

Adaptive Driving Speed Guiding to Avoid Red Traffic Lights

Lushen Cai

Automotive College and Clean Energy Automotive
Engineering Center
Tongji University
Shanghai, China
e-mail: lushen.cai@gmail.com

Guobao Ning (Corresponding Author)

Automotive College and Clean Energy Automotive
Engineering Center
Tongji University
Shanghai, China
e-mail: Ngb2000@sina.com

Abstract—This paper proposed an Adaptive Driving Speed Guiding (ADSG) concept, which can assist drivers to avoid red traffic lights and thus improve the driving efficiency. The functional workflow of ADSG is defined and a smartphone based system composition is designed. In order to figure out a legal, feasible and efficient guidance speed, 2 algorithms are designed, simulated and evaluated in the paper. After the evaluation a control strategy is raised, which combines the advantages of the 2 algorithms.

Keywords—Speed Guiding; Driver Assistance Systems; Energie Saving; Vehicle Communication Systems

I. INTRODUCTION

Traffic Light is an indispensable part of the present traffic system, which organizes the vehicles from different directions to pass crossroads in a proper sequence. However, the red traffic light forces drivers to brake-to-stop and wait. This brake-to-stop and engine idling during the waiting can cause extra fuel consumption and emission [1].

Therefore, Methods to avoid red traffic lights are necessary and meaningful. Currently there are two main methods for this task, which are: (1) Construction of highways without crossroads. (2) A dynamic traffic lights infrastructure, which can adapt the switch regulation according to the real time traffic condition.

The method to construct highways is presently broadly used, but these constructions are quite costly and cannot reach every detail parts of the world. The dynamic traffic light, on the other hand, can reach every small road with traffic lights. But this technology requires a communication between traffic lights and approaching vehicles. Although technically this concept is feasible, the spread of communication components in every vehicle and traffic light need a lot of time and investment [2].

In view of the limitations of the above-mentioned solutions, this paper put forward a new method to solve the problem: Adaptive Driving Speed Guiding (Abb. ADSG).

II. THE FUNCTION AND COMPOSITION OF ADAPTIVE DRIVING SPEED GUIDING (ADSG)

A. Function and Workflow of ADSG

The sub-functions of Adaptive Driving Speed Guiding (ADSG) are: (1) Collect real time traffic lights information

of the approaching crossroad and the information of vehicle speed and destination. (2) Calculate an optimized guidance speed that can guide driver to pass crossroad when the green light is on. (3) Present the calculated result to the driver. (4) Observe the drivers' reaction and adaptively generate new guiding speed according to different driver reaction. The workflow of ADSG is described as Figure 1.

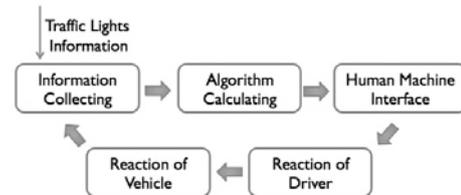


Figure 1. Workflow of ADSG System

B. Composition of Adaptive Driving Speed Guiding

To realize the above-mentioned functions, the system should consist of the following 3 sub-functional components: (1) Information Collecting Unit, (2) Calculation Unit and (3) Human Machine Interface.

Here, the necessary information of ADSG is:

- Real time vehicle speed.
- Real time vehicle position and distance to the approaching crossroad.
- Real time traffic lights information and switch intervals at the approaching crossroad.
- Speed limit of the road, on which the vehicle is running.
- Orientation information to know in which direction the driver want to go at the approaching crossroad.

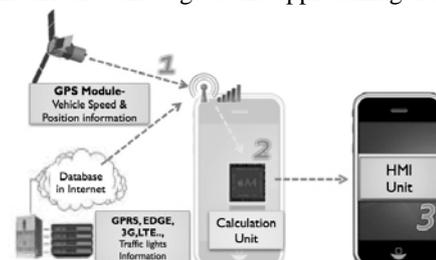


Figure 2. ADSG Workflow on Smartphone Platforms

In order to make ADSG system affordable for most drivers, this paper chose smartphone as the working platform of ADSG. Figure 1 explains how ADSG system

works on smartphone: Firstly, via GPS module the vehicle position and speed can be acquired. Meanwhile, via mobile network such as 3G and LTE the traffic lights information and road speed limit can be downloaded from the online database. Secondly, the collected information is further calculated by the smartphone CPU and finally the results are presented to drivers through the screen and loudspeaker of smartphone.

III. ALGORITHM OF ADAPTIVE DRIVING SPEED GUIDING

A. Input parameters and their acquisition method

The algorithm of ADSG aims to figure out an optimized driving speed, which can guide drivers to pass the approaching crossroad at a proper speed. To this purpose the following input parameters are necessary:

1) Distance from vehicle to the approaching crossroad.

In smartphone based ADSG, the distance from vehicle to the approaching crossroad is acquired and continuously updated by the GPS module of smartphone. Since the accuracy of GPS module is limited and the signals will sometimes missing, it is recommended to use Dead Reckoning algorithm to modify the distance [3][4].

2) Real time traffic lights information and their switch intervals.

In ADSG algorithm, the real time traffic lights information is represented by parameter t_{re} , ('re' refers to 'remain'), which indicates the remaining time of the current light color. In order to differentiate red light and green light, it is defined as below:

- $t_{re} > 0$, green light is on.
- $t_{re} < 0$, red light is on.
- $t_{re} = 0$, traffic light is switching.

For example, $t_{re} = 5$ indicates that green light is currently on and will turn red after 5 seconds.

For safety reasons, ADSG will not guide drivers to pass through the crossroad when the yellow light is on. Therefore the yellow light condition is regarded as red light in the algorithm.

The switch interval T_{red} , T_{green} indicate the duration of red light and green light. As what has been mentioned, in ADSG algorithm $T_{red} = T_{red} + T_{yellow}$.

In smartphone based ADSG, t_{re} , T_{red} and T_{green} are beforehand downloaded and continuously updated from the online database via the mobile network. In China the database of real time traffic light information are owned by the ministry of transportation.

3) Speed limit of the Road

For safety reasons, ADSG will never output a recommended speed that exceeds the speed limit. The speed limit V_{limit} is regarded as a threshold in the algorithm.

The speed limit information can be obtained from the online database.

4) Real time vehicle speed.

The Real time vehicle speed has 2 functions in the ADSG algorithm. On one hand, it is used to calculate the

'response time', namely, the time the driver will need to accelerate / decelerate the vehicle from current speed to the guidance speed. On the other hand, the real time vehicle speed serves as a feedback signal in the control loop. The algorithm will continuously compare the real time vehicle speed with the guidance speed, when the vehicle speed dose not follow the guidance speed, or the vehicle speed is suddenly changed because of e.g. an emergency, the system will immediately recalculate a new guidance speed.

In smartphone based ADSG, the real time vehicle speed is acquired via GPS system. In order to get a more accurate vehicle speed, it is recommended to get this information from the vehicle CAN system (e.g. OBD Interface), however some extra device maybe needed.

5) Vehicle acceleration

As what has been mentioned in III.A.4), vehicle acceleration is used to forecast the 'response time'. Besides, as what Figure 3 has shown, the vehicle acceleration can also be regarded as a feedback signal, which indicates the driver's intention and shows if the driver will follow the guidance of ADSG.

The acceleration information can be obtained from the derivation of speed or acceleration sensors. In the simulation of this paper, the average acceleration are assumed to the following values to describe the 4 possible driving situations:

- $a = 1.5 \text{ m/s}^2$: accelerating
- $a = 0 \text{ m/s}^2$: cruising
- $a = -0.5 \text{ m/s}^2$: gliding without braking and accelerating
- $a = -3 \text{ m/s}^2$: braking for emergency

The assumption is aligned with the accelerating strategy of the Full Range Speed ACC, which is defined by working group 14 of ISO TC214 [5]. (see also Figure 4)

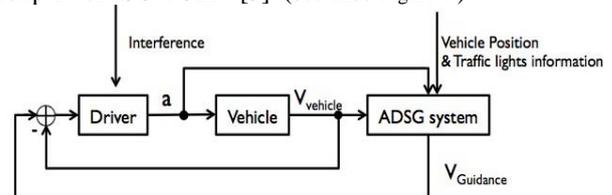


Figure 3 Flow chart of ADSG Control Loop

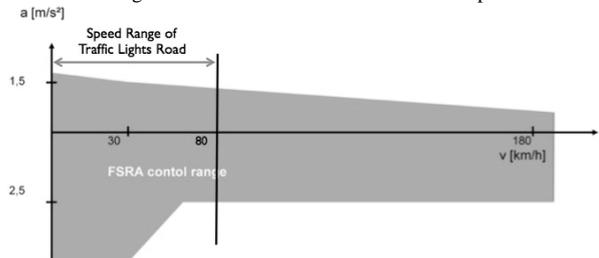


Figure 4. Acceleration Strategy of Full Range Speed ACC

B. Algorithm Output and Its Requirements

The target of the algorithm is to output a guidance speed, at which the driver can pass the crossroad when the green light is on. Because the green light last for a period and will

appear again in the next red-green switch cycle, there exist a lot of possible guidance speeds.

Therefore it is necessary to first define the requirements of the output. An ideal output should meet the following requirements:

1) *Legality*

The guidance speed should by no means exceed the speed limit of the road.

2) *Feasibility*

The recommended speed should be feasible for the driver under any traffic conditions.

This paper regards the current vehicle speed as the most feasible speed for the driver, therefore the feasibility of the guidance speed is described by the difference between real time vehicle speed and guidance speed. The smaller the difference is, the better the feasibility.

3) *Driving efficiency*

Driving efficiency is also an important requirement for the guidance speed. Driving efficiency requires the ADSSG to provide a guidance speed, which can let the driver pass the road as fast as possible.

Among the 3 requirements, the requirement of legality is a must and can easily be fulfilled. However, the requirements of feasibility and driving efficiency are somehow contradictory to each other, an efficient driving speed is sometimes risky and not feasible. Therefore this paper designed 2 different algorithm logics to respectively fulfill the 2 requirements.

C. 'Best Feasibility' Algorithm

As what has been defined in III.B.2), in order to achieve the best feasibility, the difference between guidance speed and real time vehicle speed should always be the smallest.

Therefore the core idea of the 'Best Feasibility' algorithm is: use the real time vehicle speed as an initial

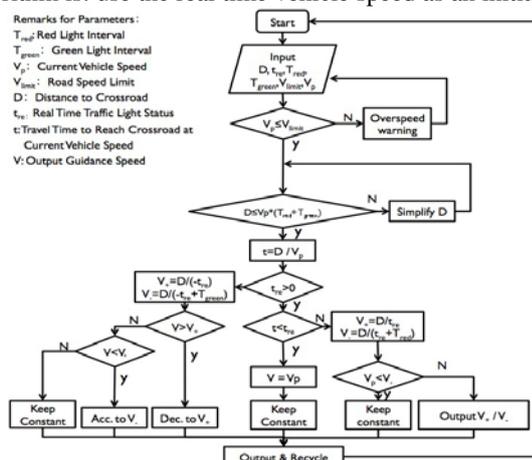


Figure 5. Flow Chart of 'Best Feasibility' Algorithm

The core process of 'Best Feasibility' algorithm is shown in Figure 5.

D. 'Best Efficiency' Algorithm

The 'Best Efficiency' algorithm aims to figure out a fastest speed, which is under speed limit and can let the driver pass the crossroad with the shortest time.

Therefore the algorithm use the road speed limit as the initial value, first calculate if the vehicle can pass the crossroad at the speed limit. If not, the algorithm will downwards to find a fastest speed as an output. Specially, in order to improve the feasibility of this algorithm, the system will also calculate and output a lower limit. The core process of 'Best Efficiency' algorithm is shown in Figure 6 .

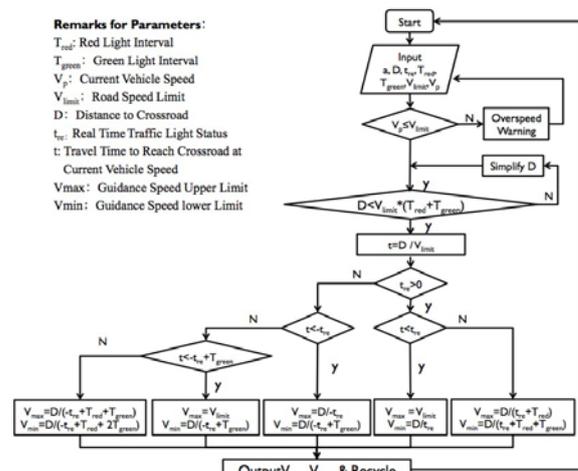


Figure 6. Flow Chart of 'Best Efficiency' Algorithm

IV. SIMULATION AND EVALUATION OF THE ALGORITHMS

With the purpose of verification and evaluation, the algorithms are coded and simulated in Matlab. The input parameters are listed in Table 1.

Figure 7 is the simulation result of 'Best Feasibility' Algorithm. In this simulation, the vehicle speed dose not response to the guidance speed, namely the speed control system is an open loop. It is obvious, that the 'Best Feasibility' algorithm can always generate a guidance speed (black line), which is closest to the current vehicle speed.

Figure 8 shows the result of 'Best Efficiency' algorithm. The simulation input is the same as the input of Figure 7. According to the vehicle speed (red line), 'Best Efficiency' algorithm output an upper speed limit (blue line) and a lower speed limit (green line). This 2 guidance speed limit indicate a speed range, all speed in this area can let the driver pass the crossroad when the green light is on. It can be seen that in the first 30s of the simulation, the actual upper speed limit is higher than the speed limit, for legality reasons the algorithm output the speed limit as the recommended upper speed limit.

Figure 9 combined the results of Figure 7 and Figure 8 together. It is obvious, that compared with 'Best Efficiency' algorithm, 'Best Feasibility' algorithm always

recommended a speed which is closest to the current vehicle speed. The algorithm usually tells the driver to keep the speed and rarely to change speed.

On the other hand, 'Best Efficiency' algorithm always recommended an upper speed limit which is closest to the road speed limit. The algorithm always tells the driver to accelerate.

Figure 10 simulated a closed loop control condition, namely the driver will change the vehicle speed according to the guidance speed. Specially, the simulation also assumed an emergency condition (Grey area), in this emergency area the vehicle are not allowed to run faster than 6 m/s.

The red line shows the responding vehicle speed under the control of 'Best Efficiency' algorithm. Before the Emergency the algorithm guide the driver to reach the maximum speed. However, when the emergency happened, the algorithm cannot recognize it and still ask the driver to accelerate (blue and green line).

The blue dash line shows the responding vehicle speed under the control of 'Best Feasibility' algorithm. Although the traffic condition before the emergency allows the vehicle to accelerate, the algorithm asked the driver to keep current speed. When the emergency happened, the guidance speed (black line) detected it and guided the driver to drive at a speed lower than 6 m/s, if the emergency lasted long enough, this speed would be the best. However when the emergency is past and allows the vehicle to accelerate again, the algorithm fail to recognize this and continue to guide the vehicle at an inefficient speed.

As a conclusion, 'Most Feasibility' algorithm is good for low speed driving and braking, it react especially well by emergency brake situation. Conversely, 'Most Efficiency' algorithm cannot provide an useful and feasible speed guidance by the emergency, but it can take full advantage of the acceleration potentials at a good traffic condition.

Therefore, an optimized speed guiding strategy is to use 'Best Feasibility' algorithm in the low speed and braking situation, and use 'Best Efficiency' algorithm in the high speed and accelerating situation.

Table 1. Initialization of the Input Parameters

Symbol	Value	Unit	Meaning	Symbol	Value	Unit	Meaning
T_{red}	30	s	Red Light Interval	V_{limit}	14	m/s	Road Speed Limit
T_{green}	30	s	Green Light Interval	D_0	1500	m	Distance to Crossroad
V_{pt}	8	m/s	Present Vehicle Speed	t_{to}	8	s	Present Traffic Light Status

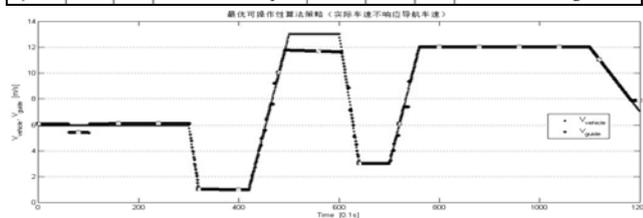


Figure 7. Simulation of 'Best Feasibility' Algorithm (Open Loop)

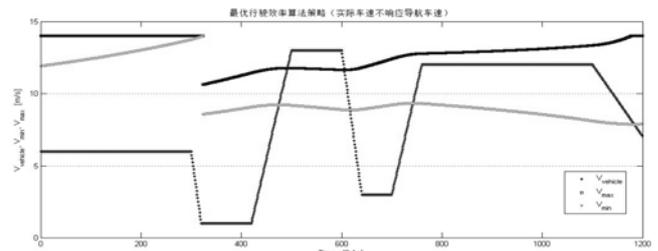


Figure 8. Simulation of 'Best Efficiency' Algorithm (Open Loop)

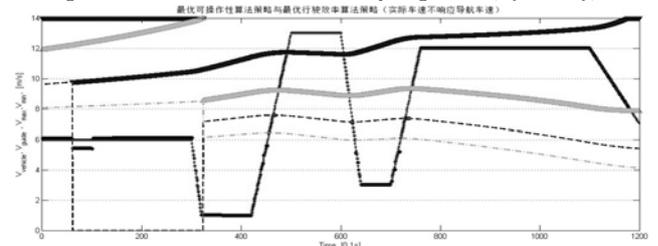


Figure 9. Comparison of 'Best Feasibility' and 'Best Efficiency' Algorithm (Open Loop)

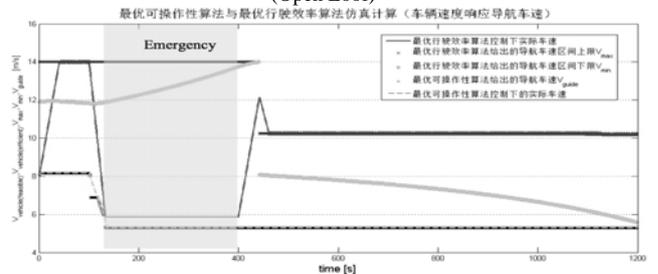


Figure 10. Comparison of 'Best Feasibility' and 'Best Efficiency' Algorithm (Closed Loop)

V. CONCLUSIONS

The conclusions of this paper are as follows:

- Raised the concept of Adaptive Driving Speed Guiding (ADSG).
- Designed the workflow and hardware composition of ADSG.
- Defined the requirements of ADSG algorithm.
- Designed and evaluated 2 algorithms for ADSG.
- Found a control strategy to meet both feasibility and efficiency requirements of ADSG.

REFERENCE

- [1] Ullrich, Günter. Fahrerloses Transportsystem, Wiesbaden. 2011
- [2] Richard van Basshuysen. Fahrgewentwicklung im Wandel. ATZ, 2010
- [3] Paul D Groves, Principles of GNSS, inertial, and multisensor integrated navigation systems. 2008
- [4] Nico Dziubek, Lehrfolien – Positioning and Navigation, FZD in TU Darmstadt, 2010
- [5] Hermann Winner. Handbuch Fahrerassistenzsysteme. S480. ATZ. 2009, pp.481