering control for autonomous vehicle systems Transactions on Control Systems Technology vol 15 pp 566–80.
[7] Guo J, Hu P, Li L, et al 2012 Design of Automatic Steering Controller for Trajectory Tracking of Unmanned Vehicles Using Genetic Algorithms IEEE Transactions on Vehicular Technology vol 61(7) pp 2913-24.
[8] Nguyen A T, Sentouh C and Popiele J C 2017 Fuzzy Steering Control for Autonomous Vehicles under Actuator Saturation: Design and Experiments Journal of the Franklin Institute S0016003217306130.
[9] Zhang K, Cui S M and Wang J F 2014 Intelligent vehicle's path tracking control based on self-adaptive RBF network compensation Control and Decision vol 29(4) pp 627-31.