Intelligent Vehicle Lighting Control System

Funding Source: China FAW (First Auto Work) Group Corporation

Dec. 2020 – Jun. 2022

We proposed an intelligent vehicle lighting control system composed of adaptive driving beam (ADB) and adaptive front lighting system (AFS). The adaptive driving beam system consists of an intelligent front camera and a matrix LED beam controller, which can intelligently adjust the lighting area of driving beam to prevent dazzling other drivers on the road (FIG.1). The adaptive front lighting system consists of the vehicle body-height sensor and some the orientation control motor, which can stabilize the lighting area of low beam according to the dynamic pitch posture (FIG.2A) and adjust the lighting area of steering beam according to the yaw state (FIG.2B). The system is implemented by an embedded automotive electronic control unit (ECU).

FIG.1 In the experiment, the adaptive driving beam system adjusts the lighting area after detecting the vehicle ahead, the red rectangle is the position of the target vehicle, and the blue curve is the cut-off line of lighting area.

(A)

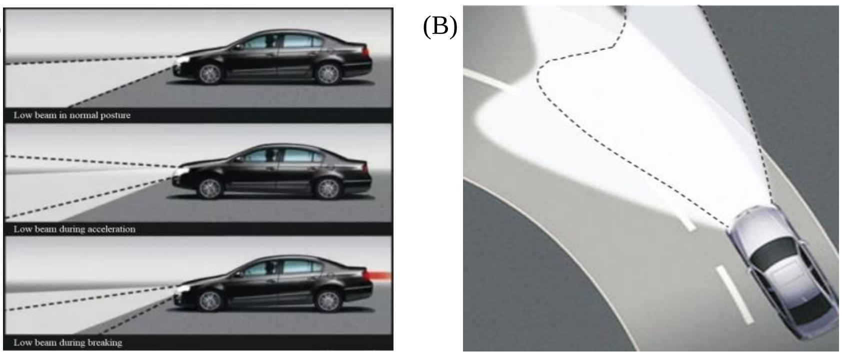

FIG.2 Functions of Adaptive Front Lighting System (A) With the adjustment of adaptive front lighting system, the lighting area does not change with the dynamic pitch posture; The dotted line shows the lighting area of low beam without adjustment. **(B)** The steering beam is adjusted to the yaw direction of the vehicle for a wider range of illumination; The dotted line shows the normal lighting area of steering beam.

My contributions:

I worked with my supervisor to lead the development and testing of application layer software of the project. The main work I am responsible for are shown at the Table.1 below. covering the development of all important algorithms and software for the intelligent front camera, adaptive driving beam control unit and adaptive front lighting control unit. It also includes the simulation test of the main software, the hardware-in-the-loop test of the control unit, the comprehensive test on the real vehicle, and the construction of the related test platform.

Table.1 Main contribution in Intelligent Vehicle Lighting Control System

Developing procedure	Adaptive Driving Beam System (ADB)	
	Intelligent front camera	Vehicle detection model specific at dark environment

	ADB control unit	Vehicle tracking algorithm based on Kalman filter
		Matrix LED driving beam partition control algorithm
	Adaptive Front Lighting System (AFS)	
	Automatic low beam leveling	Road type recognition algorithm based on decision tree
		*The automatic low beam leveling algorithm
	Steering assist lighting	*The estimation algorithm of vehicle yaw state
		*The follow-up steering algorithm
Testing procedure	A model simulation testing platform	
	A HIL simulation testing platform	
	Automated test scripts and frameworks	
	Comprehensive testing and parameter calibration	
Note	The work marked with * was done in collaboration with others	

I trained a vehicle detection model based on transfer learning that dedicated to the adaptive driving beam system. The dataset was created by taking field shots and sifting through web datasets. The videos and images were acquired by driving the test vehicle in different environments such as urban, rural, and highway after 21 o'clock (FIG. 3), and examining the public datasets (e.g., *ExDark*, D^2-*City*) to select the images that include vehicles in low-light condition. The dataset contains 650 minutes of video and 5211 images. After data cleaning, the dataset was annotated according to five categories: on-coming car, same-lane car, crossing car, oversize vehicle and motorcycles. We carried out transfer learning based on the existing model, improved the detection performance by continuously adjusting the model parameters and hyperparameters, reduced the overfitting of the model through data augmentation and drop-off regularization, and finally achieved 88.7% mAP. The model was deployed on the intelligent front camera. The average detection rate is 31ms per frame.

(A)

(B)

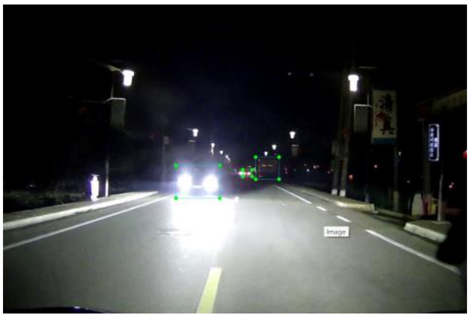

FIG.3 Demonstration of Dataset annotation (A) Highway with a same-lane car. **(B)** Rural road with one on-coming car, one oversize vehicle and one same-lane car.

I designed the model of adaptive driving beam system controller in Simulink and generated the C code in AutoSAR style (FIG.4A). The system has three main blocks: Automatic switching, Target tracking, LED response (FIG.4B). The Automatic switching strategies block judges the current driving scene based on the driver's operation, sensor signal and the vehicle status. I have formulated driving-beam lighting strategies for different scenarios, which decides whether to turn on the driving beam and the lighting mode of the driving beam. It sends corresponding instructions to other blocks. The target tracking block realizes target distance estimation and position tracking. I proposed a monocular camera vehicle distance estimation algorithm that fuses two different distance estimation models using weighted averaging (FIG.4C), and a third-order Kalman filter to track and correct the horizontal position of the input target. The LED responding block adopts a partition control algorithm. After receiving the position of the target vehicle, it can calculate that which LED lamp bead needs to reduce the brightness in real time (FIG.5A). Through a method of linear interpolation, the PWM dimming output is adjusted linearly according to the distance and the type of the target vehicle ahead (FIG.5B). The system I designed can adapt to different car models and different models of matrix LED driving beam. A series of calibratable parameters were reserved in the system design. These parameters should be confirmed during the test before leaving factory and implemented during initialization. Currently the system could support 400 LED beads on each side of the beam at most.

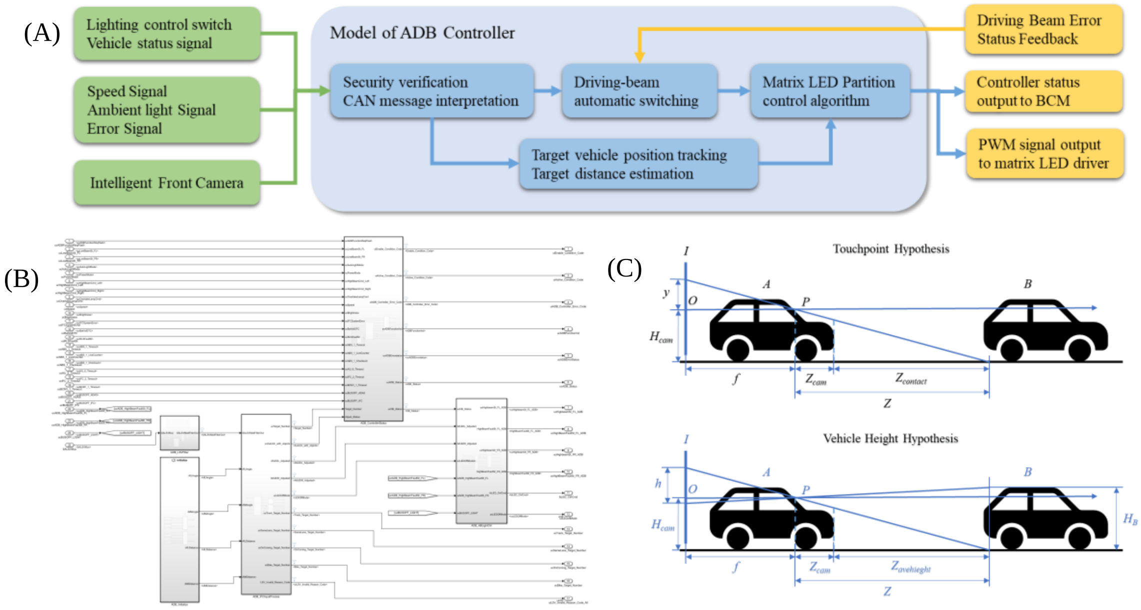

FIG.4 Model of Adaptive Driving Beam System (A) The architecture and functions of ADB controller; **(B)** The control model in Simulink, the blocks from left to right and top to bottom are: input verification, initialization, automatic switching, target tracking and LED response; **(C)** The geometric relationship diagram of vehicle distance estimation based on Touchpoint (top) hypothesis and Vehicle Height (bottom) hypothesis respectively.

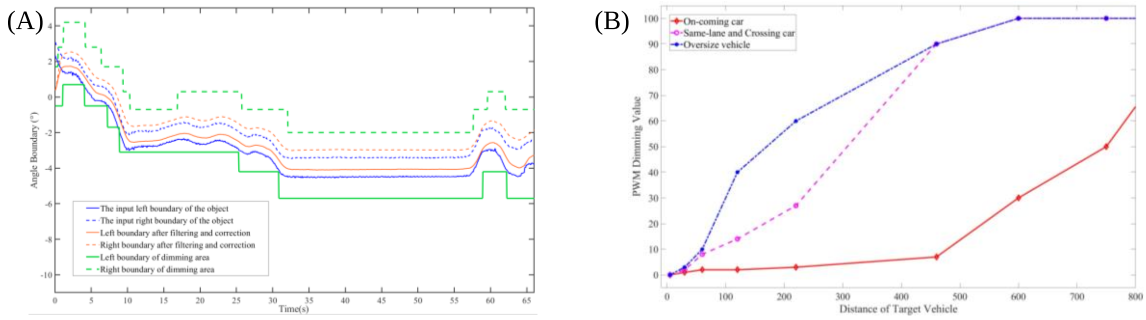

FIG.5 Simulation of LED Response Block (A) The results of the simulation verification of the LED response blocks with a driving beam model contains 20 LED lamp beads on each side; **(B)** The relationship between PWM dimming value with the distance and type of vehicle.

I designed a road type recognition algorithm based on the decision tree model. For the automatic low beam leveling system, the correct and rapid identification of the current road surface type is an important prerequisite for ensuring the stability of the lighting area. Because it is difficult to adapt to roads with different roughness using a single control strategy. In our tests, we found that the leveling function suitable for asphalt road may bring negative optimization effect on some rough gravel road. There are studies using computer vision to recognize road types. But these methods are not effective in dark environment. We proposed a road type recognition algorithm using the data from vehicle height sensor installed on the suspension. The vehicle height sensor is widely used in automobiles at present to detect the distance between the suspension and the vehicle body. The response time of this sensor is within 1ms. We use the discrete-time Hilbert transform to find the instantaneous frequency of the signal, and use the short-time Fourier transform to extract the autocorrelation function of the signal. We combine these signal features with some basic features, such as suspension displacement, vehicle speed and acceleration, as the feature set of the decision tree model. By driving the test vehicles on different types of roads at test plant, we collected enough data to build a decision tree model. After training and optimization, the precision of road surface recognition algorithm for rough roads is higher than 93%, the recognition is lower than 400 μs.

I assisted the junior member of my laboratory to model the adaptive front lighting system in Simulink and generate C code in AutoSAR style using automated tools. In addition to the previously introduced road type identifier, there are five main blocks in the model of AFS controller (FIG.6). The motor state initialization block including a motor error state detection module and a motor zero-point learning algorithm. The vehicle yaw state estimation block estimates the vehicle heading angle based on the speed and acceleration, and predicts the vehicle track assisted by the analyzing of steering wheel angle. The following-up steering block and the automatic low beam leveling block calculate the output angle of the orientation motor of steering beam and low beam respectively at the current state. Finally, the controller of orientation motor calculates the pulse signal for the step motor.

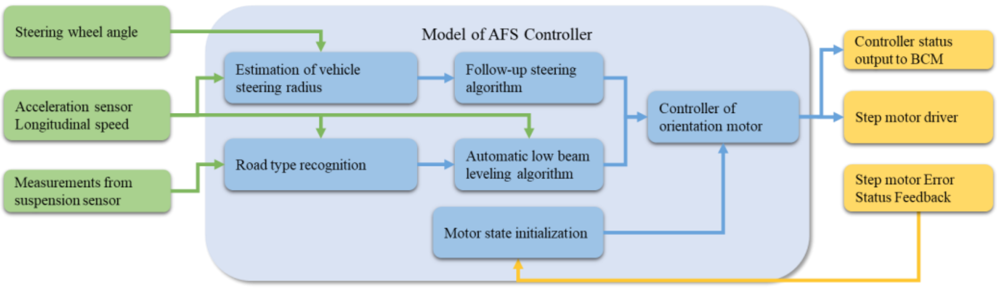

FIG.6 The architecture and functions AFS controller model.

I designed a series of testing procedures for the intelligent vehicle lighting control system, built the model-in-loop (MIL) and the hardware-in-loop (HIL) simulation platform. All the calibration and testing of the system have been done at the test plant provided by China FAW Group. The MIL simulation platform is built based on Simulink, ANSYS Vrexperience and veDYNA. I designed the control interface of the intelligent vehicle lighting system in Simulink (FIG.7A), which can simulate the driver operating the light switch or depressing the brake and gas pedal. The tester can also manually adjust the error status of each sensor or controller through this interface to verify whether the system responses correctly to the error. ANSYS Vrexperience and veDYNA were used to simulate the optical environment and the virtual vehicle attitude changing during acceleration or deceleration. It is intuitively to observe the effect of designed lighting control system through this MIL simulation platform (FIG.7B). It is also available to observe and verify the light intensity of vehicle headlamps at the specific checkpoints (FIG.7C and FIG.7D).

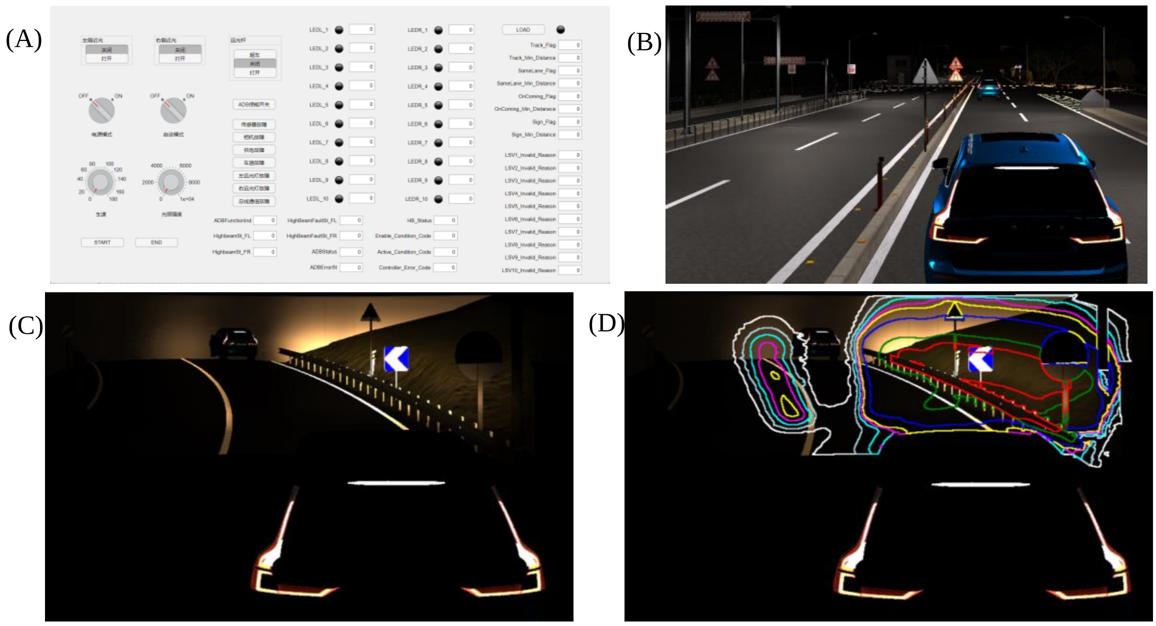

FIG.7 MIL simulation platform (A) The tester interface of MIL simulation platform; **(B)** The virtual optical environment in ANSYS Vrexperience. **(C)** Checking the light intensity with Aiming wall; **(D)** Checking the light intensity with Isolines.

The HIL simulation platform consists of a simulation computer, the Vector CANoe, the electronic control unit of lighting system, the LED headlamps and some step motors. All parts of the platform are connected by CAN bus (FIG.8A). Specifically, the CANoe is able to send the CAN messages generated from the simulation computer or replay the Can messages collected from the real test vehicle. The embedded software can be further debugged by observing the control effect of the lights and motors (FIG.8B).

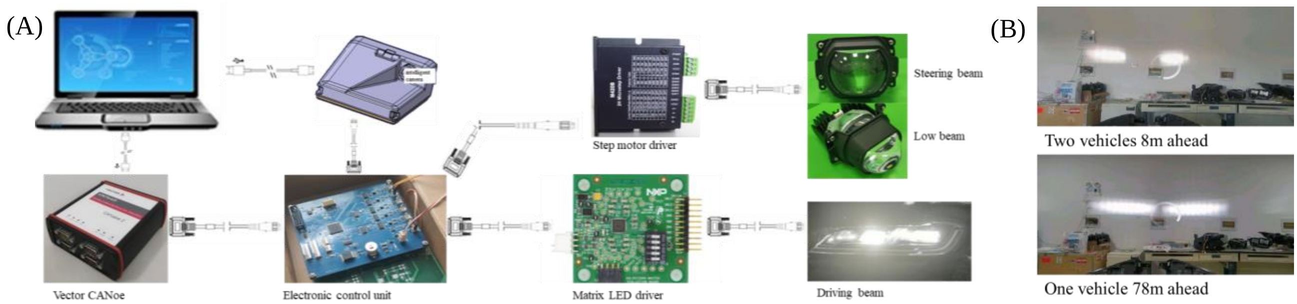

FIG.8 HIL simulation platform (A) The architecture of HIL simulation platform; **(B)** The process of debugging the adaptive driving beam system by replaying data.