Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 48-15 (2015) 462–469

Priority-based Multi-level Monitoring of Signal Integrity in a Distributed Powertrain Control System

Vipin Kumar Kukkala*, Thomas H. Bradley, Sudeep Pasricha*****

Colorado State University, Fort Collins, CO 80523 USA

** (e-mail: kvipin@rams.colostate.edu)*

*** (e-mail: thomas.bradley@colostate.edu)*

**** (e-mail: sudeep.pasricha@colostate.edu)*

Abstract: With the increasing numbers and importance of Electronic Control Units (ECUs) in modern automobiles, there is a need to monitor system and signal integrity to enable desired system behavior. Among the various signals in a vehicle, those associated with torque commands must be prioritized over others while preserving signal integrity as they have a large impact on meeting the powertrain performance and safety requirements. In the case of a Hybrid Electric Vehicle (HEV), there are multiple torque actuators (Motor, ICE) that need to be controlled and failure to do so can damage the powertrain. Hence in all cases, the torque requested by the driver should be the same as the sum of the outputs of all the torque actuators and within the limits, which is referred to as ensuring Torque Security (TS). In this paper, we propose a priority-based multi-level signal integrity monitoring technique in which we divide controller signals into different groups and monitor them by making use of performance counters. The proposed technique was implemented and verified using Hardware-In-the-Loop (HIL) testing as a part of the Colorado State University (CSU) EcoCAR3 advanced vehicle technology competition.

Keywords: Torque security, Powertrain control, Hybrid Electric Vehicle (HEV), Signal integrity, Powertrain Control System

1. INTRODUCTION

As the number of Electronic Control Units (ECUs) in modern vehicles increases, the in-vehicle control and communication network has become highly complex. This problem is perhaps more acute in the case of a Hybrid Electric Vehicle (HEV) as there are a greater number of electrical components such as motors, batteries, and DC-DC converters, than in conventional vehicles. In most modern HEVs and conventional vehicles, Controller Area Network (CAN) is the most widely used communication protocol because of its simplicity, low cost, noise immunity and ease of implementation. However, it suffers from low bandwidth, poor security, message delays etc. as discussed in (Kleberger et al. 2011). The typical communication system in a HEV is shown in figure 1. The driver inputs are sent to the supervisory controller via CAN, and it sends the appropriate signals to all the other local controllers via CAN messages. If there are multiple nodes connected to the same CAN channel, then the network congestion increases which leads to many issues such as signal delay, loss of signal integrity, jitter, and other failures as mentioned in (Tindell et al. 1994). All of these communication system limitations must be detected and remedied to avoid catastrophic vehicle-level malfunction.

Comprehensively monitoring the integrity of 100% of the signals in the vehicle would greatly increase the required bandwidth of the communication system. To avoid this naïve approach to signal integrity monitoring, we seek in this study to monitor signal integrity for only those signals that are of importance to vehicle safety and performance. In particular, we focus on monitoring the signal integrity of torque-related signals because of their importance in preserving the safety and performance of the vehicles.

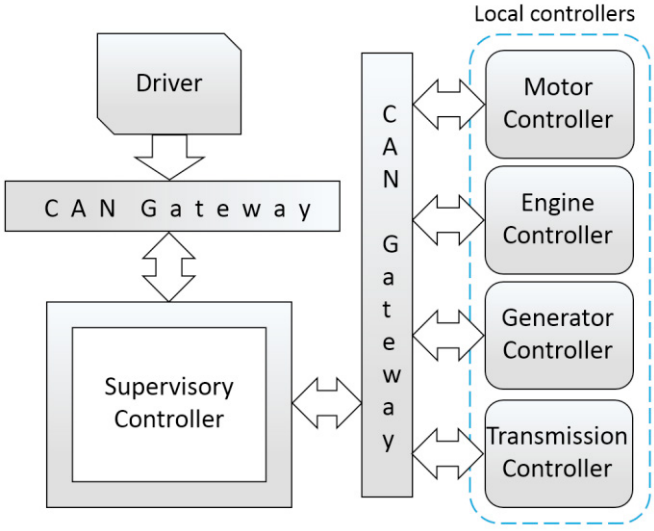


Fig. 1. Typical communication system in a HEV

10.1016/j.ifacol.2015.10.066

2. RELATED WORK

Various techniques have been proposed to protect the signal integrity in CAN and other busses. (Nilsson, D.K et al. 2008) present a delayed data authentication technique using the KASUMI encryption algorithm in CBC-MAC (Cipher Block Chaining- Message Authentication Code) mode to generate a 64-bit compound MAC for a group of four messages, which is further divided into four 16-bit MACs and each of them is stored in the 16-bit CRC (Cyclic Redundancy Check) field of the next four CAN messages. This technique assures message integrity, but suffers significant delay as the first four messages are validated by comparing the generated MAC with the received MAC in the CRC field of the next four messages. Furthermore, the delay in receiving the second group of messages adds to the overall delay, limiting this algorithm's usefulness in time-critical powertrain control applications.

(Sundaram et al. 2006) discuss different controller integrity techniques and propose asymmetric and extended asymmetric controller strategies. In these strategies an auxiliary controller is used to check the integrity of the primary controller. Some of these techniques have synchronization hardships, size and power overhead, and also encounter overhead in making changes to the existing network. (Buur. H et al. 2013) introduce a signal integrity technique in which the original signal, along with a redundant signal in encrypted form is sent in the same message. This signal is validated by comparing the original signal and decrypted redundant signal. The main limitation of this technique is that the bandwidth requirements get doubled, leading to a high bus load. Also, encrypting and decrypting signals can incur high computational overhead in the system.

The motivation for our research is to come up with a signal integrity technique that helps to improve the signal integrity in the CSU EcoCAR3 project without incurring significant overhead on the controller while trying to minimize bus traffic. In this paper we introduce a priority based, multi-level signal integrity technique in which, error tolerance for different signals are set according to the order of priority and are monitored using performance counters. The rest of the paper is organized as follows: In section 3, the proposed technique is presented in detail and in section 4 and 5, the experimental setup and results are discussed. In section 6, this method is illustrated by performing analysis on bus parameters and vehicle performance and section 7 presents our conclusions.

3. PRIORITY BASED MULTI-LEVEL SIGNAL INTEGRITY TECHNIQUE

An illustration of the proposed technique is shown in figure 2. In the first step, we divide different controller signals into different groups based on their criticality. In the next step, messages are transmitted and a handshake signal is sent from the supervisory controller to the local controller to which the torque command is sent. To monitor the signal integrity we make use of performance counters in the supervisory controller and set a threshold for the number of negative acknowledgements that command the local controller to discard the message and take appropriate remediation actions. If the number of negative acknowledgements exceed the threshold, the vehicle is moved into one of the limp modes depending on the criticality level of the failed signal, otherwise the vehicle operates normally.

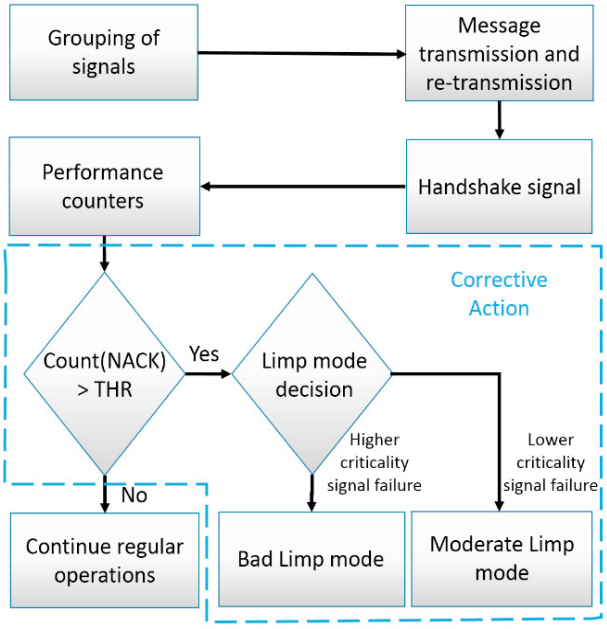


Fig. 2. Flow chart of proposed technique. (NACK = Negative Acknowledgement, THR = Threshold)

3.1 Grouping of torque related signals

In this subsection, we discuss how different torque–related signals are considered from various components in the vehicle and grouped together based on their criticality. Each group is assigned different levels of criticality with level-1 being the most critical group and level-4 being the lowest. The signals in different levels impact the vehicle performance and safety in different ways. Detection of failures in signal integrity should therefore be handled differently at each level of criticality. Thus, the limit of fault tolerance varies from level to level as discussed in the next sub-section. Table 1 shows the grouping of different controller signals and their criticality levels. Inefficient grouping of signals can have a negative impact on the safety and performance of the vehicle.

Table 1. Grouping of signals and their criticality levels

Criticality level	Signals
Level- 1	Physical brake request, Motor regen request
Level- 2	Motor torque request, Engine torque request, Acceleration pedal position sensors (A & B)
Level- 3	PRNDL, Ignition
Level- 4	Infotainment

3.2 Message Transmission with handshake signals

In this subsection, the concept of measuring and verifying signal integrity is discussed in detail. (Mandal 2003) states that in order to preserve signal integrity, it is necessary to add some redundancy to the message which inevitably increases the bandwidth required. In the signal integrity preservation technique proposed by (Buur. H et al. 2013), every CAN signal is transmitted twice, at every sample, which doubles the required bandwidth and increases delays. In our technique we propose a multi-channel CAN setup, in which there is a dedicated CAN channel for all the redundant and handshake signals labelled Control Bus (CB) and there can be one or multiple CAN channels for regular communications. When a CAN message is transmitted from the supervisory controller to the local controller via the CAN bus, the local controller re-sends the received message back to the supervisory controller via the CB. Then the supervisory controller compares the sent and received messages and sends a handshake signal to the local controller via CB, indicating faulty or not-faulty signal transmission. Performing this signal integrity monitoring and confirmation for every transmission can be expensive and bandwidth-intense. In this study, for a particular criticality level, the signal integrity monitoring and conformation is performed periodically at a rate equal to the time window of that particular level. In other words, we do not monitor all the samples of the signal, instead we perform our technique only on every n^{th} sample of the signal (where n = Time window (ms)/ Rate of transmission of original signal (ms)). The CAN network setup for our technique is shown in figure 3.

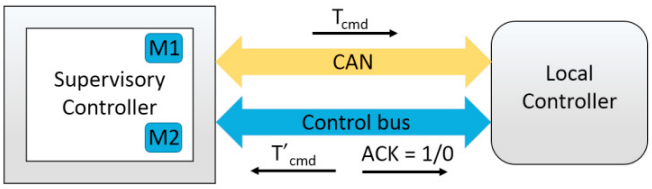


Fig. 3. CAN network setup for our proposed technique

In figure 3, the driver torque request is sent to the local controller continuously via CAN and is stored in the memory (M1) of the supervisory controller at the beginning of each time window. In the same time window the local controller immediately sends the received torque command back to the supervisory controller via CB, which is then stored at a different memory location (M2). The supervisory controller then sends out a signal ACK = 1 (positive acknowledgement) if values in both memory locations M1 and M2 match, else, it sends ACK = 0 (negative acknowledgement) via CB, which are referred to as handshake signals. Thus, ACK = 0 notifies the local controller that the received message is faulty while ACK = 1 indicates a flawless transmission. The torque command is not executed at the local controller until it encounters an ACK=1 (positive acknowledgement) signal from the supervisory controller.

3.3 Performance counters

Any erroneous transmissions are reported to the local controller via negative acknowledgement signals from the supervisory controller. The supervisory controller keeps a record of these signals by making use of different performance counters, which monitor the number of NACKs (Negative Acknowledgements) and the THR (Threshold). The values of NACK and THR are specific for each signal type. For example, a motor torque request signal has a dedicated counter MG_torque_NACK, which counts all the negative acknowledgements (ACK = 0) from the supervisory controller that are associated with the motor torque request. MG_torque_NACK increments every time a negative acknowledgement is encountered until it exceeds the threshold, MG_torque_THR. This threshold is set to different values for different criticality levels. The counter is reset only when the vehicle moves from either of the limp modes (discussed in the next subsection) to a normal operating state. This happens only when a set of positive acknowledgements are sent to the local controller over a set of time windows.

Table 2. Time window and threshold for different levels

Criticality Level	Time window (ms)	Threshold
Level- 1	150	5
Level- 2	100	10
Level- 3	2000	3
Level- 4	-	-

The time windows and thresholds (shown in table 2) for different levels of signals are chosen by taking into account both safety and bus load. For example, for signals of criticality level-2, the signal integrity check must be failed 10 times to enable a corrective action. With a time window of 100ms, the time between signal integrity failure and a corrective action is 1000 ms. The choice of the time window values is discussed in more detail in the analysis section (Section 6.1).

3.4 Corrective action

The corrective or remedial action involves taking an appropriate action once a failure in a controller is detected because of erroneous message transmissions. In developing the corrective action, the first step is to keep a track of NACK counters for all the torque related signals that are monitored. The NACK counter value is less than THR for normal operation of the controller. When this condition is violated, the corrective action mechanism is initiated. As there are different levels of signal criticalities and thresholds associated with them, there can be multiple safety modes (we label them as *limp* modes) in the vehicle. The two different types of limp modes in our technique are *moderate* limp mode and *bad* limp mode.

The vehicle enters moderate limp mode when a low criticality level signal (e.g. level-2 or level-3) loses signal integrity. The function of the moderate limp mode depends on the failure that

is detected. Consider the example of an accelerator pedal position sensor (PPS-A) value going out of range while the other sensor (PPS- B) is in range. In this case, the error is only associated with PPS-A, which is discarded and the vehicle is moved to moderate limp mode. The PPS-B signal is taken as the valid torque command signal, and only 50% of the total requested value is sent to the local controller. In contrast, when a higher criticality level signal (e.g., brake request at level-1) is faulty, the vehicle is moved to bad limp mode in which only 20% of the maximum allowable acceleration is commanded to the vehicle, so that the driver can safely pull off the road. In the case of multiple signal failures, depending on the criticality of the failed signal, one of these limp modes is chosen. In some cases, if multiple low level signals are faulty, a bad limp mode can be chosen over a moderate limp mode. This can be in the case when, for example, when both the acceleration pedal position sensor values are faulty, whereby the vehicle is moved into bad limp mode and only 20% of the maximum allowable acceleration is given to the vehicle. The supervisory controller decides which limp mode the vehicle should enter in the case of a signal integrity failure to protect the powertrain and other actuators from deterioration. Figure 4 illustrates the Stateflow logic of a level-2 signal (motor torque request) performance counter and associated remedial action.

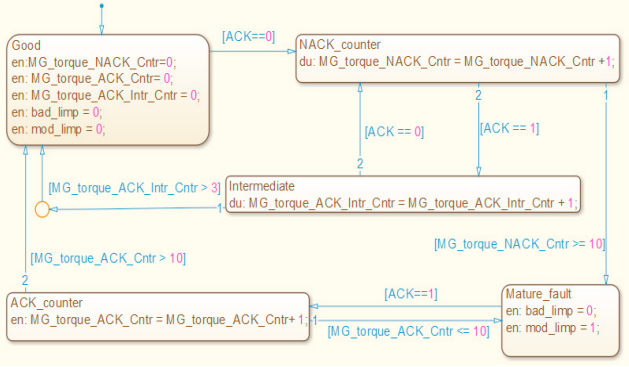


Fig. 4. Stateflow logic of a level-2 signal (motor torque request) performance counter and corrective action

In the initial Stateflow state, all the counters and the bad and moderate limp mode variables are set to zero. The vehicle remains in this state, until a NACK (ACK=0) is encountered. The controller checks for the maturity of the fault in the NACK_counter state. During this, if a series of positive acknowledgements are encountered, the vehicle goes back to normal operation state. If the fault is matured, the vehicle is moved into an appropriate limp mode. The vehicle goes back to normal operation state only after ensuring there's no erroneous transmission in the network.

4. EXPERIMENTAL SETUP

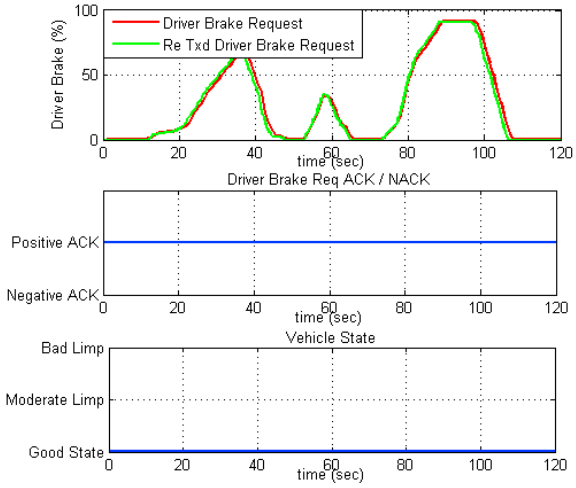
The function and performance of the proposed technique was verified using HIL testing. As a part of HIL testing, we used the Woodward Motohawk SECM-112 controller (Woodward 2015) as the supervisory controller. We adapted model based design approach to build a P2 type HEV fuel economy and powertrain control model using MATLAB/SIMULINK™. The sizing of various components such as electric motor, engine, battery etc. are determined so that they meet the EcoCAR3 competition requirements such as range of the vehicle, time for 0-60 mph, time for 50-70 mph etc. The control software was developed using Simulink and the Motohawk library and was tested using various driver inputs from hardware, including acceleration and brake pedals, PRNDL, ignition etc. Stateflow charts were used to implement performance counters and to define corrective action for various failures as shown in figure 4. Since signals of different criticality levels have different time windows, their integrity testing mechanisms are triggered at a different rate. We used the dSPACE Mid-size real time simulator (RTS) (dSPACE 2015) to run the vehicle model in real time and established two channel CAN communication between the Motohawk controller and the dSPACE RTS as illustrated in figure 3.

5. RESULTS

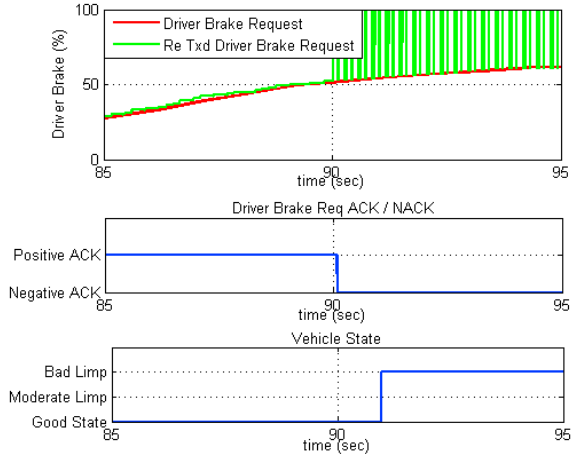
In this section we discuss the vehicle control systems' behavior with respect to the various signals subjected to the integrity test.

Figure 5(a) shows the normal operation of the vehicle with no erroneous transmissions. The first subplot in figure 5(a) shows the driver brake command issued by the supervisory controller (in red) and the re-transmitted command received from the local controller (in green). The second and the third subplots show the acknowledgements and vehicle state respectively. Since both of the signals in first subplot are equivalent all the time, there are no negative acknowledgements and hence, the vehicle remains in its normal operation state (Good State). In the next case, noise is introduced into the CAN channel and the driver command received at the local controller is therefore different from what is actually commanded. In this case, the supervisory controller sets a negative acknowledgement (NACK) as shown in figure 5(b) indicating an erroneous transmission. The NACK prevents the local controller from executing the faulty torque command. Since the mismatch persists between the requested and re-transmitted signals, the vehicle is moved to a bad limp mode. It can be seen that there is a small time interval between the vehicle going to one of the limp mode and the beginning of the series of negative acknowledgements. This indicates the buffer period during which the performance counters check for the maturity of the fault.

Similar results are obtained for the signals at different criticality levels which are shown in figures 6-9. In the case of failure of signal integrity test for level- 2 and level- 3 signals, it can be seen that the vehicle is moved into a moderate limp mode as shown in figures 7(b), 8(b) and 9 (b).

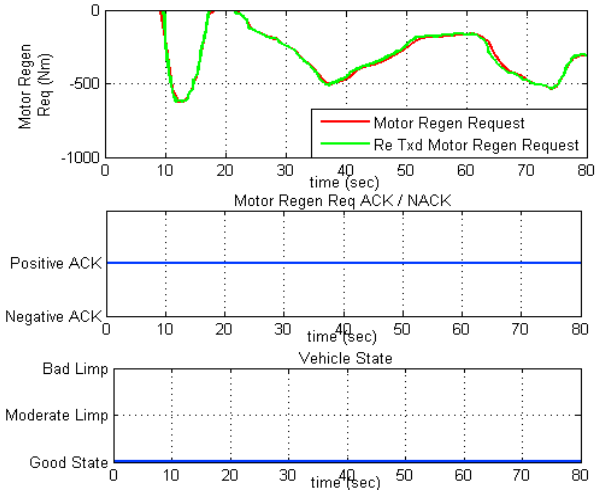


(a)

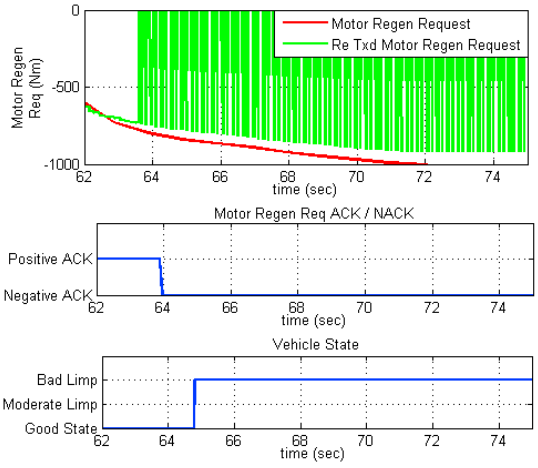


(b)

Fig. 5. Signal integrity of level-1 signal (Driver brake request) with (a) no faults in transmission; (b) noise during transmission.

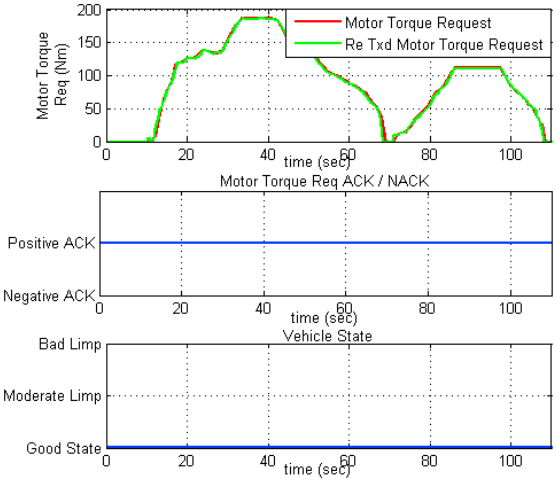


(a)

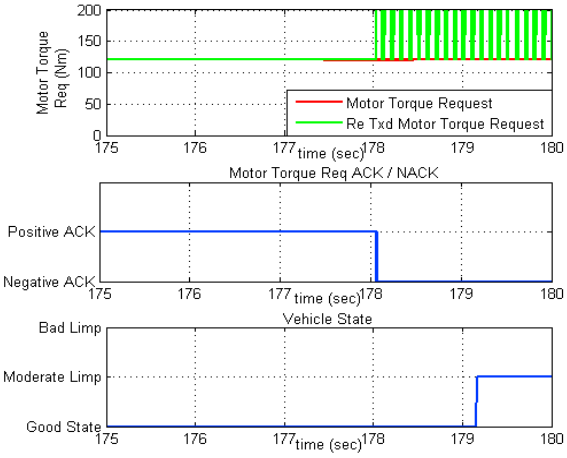


(b)

Fig. 6. Signal integrity of level-1 signal (Motor regen request) with (a) no faults in transmission; (b) noise during transmission.

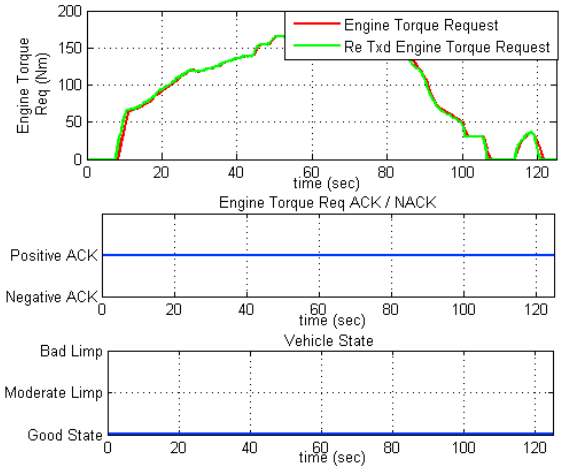


(a)

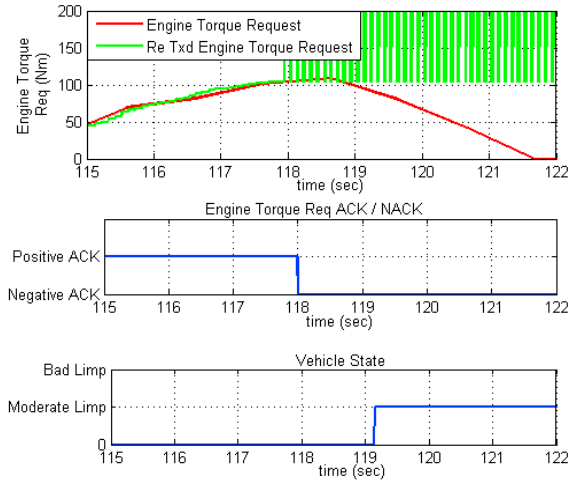


(b)

Fig. 7. Signal integrity of level-2 signal (Motor torque request) with (a) no faults in transmission; (b) noise during transmission.

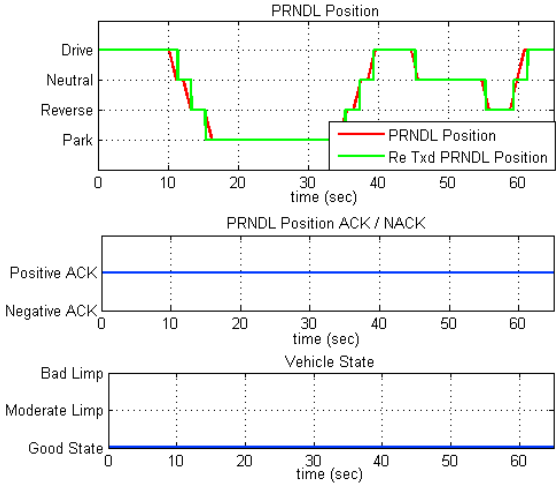


(a)

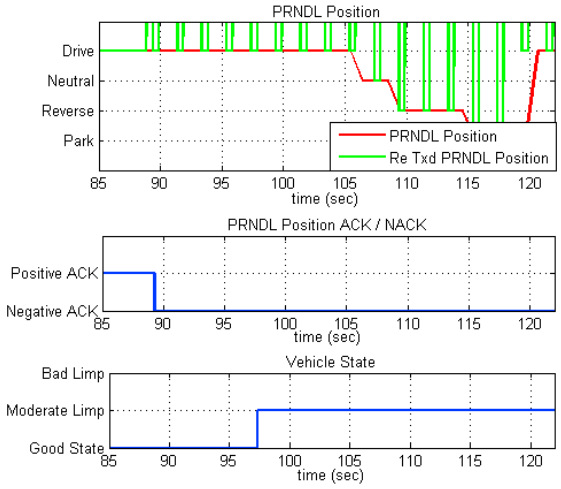


(b)

Fig. 8. Signal integrity of level-2 signal (Engine torque request) with (a) no faults in transmission; (b) noise during transmission.



(a)



(b)

Fig. 9. Signal integrity of level-3 signal (PRNDL Position) with (a) no faults in transmission; (b) noise during transmission.

By implementing our technique, the total bus load in the control bus was found to be significantly lower and is approximately 0.5% for all the signals considered.

6. PARAMETER ANALYSIS

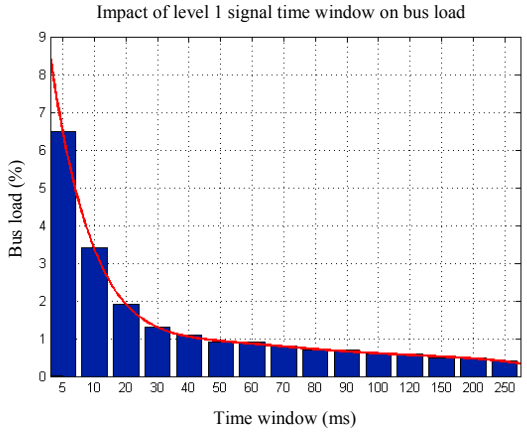
In order to determine the efficiency and overhead of our proposed technique, we carried out a detailed analysis on two key parameters as discussed below.

6.1 Impact of time window size on bus load

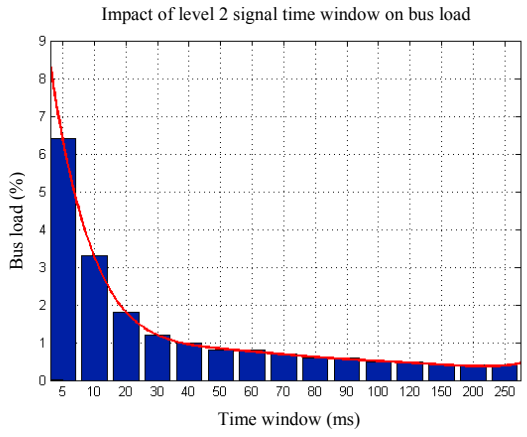
In this subsection, we present the results of how the bus load varies with different time window sizes for a particular signal level. Time window size determines the frequency of monitoring the signal. Having a higher time window size decreases the responsiveness of the vehicle to a signal integrity faulty, but a smaller time window size increases the bus load.

From the figures 10(a), 10(b) and 10(c) it is evident that bus load drops almost hyperbolically with increasing time window size. Based on these tradeoffs it is important to choose the appropriate time window for different levels. Time window for different levels are chosen such that the bus load is minimal while the vehicle is being responsive. For example, in the case of level-1, signals are transmitted every 50ms and the time window size of 150ms is chosen as the drop in bus load is insubstantial after this point (as shown in fig 10(a)) and checking every third sample makes the system sufficiently responsive to failures in signal integrity. Also, from our study on Electric Power Research Institute (EPRI) databases of various HEVs it is observed that the rate of change of the accelerator pedal is higher than the rate of change of the brake pedal, PRNDL or ignition switch and hence the transmission rate of their associated signals. Since the signals that are

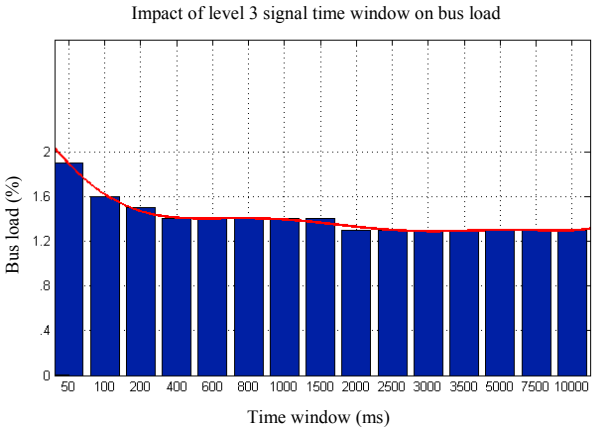
transmitted at a higher rate should be checked more often for signal integrity, level-2 has the smallest time window, followed by level-1 and level-3. Analysis on level-4 signals is not presented as our technique is not applied to signals in this lowest level of priority. Passenger comfort, Infotainment related signals fall under this level.



(a)



(b)

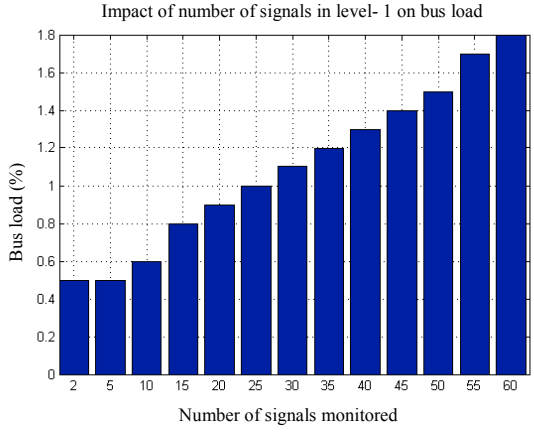


(c)

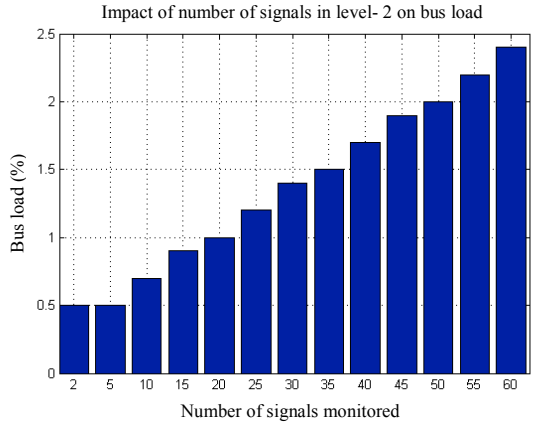
Fig. 10. Bus load of control bus for varying time windows of (a) level-1 signals; (b) level-2 signals; (c) level-3 signals.

6.2 Impact of number of monitored signals on bus load

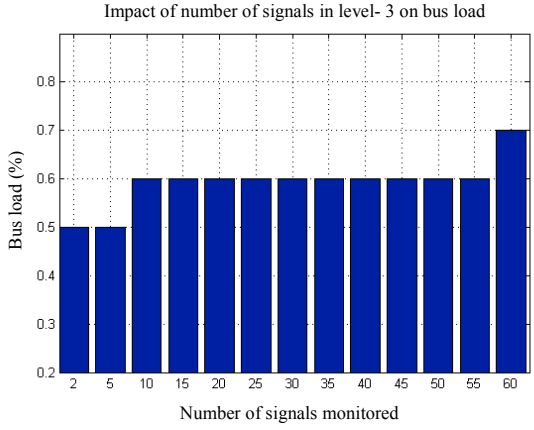
In this subsection, we discuss how the bus load is affected by the number of signals that are being monitored. This helps to understand the communication overhead of the proposed technique. The number of signals under consideration are varied for different levels and the bus load of the control bus (CB) is obtained.



(a)



(b)



(c)

Fig. 11. Bus load of control bus for different number of (a) level-1 signals; (b) level-2 signals; (c) level- 3 signals.

Figures 11(a), 11(b) and 11(c) illustrate the change in bus load as a function of the number of signals monitored in each level by maintaining the base line values of time windows for each level (shown in table 2). It can be seen that the bus load increases at a faster rate in the case of level-2 signals than level-1 and it almost negligible in level-3 signals. This is because, the time windows defined for each of these levels dictate the rate of increase in bus load with increase in number of signals in that group. From this study, it can be seen that the overhead incurred by implementing our technique is minimal and also, the proposed technique is linearly scalable with the number of signals to be monitored.

7. CONCLUSIONS

In this paper, we have proposed a priority based multi-level signal integrity monitoring and remediation technique which groups the controller signals based on their criticality and uses performance counters and handshake signals to monitor signal integrity. Our technique effectively handles different possible faults by changing the vehicle state to appropriate limp modes that ensure the safety of the vehicle. We verified our technique using HIL testing as a part of the CSU EcoCAR3 project and verified the signal integrity of various torque associated signals.

REFERENCES

Buur, Hanne, Cawthorne William R., Haines Trenton W., Park Jeong J. and Wozniak Leonard G. 2013, 'Method And Apparatus For Monitoring Software And Signal Integrity In A Distributed Control Module System For A Powertrain System'. US Patent 8428816 B2

dSPACE 2015, *dSPACE Simulator Mid-Size: Standardized, off- the- shelf HIL simulator based on DS2202 or DS2211 HIL I/O Board.* Available from <http://www.dspace.com/en/pub/home/products/hw/simulator_hardware/dspace_simulator_mid_size.cfm>. [5 June 2015]

Kleberger, P., T. Olovsson, and E. Jonsson. 'Security Aspects Of The In-Vehicle Network In The Connected Car'. *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2015. 528 - 533. Print.

Mandal, Mrinal Kr. *Multimedia Signals And Systems*. Boston: Kluwer Academic Publishers, 2003. 303 - 305. Print.

Nilsson, D. K, U. E Larson, and E. Jonsson. 'Efficient In-Vehicle Delayed Data Authentication Based On Compound Message Authentication Codes'. *Vehicular Technology Conference, 2008.* IEEE, 2008. 1 - 5. Print.

Sundaram, P., and J. D'Ambrosio. 'Controller Integrity In Automotive Failsafe System Architectures'. *SAE 2006 World Congress & Exhibition*. SAE, 2006. Print.

Tindell, K. W, H. Hansson, and A.J Wellings. 'Analysing Real-Time Communications: Controller Area Network (CAN)'. *Real-Time Systems Symposium, 1994., Proceedings*. IEEE, 1994. 259 - 263. Print.

Woodward 2015, *SECM 112.* Available from <mcs.woodward.com/support/wiki/index.php?title=SECM112>. [5 June 2015]