Distributed Fault Detection and Estimation for Cooperative Adaptive Cruise Control System in a Platoon

Zoleikha Abdollahi Biron[1], and Pierluigi Pisu[2]

[1,2]*Department of Automotive Engineering Clemson University, Greenville, SC, 29607, USA*

zabdoll@clemson.edu
pisup@clemson.edu

ABSTRACT

Wireless vehicle to vehicle communication in a vehicle platoon is proposed in intelligent transportation systems to increase safety of the transportation system and assist drivers for improved decision making. However, similar to any networked system, cooperative connected vehicles in a platoon are vulnerable to malfunction due to failures in network communication. In addition to the possible data corruption, sensor and actuator faults can have significant effects on the control strategy for cruise control. This paper considers a platoon of connected vehicles equipped with cooperative adaptive cruise control and presents a reconstructive method based on sliding mode observer to estimate and reconstruct the faults in the sensors and actuators of vehicles.

1. INTRODUCTION

These Traffic congestions, limited road throughput and safety concerns in the past two decades led the automobile industry towards the idea of traffic control using intelligent vehicles. Consequently, the connectivity concept in vehicles has become a hot topic of research. Current smart vehicles are equipped with more than 70 electronic controls units (ECUs), Bluetooth, and Wi-Fi enabling them to communicate with the external networks (Larson, and Nilsson, 2008). Research in automotive control led to the development of radar-based adaptive cruise control (ACC) which can have a positive impact on vehicle safety, driver comfort, and highway efficiency (Kester, Willigen, and Jongh, 2014). Research and development in cruise control focuses on enabling more and better cooperation between ACC systems using communication and information transmission between vehicles using specific communication protocols such as Dedicated Short Range Communication (DSRC). Adaptive cruise control is operated without wireless communication link to enhance driving ability in traffic throughput, while maintaining a sufficient level of safety distance between vehicles (Ploeg, Semsar-Kazerooni, Lijster, Wouw, and Nijmeijer, 2013). Cooperative adaptive cruise control is indeed a step ahead of the ACC system in the vehicles and can be considered as a major development in recent research on intelligent transportation systems (ITS). CACC takes the ACC to the next level by receiving the other vehicle’s information through wireless communication, and tries to minimize the distance between vehicles in the range of couple of meters.

In addition to reducing the traffic congestion and increasing the road throughput, the CACC system can cause significant reduction in aerodynamic drag, especially for heavy-duty vehicles, thereby decreasing fuel consumption (Al Alam, Gattami, and Johansson, 2010).

As it is mentioned before, cooperative vehicles regarding to their Wi-Fi communications and also in-vehicle network communications as in a CAN bus, are vulnerable to network communication failures and faults in sensor and actuators. Numbers of papers have addressed the significant vulnerabilities of the vehicles and possible hacking in the vehicle network which can lead to significant security issues for all vehicles connected to the victim vehicle (Nilsson, and Larson, 2008).

Several researches on platooning vehicles with the main focus of control strategy design for CACC system and fault tolerant control is available in the literature (Lygeros, Godbole, and Broucke (2008), Zhang, Gantt, Rychlinski, Edwards, Correia, and Wolf (2009), Han, Chen, Wang, and Abraham (2013)). In (Nunen, Ploeg, Medina, and Nijmeijer, 2013), authors discuss that the CACC system cannot rely on the driver as a backup and is constantly active, therefore more prominent to the occurrences of faults (such as packet loss in the wireless communication). Hence, they present an algorithm which uses the availability of sensor-data in each moment in time to calculate in real-time a safe distance for the CACC system.

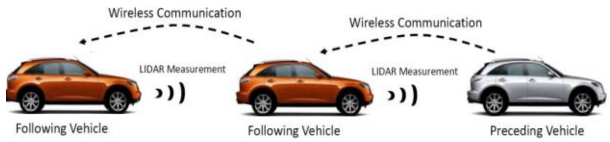

Figure 1. A platoon of vehicles equipped with CACC

In this paper, a distributed fault reconstructive method based on sliding mode observer has been presented. In this method an estimation of faults on the sensors can be derived from the observers and consequently, each vehicle can reconstruct the faulty data before transmitting the data to the neighboring vehicles. Therefore, all incoming data from the network to the vehicles are considered to be healthy data, which enables the second observer to detect the faults on the actuators.

The rest of this paper is organized as follows: Section II presents modeling of the vehicle platoon. Fault diagnosis problem statement is described in section III. In section IV, we provide observer based fault reconstructive method and finally, simulation results for a vehicle platoon with 3 vehicles are shown and discussed in section V.

2. System modeling

In the current existing ACC system, the range (i.e., relative distance) and range rate to the preceding vehicle are measured with a radar or LIDAR sensor (Bu, Tan, and Huang, 2010). While, cooperative adaptive cruise control (CACC) is essentially a vehicle-following control methodology that automatically accelerates and decelerates so as to keep a desired distance to the preceding vehicle (Rajamani, and Zhu, 2002). To do this, in addition to onboard sensors, such as radars, vehicles should be equipped with wireless communication devices, such as DSRC, to receive extra information of the preceding vehicle(s) e.g., the desired acceleration is received through a wireless communication link.

2.1. Vehicle Dynamics

A nonlinear model has been considered for each vehicle in the platoon as shown in Eq. (1).

$$\begin{pmatrix} \dot{x}_i \\ \dot{v}_i \end{pmatrix} = \begin{pmatrix} v_i \\ u_i - \dfrac{1}{2M_i} C_d \rho_a A_i v_i^2 - C_r g\cos(\theta) - g\sin(\theta) \end{pmatrix} \quad (1)$$

$$= \begin{pmatrix} v_i \\ u_i - \alpha_i v_i^2 - \beta_i \end{pmatrix} \qquad i = 1,2,3$$

Where, x_i and v_i are two states of each vehicle representing the absolute position and velocity of the i[th] vehicle. u_i is the acceleration control input per unit mass generated by CACC strategy to accelerate or decelerate the car which is derived from control strategy. Also, M_i, C_d, ρ_a, A_i, C_r and θ are the mass, drag coefficient, the air density and frontal area, the rolling resistance coefficient and the road gradient, respectively for the i[th] vehicle in platoon Assuming the road gradient as small enough, both α_i and β_i will be positive constants.

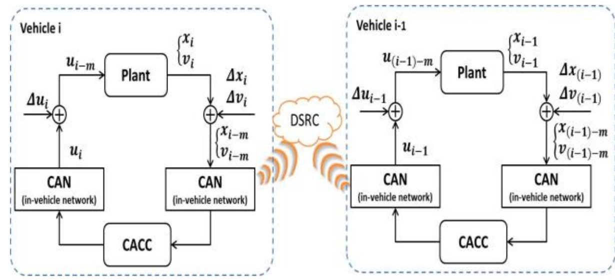

Figure 2. Block diagram of connected vehicles with faults and communication networks

As it can be inferred from Fig. 1, each vehicle transmits its own information through the CAN bus and DSRC communication. CAN bus is considered as an in-vehicle network and DSRC is indeed inter-vehicle network.

2.2. Control Strategy

A cooperative adaptive cruise control is considered for this paper as the control strategy for each vehicle in the platoon. This controller works as a higher level of controller generating the demanded acceleration for the vehicle based on the information exchange with the preceding vehicle and the current states of the vehicle. The block diagram of the whole platoon model is depicted in Fig .2.

The control strategy used in this paper can be modeled as a discrete event system containing three main states as gap filling, gap regulating and suitable distance state. Each state has its own specific control rule (dynamic) based on the relative distance and speed between the host vehicle and its target (which is its preceding vehicle) Fig. 3.

In each state, control command is generated such that all constraints on the vehicle's speed limits; acceleration and deceleration limits are satisfied.

$$\begin{aligned} &v_{\min} < v_i < v_{\max} \\ &a_{\min} < a_i = u_i < a_{\max} \end{aligned} \quad (2)$$

Where, $v_{\min}$ and $v_{\max}$ are minimum and maximum speed limitations respectively. Similarly, $a_{\min}$ and $a_{\max}$ are the minimum and maximum acceleration limitations for each vehicle.

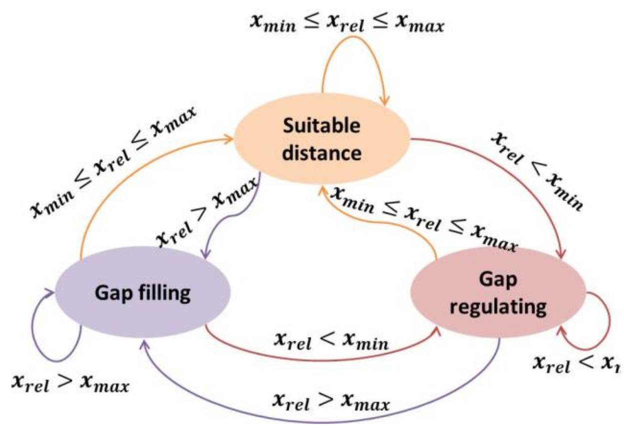

Figure 3. Discrete event state machine of control strategy in CACC of each vehicle

1- Gap filling state:

When the relative distance between host vehicle and target, x_{rel} , is more than maximum distance limit $x_{\max}$, the CACC controller of host vehicle enters the gap filling state and based on two vehicles' current states (velocity and position), and by considering the system constraints, a satisfactory acceleration for the host vehicle is determined to fill the gap between the two vehicles. The preceding vehicle is identified as the vehicle directly in front of the host vehicle.

2- Gap regulating state:

When the relative distance between two vehicles is less than the desired gap, $x_{\min}$, CACC gap regulation controller is engaged immediately. The gap regulating controller should be precise enough to satisfy stringent performance criteria under various constraints such as limitation on speed and brake actuation.

3- Suitable distance sate:

When the relative distance between target and host vehicle is between the gap filling and gap regulating conditions, both vehicles cruise with zero relative speed keeping similar acceleration profile.

3. PROBLEM STATEMENT

As mentioned before, the cooperating vehicles are vulnerable to communication failures and faults in sensors and actuators. These faults can be physical faults on the system or can be a consequence of a virus on the CAN bus in the vehicle internal network due to the open source ECUs, Bluetooth and Wi-Fi connection (Nilsson et.al, 2008).

In this paper, we consider faults in the sensors (velocity and position) and the actuator (the accelerating or braking pedal). To this end, the main goal of this paper is fault detection and isolation using advantages of communication and extra information in the connected vehicles. To achieve this, we propose an observer-based diagnostics strategy to estimate and reconstruct faults in the sensors in each individual vehicle. This strategy will enable each vehicle in the platoon to detect, isolate and reconstruct the fault in its position or velocity sensor. Therefore, each vehicle can correct the sensor reading rejecting the effect of the faults before transmitting data through the network. Consequently, each vehicle in the platoon will receive correct sensor data from its preceding vehicle which in turn will make the reliable control input command available to the observers. Therefore, the controllers' output will not be corrupted and the observers in each individual vehicle will be able to detect the actuator faults. The following section describes the observer design strategy and fault signature for each vehicle in the platoon.

4. OBSERVER DESIGN

Considering vehicle dynamics in Eq. (1), sensors fault can be injected as measurement fault for each state. As a result, faults in sensors and actuator can be modeled as:

$$\begin{aligned} x_{i-m} &= x_i + \Delta x_i \\ v_{i-m} &= v_i + \Delta v_i \qquad\qquad i = 2,3 \\ u_{i-m} &= u_i + \Delta u_i \end{aligned} \tag{3}$$

Where x_{i-m} , v_{i-m} and u_{i-m} are measured values of position, velocity and control command of i[th] vehicle respectively. Δx_i , Δv_i and Δu_i represent measurement faults on position sensor, velocity sensors and actuator.

The observer structure is chosen based on sliding mode methodology and is given by:

$$\dot{\hat{x}}_i = v_{i-m} + \eta_{xi}\,\mathrm{sgn}(x_{i-m} - \hat{x}_i) \tag{4}$$

$$\dot{\hat{v}}_i = u_i - \alpha_i \hat{v}_i^2 - \beta_i + \eta_{vi}\,\mathrm{sgn}(v_{i-m} - \hat{v}_i) \tag{5}$$

Where $\hat{x}_i$ and $\hat{v}_i$ present the estimated position and velocity of i[th] vehicle respectively. $\mathrm{sgn}(.)$ stands for sign function and η_{vi} is sliding mode observer gain. Consequently, the error dynamics will be:

$$\dot{\tilde{x}}_i = \dot{x}_i - \dot{\hat{x}}_i = v_i - v_{i-m} - \eta_{vi}\,\mathrm{sgn}(x_{i-m} - \hat{x}_i) \tag{6}$$

$$\dot{\tilde{v}}_i = \dot{v}_i - \dot{\hat{v}}_i = -\alpha_i (v_i^2 - \hat{v}_i^2) - \eta_{vi}\,\mathrm{sgn}(v_{i-m} - \hat{v}_i) \tag{7}$$

Where $\tilde{x}_i$ and $\tilde{v}_i$ stand for position estimation error and velocity estimation error respectively.

4.1. Velocity Sensor Fault

In presence of fault on the velocity sensor in each vehicle, by considering the second observer equation, Eq. (5), the sliding surface can be written as:

$$s_{vi} = v_{i-m} - \hat{v}_i = v_i + \Delta v_i - \hat{v}_i \tag{8}$$

We choose the Lyapunov candidate as:

$$V_{vi} = \frac{1}{2} s_{vi}^2 \tag{9}$$

Using Eq. (7) as error dynamic, the derivative of the Lyapunov function candidate can be written as:

$$\begin{aligned} \dot{V}_{vi} &= s_{vi}\,\dot{s}_{vi} = s_{vi}\left[\dot{v}_{i-m} - \dot{\hat{v}}_i\right] \\ &= s_{vi}\left[-\alpha_i(v_{i-m}^2 - \hat{v}_i^2) - \eta_{vi}\,\text{sgn}(v_{i-m} - \hat{v}_i)\right] \end{aligned} \tag{10}$$

$$\begin{aligned} \dot{V}_{vi} &= s_{vi}\left[-\alpha_i(v_{i-m} - \hat{v}_i)(v_{i-m} + \hat{v}_i) - \eta_{vi}\,\text{sgn}(v_{i-m} - \hat{v}_i)\right] \\ &= s_{vi}\left[-\alpha_i(s_{vi})(v_{i-m} + \hat{v}_i) - \eta_{vi}\,\text{sgn}(s_{vi})\right] \\ &= -\alpha_i\,(v_{i-m} + \hat{v}_i)s_{vi}^2 - \eta_{vi}\,s_{vi}\,\text{sgn}(s_{vi}) \\ &\le -\alpha_i\,(v_{i-m} + \hat{v}_i)\left|s_{vi}\right|^2 - \eta_{vi}\left|s_{vi}\right| \end{aligned} \tag{11}$$

By selecting η_{vi} as a large positive constant, $\dot{V}_{vi}$ can be made negative definite and the sliding surface can be achieved in finite time.

On the sliding manifold, we have $s_{vi} = 0, \dot{s}_{vi} = 0$ from which we can write the following:

$$\begin{aligned} & s_{vi} = v_i + \Delta v_i - \hat{v}_i = \tilde{v}_{vi} + \Delta v_i \\ & \begin{cases} s_{vi} \to 0 \\ \dot{s}_{vi} \to 0 \end{cases} \Rightarrow \begin{cases} \tilde{v}_{vi} = -\Delta v_i \\ \dot{\tilde{v}}_{vi} = -\Delta \dot{v}_i \end{cases} \end{aligned} \tag{12}$$

4.2. Position Sensor Fault

In occurrence of position sensor fault, the first sliding mode observer, Eq. (4), is considered and the sliding surface is defined as the following:

$$s_{xi} = x_{i-m} - \hat{x}_i = x_i + \Delta x_i - \hat{x}_i \tag{16}$$

A positive function $V_{xi} = \frac{1}{2} s_{xi}^2$ is chosen as a Lyapunov candidate to analyze the stability of the first observer error dynamics.

$$\begin{aligned} \dot{V}_{xi} &= s_{xi}\,\dot{s}_{xi} \\ &= s_{xi}\left[\dot{x}_{i-m} - \dot{\hat{x}}_i\right] = s_{xi}\left[v_{i-m} - v_{i-m} - \eta_{xi}\,\text{sgn}(x_{i-m} - \hat{x}_i)\right] \end{aligned} \tag{17}$$

$$\begin{aligned} \dot{V}_{xi} &= s_{xi}\left[-\eta_{xi}\,\text{sgn}(x_{i-m} - \hat{x}_i)\right] = -\eta_{xi} s_{xi}\,\text{sgn}(s_{xi}) \\ &\le -\eta_{xi}\left|s_{xi}\right| \end{aligned} \tag{18}$$

Therefore, by choosing sliding mode gain, η_{xi} , as a large positive constant, $\dot{V}_{xi}$ can be made negative definite and the sliding surface can be achieved in finite time.

On the sliding manifold,

$$\begin{aligned} & s_{xi} \to 0\ ,\ \dot{s}_{xi} \to 0 \\ & s_{xi} = x_i + \Delta x_i - \hat{x}_i = 0 \\ & \tilde{x}_i = x_i - \hat{x}_i = -\Delta x_i \end{aligned} \tag{19}$$

Then the error dynamic can be rewritten as:

$$\dot{\tilde{x}}_i = -\Delta \dot{x}_i = -K_{xi} \tag{20}$$

Where K_{xi} is filtered version of switching term $\eta_{xi}\,\text{sgn}(x_{i-m} - \hat{x}_i)$ and it is referred as equivalent output error.

Based on Eq. (20), integrating K_{xi} gives estimation of the fault on position sensor.

Residual R_2 is defined equal to the equivalent output error, K_{xi}, on sliding surface . As it is expected from Eq. (20), fault in the position sensor will show up in this residual.

4.3. Actuator Fault

Assuming the correct data received from preceding vehicle, second sliding model observer is used to detect the actuator fault.

The error dynamic is described in the following set of equations

$$\begin{cases} \dot{v}_i = u_i - \alpha_i v_i^2 - \beta_i \\ \dot{\hat{v}}_i = u_{i-m} - \alpha_i \hat{v}_i^2 - \beta_i + K_{vi} \end{cases} \tag{21}$$

$$\begin{aligned} \dot{\tilde{v}}_i &= u_i - u_{i-m} - \alpha_i(v_i^2 - \hat{v}_i^2) + K_{vi} \\ &= \Delta u_i - \alpha_i(v_i^2 - \hat{v}_i^2) + K_{vi} \end{aligned} \tag{22}$$

Therefore, choosing the sliding surface as Eq. (8) and Lyapunov candidate as Eq. (9), the stability of the error dynamic can be described via Eq. (11).

Hence, on the sliding surface, fault on the actuator will show up in the residual R_2 which is generated by filtering the switching term of second sliding mode observer.

4.4. Fault Signature

As it is explained in the previous sections, each fault shows up in different residual. Based on the residuals and threshold setting for each residual, fault signature table is generated as table 1.

Table 1. Fault signature

Fault Signature			
Residual	Velocity sensor fault	Position sensor fault	Actuator fault
S_1	1	0	1
S_2	1	1	0

As it can be inferred from the table, using the distributed fault diagnostics algorithm in each vehicle, faults in sensors and actuator can be detected and isolated.

5. SIMULATION AND RESULTS

To simulate faults in the connected vehicles, a platoon of three vehicles equipped with cooperative adaptive cruise controller has been considered. A distributed fault diagnostics algorithm is implemented for each following vehicle. The leader vehicle follows the velocity profile of US06 as highway driving profile. The length of driving cycle is 600 seconds and during the whole cycle single fault scenario is simulated for different faults on sensors and actuator for each car. In the flowing the results for the second vehicle will be explained as an example.

A position sensor fault as a bias of 0.05% of current position with constant slop occurs at t= 100 seconds and it remains for 75 seconds. A velocity sensor fault is injected at time 350 second and remains for 75 seconds. This fault is simulated as a bias of 0.5 m/s which is equal to 2 % of maximum velocity. At the end, the actuator sensor fault occurs at t= 500 seconds as a bias of 1% of normal actuator value and it remains for 50 seconds in the system. Fig. 4 shows both residuals R_1 and R_2. As it can be seen, faults in velocity sensor and actuator show up in R_1 while faults in position and velocity sensors will change the value in R_2.

In order to have a satisfactory low false alarm for each residual, a fix threshold based on probability distribution method is chosen. Fig. 5 shows threshold setting in second residual as an example.

For each residual threshold is set as (23). Whenever one of these residuals surpasses its own threshold, a fault detection signal will be triggered to declare that fault is detected in each vehicle in the platoon.

$$|R_i| > \gamma_i = \mu_i + 1.1\sigma_i \qquad (23)$$

As it is expected, faults on the velocity sensor and actuator will trigger R_1 in real time, while faults on position and velocity sensors can trigger the R_2. Therefore, faults in each vehicle in the platoon will follow the signature mentioned in the table 1. These faults can occur in a system due to GPS, radar or laser measurements. Fig. 6 depicts fault signature when three different faults happens in the system, as it be inferred from the figure, the signature is as described in table 1.

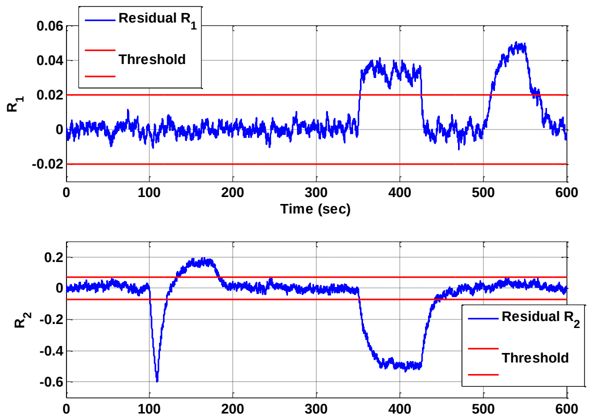

Figure 4. Residuals in presence of faults in sensors and actuator in vehicle # 2

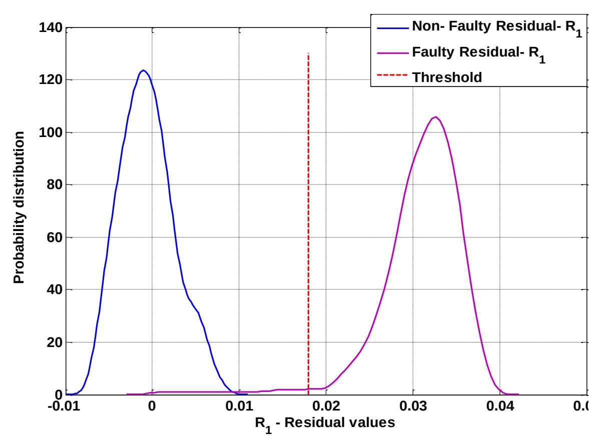

Figure 5. Threshold setting for second residual R_1

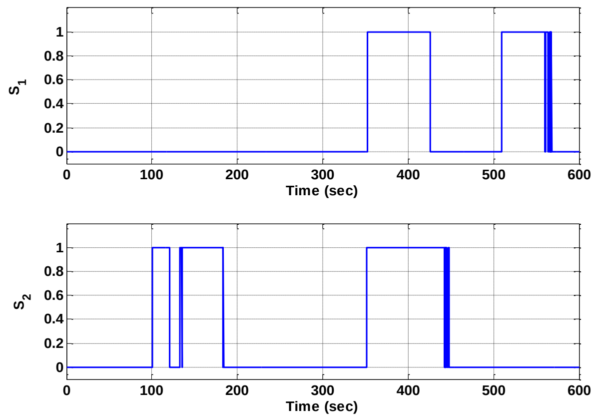

Figure 6. Fault signature in vehicle # 2

6. CONCLUSION

In this paper, a distributed fault diagnostics scheme is designed based on sliding mode observer to estimate and isolate faults on position and velocity sensors. In addition, assuming the ideal network connection between vehicles in the platoon and reconstructed faults on the sensors before data transmission, fault on the actuator can be detected for each vehicle. Future work in this research includes fault diagnostics with no-ideal network connection and packet dropout.

REFERENCES

Alam, A. Al, Gattami. A, and Johansson, K. H. (2010). An experimental study on the fuel reduction potential of heavy duty vehicle platooning. *13th Int. IEEE Conference of Intelligent Transportation System, pp. 306–311.*

Bu. F, Tan. H, and Huang. J. (2010). Design and field testing of a cooperative adaptive cruise control system. *2010 American Control Conference*, pp. 4416-4421.

Han. S, Chen. Y, Wang. L, and Abraham. A. (2013). Decentralized longitudinal tracking control for cooperatvie adaptive cruise control systems in a platoon. *2013 IEEE International Conference on Systems*, pp. 2013-2018.

Kester. L, Willigen. W, and Jongh. J. (2014). Critical headway estimation under uncertainty and non-ideal communication condition. *2014 IEEE 17th International Conference on Intelligent Transportation Systems*, pp. 320-327.

Larson. U. E, and Nilsson. D. K. (2008). Securing vehicles against cyber-attacks. *CSIIRW'08 Proceedings of the 4th annual workshop on Cyber security and information intelligence research*, pp.1-3.

Lygeros. J, Godbole. D. N, and Broucke. M. (2008). A fault tolerant control architechture for automated highway systems. *IEEE Transactions on Contorl Systems Technology*, vol. 8, No. 2, pp: 205-219.

Nilsson. D. K, and Larson. U. E. (2008). Simulated attacks on CAN buses: vehicle virus. *Proceeding of the 5th lasted Internation Conference Communication Systems and Networks*, pp.66-71.

Ploeg. J, Semsar-Kazerooni. E, Lijster. L, Wouw. N, and Nijmeijer. H. (2013). Graceful degradation of CACC performance subject to unreliable wireless communication. *16th International IEEE Annual Conference on Intelligent Transportation Systems*. Pp. 1210-1216.

Rajamani. R, and Zhu. C. (2002). Semi-autonous adaptive cruise control systems. *IEEE Transaction Vehicle Technology*, vol. 51, no. 5, pp: 1186-1192.

Zhang. Y, Gantt. G. W, Rychlinski. M, Edwards. R. M, Correia. J. J, and Wolf. C. E. (2009). *Connected vehicle diagnostics and prognostics, concept, and initial practice", IEEE Transaction Reliability*, vol. 58, No. 2, pp. 286-294.