International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 6, No.1 (Jan-2017)

http://dx.doi.org/10.12785/ijcds/060103

Distributed Control of Multiple Plants over Embedded Network

Azeem A. Khan[1], Sajjad A. Khan[2] and Zeashan H. Khan[3]

[1,2,3]Department of Electrical Engineering, Riphah International University, Islamabad, Pakistan

Received 13 May 2016, Revised 14 Nov. 2016, Accepted 5 Dec. 2016 Published 1 Jan. 2017

Abstract: This paper describes the distributed control of multiple plants over an embedded network. The effect of network protocol on the stability and performance of a control system whose feedback loop is closed over a network. The comparison of Networked Control System (NCS) using Controller Area Network (CAN) and Ethernet (802.3) is performed while considering the system without the network as a benchmark performance scenario. The overall simulation is done using Matlab/Simulink whereas; the NCS is simulated by using TrueTime. It is found that the plant dynamics becomes slower and overshoot decreases when it is converted into NCS. As far as the stability of the system is concerned, the performance of CAN is found better than the Ethernet. Whereas, the Ethernet protocol outperforms in the case of increasing error probability on the communication bus.

Keywords: Networked control system; Controller area network; Ethernet; True-time; Mass-spring-dashpot system

---

1. INTRODUCTION

A networked control system (NCS) is a closed loop system where control related data is sent over a communication network [1]. The apparent advantage of an NCS is the reduced cost, scalability and simultaneous control of multiple plants in a network [2]. While research in the automotive control system is getting popular, it is interesting to evaluate the performance of an automotive embedded control system connected to a network [3]. Moreover, some researchers have proposed add-on blocks over TrueTime to simulate real-time embedded application-specific platforms e.g. TrueFlex real-time scheduler, which provides a flexible environment for developing and evaluating the automotive embedded systems [4]. Ever increasing number of electronic control units (ECUs) require intelligent architecture for the automotive embedded system as the technology is in the transition from manual, automatic to self-driving cars today.

Various network protocols have been used in automotive systems, For example, CAN, FlexRay, LIN, MOST etc. Ethernet is a broadcast protocol. IEEE 802.3 (CSMA/CD) standard describes physical layer and data link layer's media access control (MAC) of wired Ethernet. CAN is a message-based protocol, serial bus standard which allows communication between different controllers and devices. CAN is robust in a noisy environment and uses priority signaling. Wireless networks are however, not preferred for time-critical applications [5]. Many approaches have been adopted for the co-design problem of the control and the network part of NCS. Some researchers used the combination of Matlab for the control part with network simulators to dynamically capture the network dynamics [6-9]. In the second approach, both the network and the control part are designed using Matlab/Simulink to provide more design flexibility [8, 10-14]. Our effort belongs to the later approach. In order to study effects of Ethernet and CAN based NCS in real time, TrueTime-2.0 is selected as it is Matlab/Simulink based simulator which facilitates for multitasking in real time kernels and network transmissions with the continuous time dynamics of the controlled plant [13, 15, 16]. As the autonomous designs of cars are getting popular we aim to investigate the network architecture of the future automotive. Moreover, such platforms for embedded controller design can be used for demonstration in classroom teaching [17-19]. We considered two choices: First, the evaluation of Ethernet as a possible replacement for high data rate demand and secondly, an extended CAN network as the protocol of choice for real-time control.

This paper is organized as follows: Section 2 describes the system dynamics of the mass-spring-dashpot model representing the suspension control system of a car. Section 3 discusses the networked control system

E-mail: itsmeaak@yahoo.com, sajjadahmad38@yahoo.com, zeeshan.hameed@riphah.edu.pk

simulation using True Time Matlab toolbox. Effect of interference, data rate and loss probability is discussed in section 4 while section 5 concludes the paper.

2. SYSTEM DYNAMICS

The considered second order mass-spring-dashpot system mounted on a cart mimics the car's suspension model without friction as shown in Figure 1. For time t < 0 the cart and the allied system is considered at rest. At t = 0, a force input u(t) causes displacement of the massless cart due to which it moves with uniform speed resulting in a displaced position y(t). Here, we assumed the following parameters for the model:

m = 10 kg

b = 0.2 Ns/m

k = 10 N/m

The cart is massless. On displacing the cart, the mass moves with an oscillatory response. Here, we design a PID controller using Matlab to fulfill the requirement such as:

a. Settling Time: not greater than 2.0 sec
b. Percentage Over Shoot: not greater than 10%

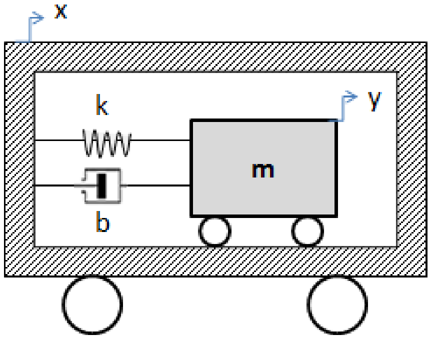


Figure 1. A System Mounted on a Cart

The state space representation of the continuous time (CT) system [2] is given by:

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du \qquad (1)$$

Where, A = [0 1; -1 -0.02]
B = [0.02 -0.996]´
C = [1 0]
D = [0]

and the continuous time (CT) transfer function of open loop system is:

$$G_{OL} = \frac{b*s+k}{m*s^2+b*s+k} \qquad (2)$$

The second-order dynamic model is evaluated in this paper for the continuous time, discrete time and with a network in the loop configuration for comparison purposes.

A. System Analysis

The continous time open loop step response of the system is as shown in Figure 2.

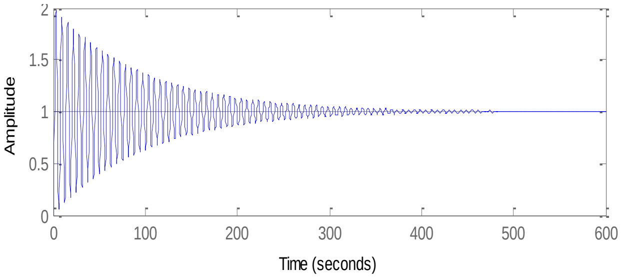


Figure 2. Open loop step response

B. PID Controller

After implementing the PID controller having gains $K_p = 150$, $K_i = 200$ and $K_d = 28.5$, the CT closed loop step response of the system is as shown in Figure 3.

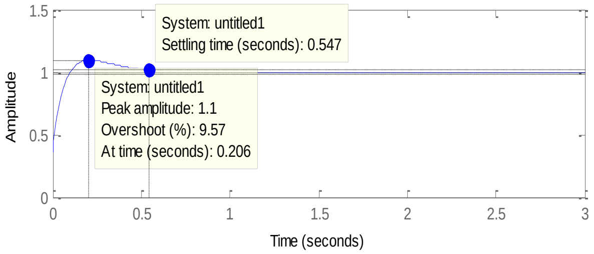


Figure 3. Closed loop step response of the mass-spring-dashpot system

The settling time is 0.547s and the overshoot is 9.57% which is within the range of desired performance parameters.

C. Networked PID Controller

To study the stability of the system, we designed a PID controller using Matlab/Simulink. After a number of iterations the PID controller gains were selected. By using Truetime-2.0 two controllers based on a single network for two identical plants having different sampling periods are designed. The network comprises of five nodes namely:

- Interfering Node (1) having highest priority 1
- Sensor/Actuator Node (2) for plant 1 having priority 2
- Controller Node (3) for plant 1 having priority 3
- Sensor/Actuator Node (4) for plant 2 having priority 4
- Controller Node (5) for plant 2 having lower priority 5

From different forms of the transfer function of PID controller, by comparison, we get K, Ti and Td.

$$C(s) = K + \frac{K}{T_i s} + KT_d s \qquad \text{and}$$

$$C(s) = K_p + \frac{K_i}{s} + K_d s \qquad (3)$$

As the controller is a continuous time dynamical system so to implement it using a computer, the continuous time system has to be approximated by a discrete time system [20]. The signal u is the sum of the proportional, integral and derivative terms as follows:

$$u = P + I + D \qquad (4)$$

where,

$$\begin{aligned} P &= K[\beta r - y(t_k)] \\ I &= I(t_k - 1) + K.h/Ti[r - y] \\ D &= ad.D(t_k - 1) + bd[y(t_k - 1) - y] \end{aligned} \qquad (5)$$

where, t_k denotes the sampling instants, i.e., the times when the computer reads its input and ad, bd terms can be calculated as below [21]:

$$\begin{aligned} ad &= Td/[N.h + T_d] \\ bd &= N.K.ad \end{aligned} \qquad (6)$$

Pre-computation of the coefficients ad and bd saves computation time in the main loop of the embedded controller. These calculations have to be done only when controller parameters are changed. The main loop is executed once every sampling period. The above-mentioned equations (5-6) are used in Matlab to design the controllers for our plants.

During designing PID controllers in Matlab/Simulink and the network simulation in True Time-2.0, we selected the sampling period of 0.01s for controller1 of plant1 and 0.02s for controller2 of plant2. Both controller 1 and 2 are simulated separately but shown as a single controller node in Fig. 5. The plant2 is exactly identical to plant1. A single network is used having network type CAN and then Ethernet in the second scenario. The network schedule is important to observe so that to verify if all the deadlines are met within the sampling period [22]. We varied certain network parameters with one parameter changing at a time and evaluated the effect of this change on the stability of the networked control system. These parameters are:

- Loss probability
- Data rate
- Frame size
- Bandwidth occupancy of the interfering node

The Simulink block diagram of NCS is shown in Figure 4 below.

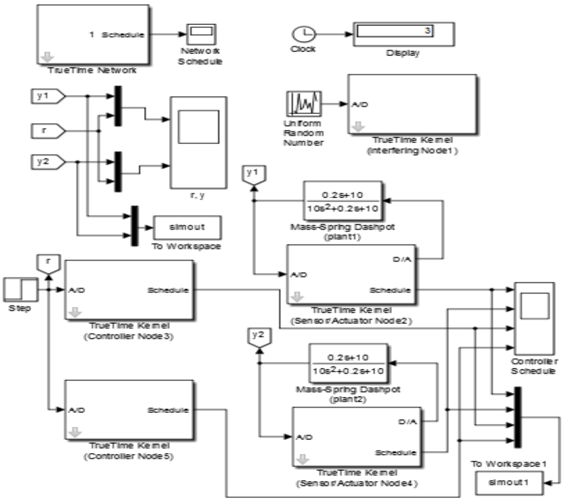


Figure 4. Block Diagram of NCS

To simplify the simulation parameters, the sensor dynamics are taken as a unity gain. Two plants effectively depict the dynamics of a partial car model with only two suspension control systems. However, a more complex model can consist of four wheel model for a full-scale simulation. Our aim is to test this control system on a network with varying parameters to simulate its effect on over all closed loop system's stability and performance.

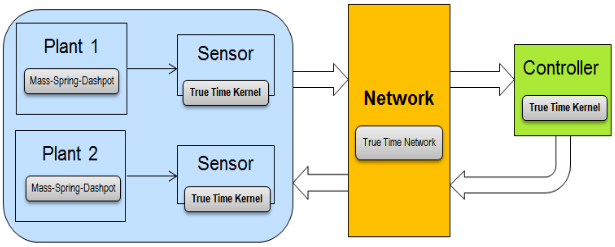


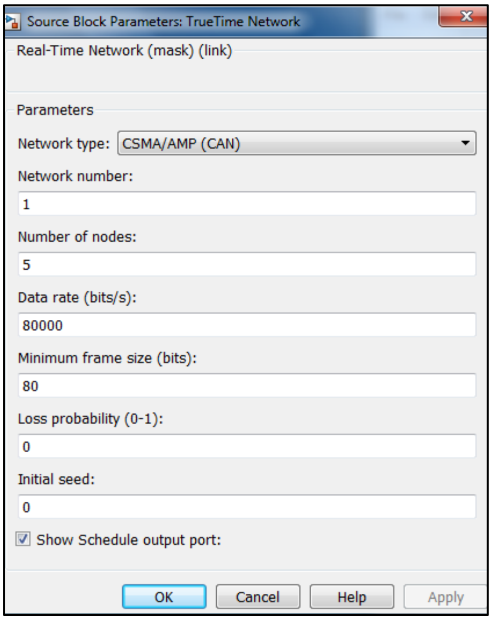
Figure 5. Generalized structure of NCS simulation in TrueTime

3. SIMULATIONS AND RESULTS

Simulations are run by comparing two different network types namely Controller Area Network (CAN) and Ethernet to evaluate the control performance as shown in the Figure 5.

3.1 Controller Area Network

CAN is being used as a standard protocol in automotive communication applications since last two decades. Network parameter is set in source block as shown in Figure 6 and the bandwidth share of interfering node is set as 20%. The data rate is set as 80 Kbps whereas the maximum data rate of 1 Mbps can be achieved on CAN. The minimum frame size of 80 bits is selected for the communication.



Source Block Parameters: TrueTime Network

Real-Time Network (mask) (link)

Parameters

Network type: CSMA/AMP (CAN)

Network number: 1

Number of nodes: 5

Data rate (bits/s): 80000

Minimum frame size (bits): 80

Loss probability (0-1): 0

Initial seed: 0

☑ Show Schedule output port:

Figure 6. CAN network parameters for simulation in True Time 2.0

After running the simulation, we got the results satisfying the design criterion as shown in Figure 6. The step response shows that now the settling times of plant1 and plant2 are 1.967s and 1.993 respectively whereas overshoot is 4.1%. Introducing the network seems to introduce delay and damping due to additional dynamics as seen in Figure 7.

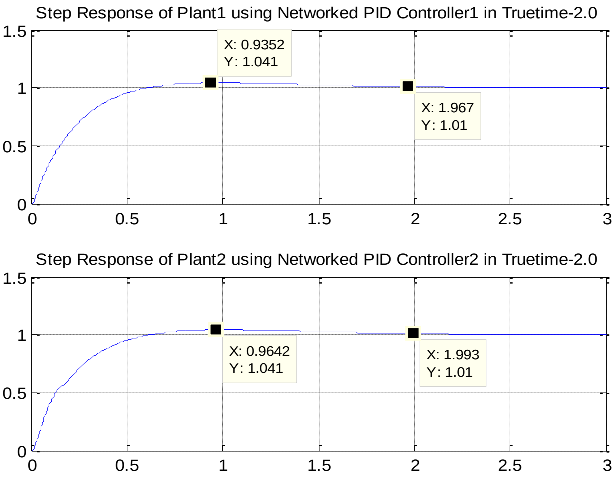


Figure 7. Closed loop step response using CAN setting in Fig. 6

The scheduling plot for plant1's controller (Cont1), the sensor (Sensor1) and an actuator (Act1) and for plant2's controller (Cont2), the sensor (Sensor2) and an actuator (Act2) is as shown in Figure 8. It is important to note that the loop starts from the sensor data as sent periodically on the network. As the controller 1 is having higher priority as compared to controller 2, the scheduler completes the higher priority first using the CSMA/AMP protocol.

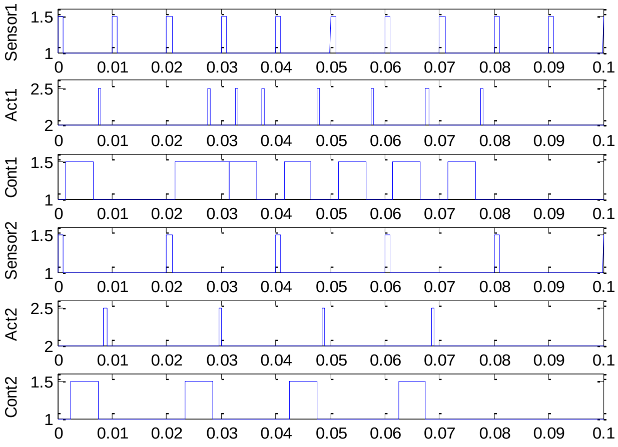


Figure 8. Scheduling Plot using CAN settings of Fig. 6

Now, we see the effect on stability of the system by changing different network parameters. One parameter change at a time strategy is followed to accurately capture the effect on the performance.

A. Loss Probability

It is observed that up to a maximum loss probability of 20%, the response of both plants is stable. At a loss probability of 30%, the closed loop step response is shown in Figure 9, where it is evident that the plant 2 is showing deteriorated performance as compared to plant 1 due to the fact that it is having a lower priority. This shows that the effective node numbers on CAN network are important as it determines the priority of the node.

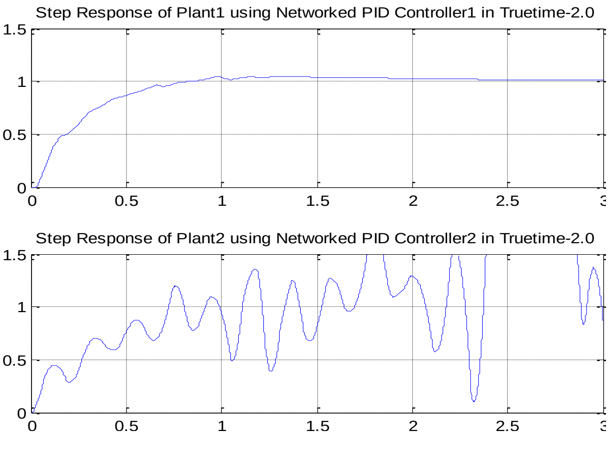


Figure 9. Closed loop step response with 30% loss probability

B. Data Rate

It is observed that up to the minimum data rate 49kbps, the response of both plants is stable. At a data rate of 45kbps, the closed loop step response is as shown in Figure 10. It shows that the plant 2 having lower priority gets unstable due to the reduced data rate as the delay increases.

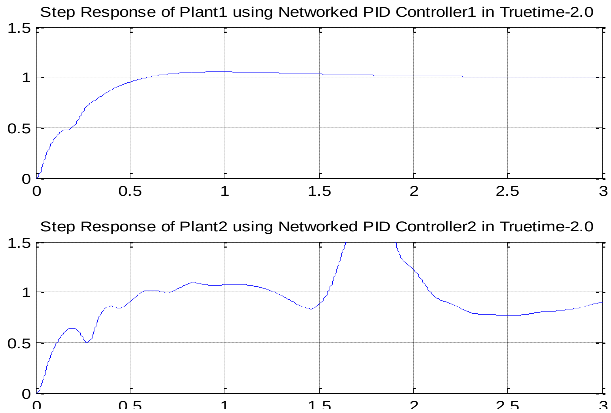


Figure 10. Closed loop step response at a data rate of 45kbps

Moreover, from the digital control theory at least 5 to 7 samples in the rise time are mandatory for stable closed loop response. So, an optimal data rate will ensure this criteria resulting in an improved step response. On the other hand, if enough samples are not available, despite the controller effort, the stability of the system will be compromised. The priority of the second plant reduces the effective data rate of the network protocol.

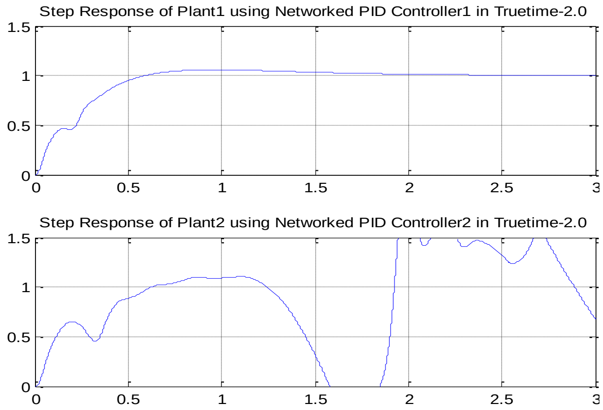


Figure 11. Closed loop step response with frame size of 150 bits

C. Frame Size

It is observed that up to a maximum frame size of 133 bits, the response of both the plants is stable. At a frame size of 150 bits, the closed loop step response is destabilized again for the plant 2 with less priority as shown in Figure 11. It is important to mention that the frame size plays a vital role in determining latency in the real time control. An optimal frame size results in a minimum delay which improves the phase margin thus ensuring stability of the dynamic system.

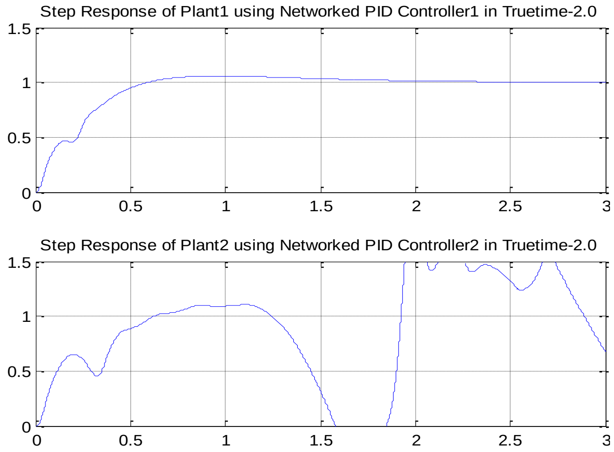
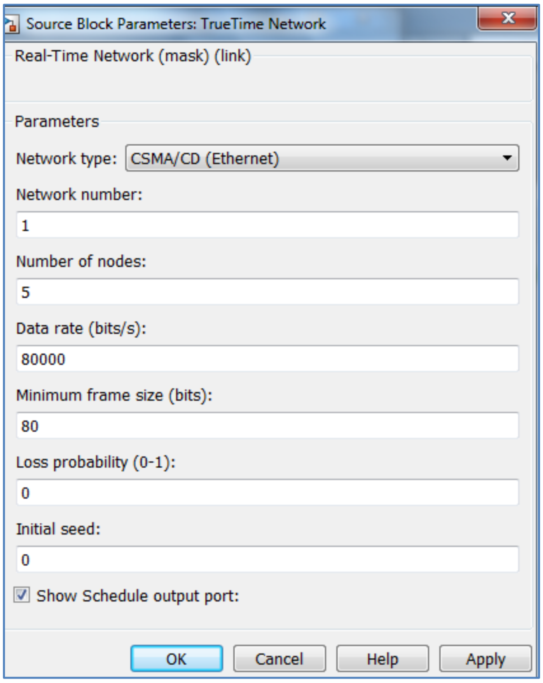


Figure 12. Closed loop step response with 60% bandwidth share for interfering node

D. Bandwith Share of Interfering Node

The closed loop response also depends on the bandwidth assignment to the controller and the sensor. In the case of congestion as simulated by an interfering node, it is observed that up to a maximum bandwidth share of interfering node equal to 50%, the response of both plants is stable. At a higher bandwidth share e.g. at 60% of the closed loop, step response gets unstable as shown in Figure 12.


Source Block Parameters: TrueTime Network

Real-Time Network (mask) (link)

Parameters

Network type: CSMA/CD (Ethernet)

Network number: 1

Number of nodes: 5

Data rate (bits/s): 80000

Minimum frame size (bits): 80

Loss probability (0-1): 0

Initial seed: 0

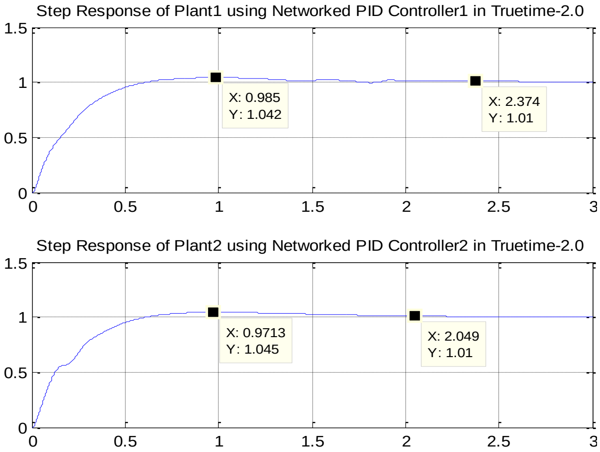
☑ Show Schedule output port:

Figure 13. Network Parameters for Ethernet (CSMA/CD) simulation in True Time 2.0

3.2 Ethernet

In the second attempt, an Ethernet protocol based on CSMA/CD is used to simulate NCS. Network parameters are set in network block of True Time 2.0 as shown in Figure 13 and the bandwidth share of interfering node is set as 20%. A data rate of 80 Kbps is selected with the minimum frame size of 80 bits. These parameters are the same as those selected for CAN in the first case.

After running the simulation, we got the results as shown in Figure 14 satisfying the design requirement concerning overshoot which is 4.2% and 4.5% for plant1 and plant2 respectively; whereas a slightly higher settling time has been achieved for both plants i.e. 2.374s and 2.049s respectively. This explains the reason as the Ethernet frame has a greater size as compared to the CAN introducing more delay in the control loop.


Step Response of Plant1 using Networked PID Controller1 in Truetime-2.0

Step Response of Plant2 using Networked PID Controller2 in Truetime-2.0

Figure 14. Closed loop step response using Ethernet protocol

The scheduling plot for plant1's controller (Cont1), the sensor (Sensor1) and an actuator (Act1) and for plant2's controller (Cont2), the sensor (Sensor2) and an actuator (Act2) is as shown in Figure 15.

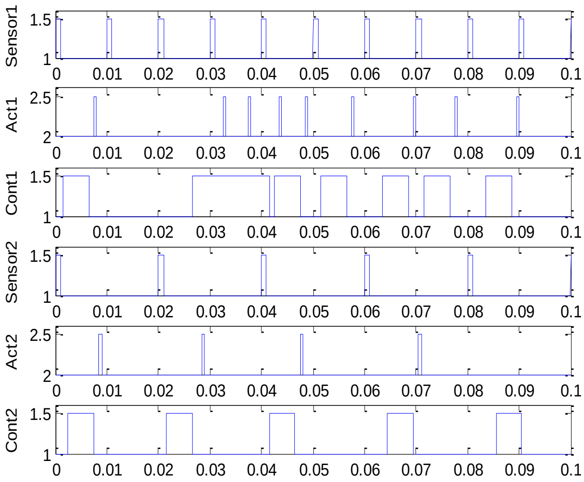
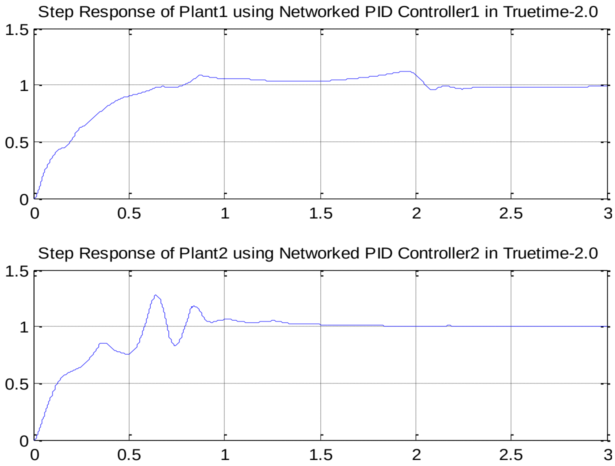


Figure 15. Scheduling plots with Ethernet protocol

Now, we see the effect on the stability of the system by changing different parameters of the Ethernet.

A. Loss Probability

The networked control system is simulated for varying loss probability of the data over the network and it has been observed that up to a maximum loss probability of 70%, the response of both plants is stable. At a loss probability of 75%, the closed loop step response is as shown in Figure 16.

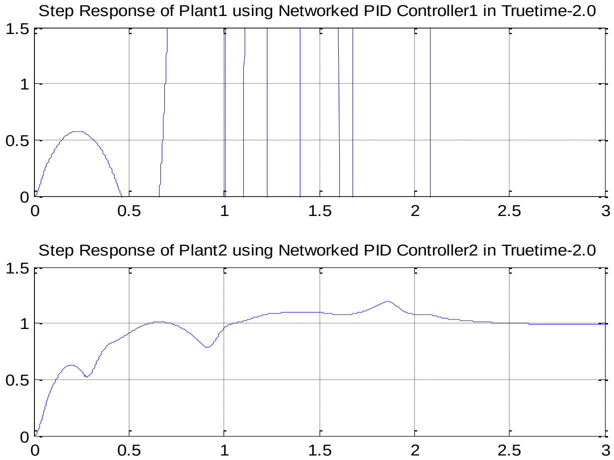

Step Response of Plant1 using Networked PID Controller1 in Truetime-2.0

Step Response of Plant2 using Networked PID Controller2 in Truetime-2.0

Figure 16. Closed loop step response with 75% loss possibility

B. Data Rate

Network data rate is an important parameter and it can be varied in the simulation. It is observed that up to minimum data rate 75 kbps the response of both plants is stable. At a data rate of 74 kbps, the closed loop step response is as shown in Figure 17.

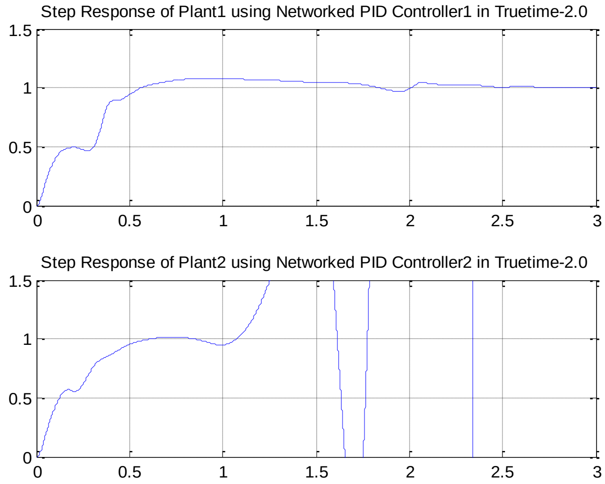

Step Response of Plant1 using Networked PID Controller1 in Truetime-2.0

Step Response of Plant2 using Networked PID Controller2 in Truetime-2.0

Figure 17. Closed loop step response on Ethernet with data rate of 74 kbps

C. Frame Size

It is observed that up to the maximum frame size 90 bits, the response of both the plants is stable. For the fixed data rate and a frame size of 100 bits, the closed loop step response is shown in Figure 18.

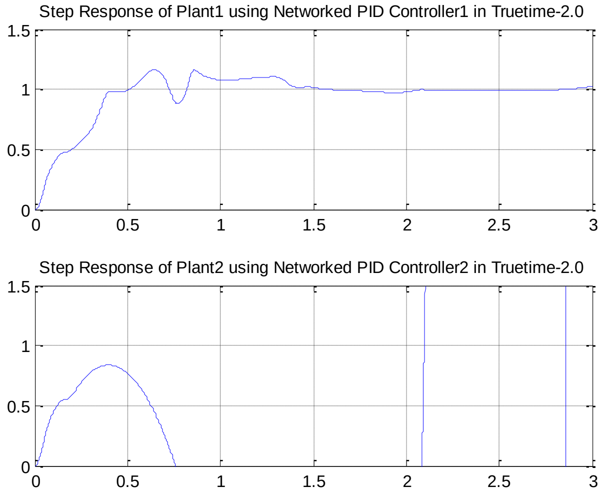

Step Response of Plant1 using Networked PID Controller1 in Truetime-2.0

Step Response of Plant2 using Networked PID Controller2 in Truetime-2.0

Figure 18. Closed loop step response with frame size of 100 bits

D. Bandwith Share of Interfering Node

It is observed that up to a maximum bandwidth share of interfering node of 20%, the response of both the plants is stable. At a bandwidth share of 30%, the closed loop step response is as shown in Figure 19.


Step Response of Plant1 using Networked PID Controller1 in Truetime-2.0

Step Response of Plant2 using Networked PID Controller2 in Truetime-2.0

Figure 19. Closed loop step response with 30% bandwidth share of interferring node

4. COMPARISON OF CAN AND ETHERNET

The comparison of controller's design requirements, Control without the network as well as with network is shown in Table 1 and the effect of modified parameters of NCS (CAN vs. Ethernet) on the stability of closed loop system is compared in Table 1 which shows the effect of the networked control system on controller design.

TABLE I. CAN VS. ETHERNET SIMULATION RESULTS

Parameters Ensuring Stability	**CAN**	**Ethernet**
Maximum loss probability (%)	20	70
Minimum Datarate (kbps)	49	75
Minimum Frame Size (bits)	133	90
Maximum Bandwidth Share of Interfering Node (%)	50	20

TABLE II. COMPARISON OF A SIMPLE CONTROL SYSTEM VS. NETWORKED CONTROL SYSTEM

Parameters	**Design Criterion**	**Plant w/o Network**	**Network Protocol in NCS**	
			CAN	*Ethernet*
Overshoot (%)	< 10	9.57	4.10 4.10	4.20 4.50
Setteling Time (sec)	< 2.0	0.547	1.967 1.993	2.374 2.049

As depicted in the simulation results, CAN is a better selection for low data rate network based controllers. We simulated the embedded control system using TrueTime so that we can evaluate its performance. Using the comparison data, it is clear that the Ethernet is able to sustain more loss probability of the messages as compared to CAN. Also, it offers high data rate but lower robustness in case of interfering node on the network. Table 2 shows the performance comparison of the system with and without networked control. It is apparent that by introducing a network within the control loop, the speed of response of the closed loop system reduces due to additional damping added by the network. Thus, a network control system's time response is slower as compared to its equivalent system without a network. The results also indicate that the effect of network protocol is such that it modifies the closed loop response. In two iterations, the overshoot and settling time for the case of a system without network and with the network are compared. It seems that the effect of Ethernet is more pronounced in adding an additional delay in the closed loop system as compared to CAN. The advantage of a lightweight protocol for automotive control is thus justified. However, for the case of error tolerance in scenarios of maximum loss probability, it has been observed that the Ethernet performs well up to 70% loss probability due to retransmission after error detection. Therefore, it has been deduced that keeping in view the more complex embedded architectures, multiple CAN networks can solve the real time issues with error prone strategies in case of data loss to fill the communication gap between the semi-autonomous to fully self-driving vehicles.

5. CONCLUSION

This paper describes a distributed embedded control of multiple plants in the presence of interfering node connected with the network. The problem posed in the design of embedded control system is the choice of the network protocol which can offer the best performance while ensuring stability and robustness. The results have shown that the CAN is performing better for the distributed control as compared to the Ethernet protocol. In future, we aim to develop a benchmark and implement a networked embedded control system for the automotive applications using wired and wireless protocols.

REFERENCES

[1] T. C. Yang, "Networked control system: a brief survey," *IEE Proceedings Control Theory and Applications*, vol. 153, p. 403, 2006.

[2] Z. H. Khan, D. G. Catalot, and J. M. Thiriet, "Hierarchical wireless network architecture for distributed applications," in *Wireless and Mobile Communications, 2009. ICWMC'09. Fifth International Conference on*, 2009, pp. 70-75.

[3] M. B. N. Shah, A. R. Husain, and A. S. A. Dahalan, "An Analysis of CAN Performance in Active Suspension Control System for Vehicle," *a− a,* vol. 4, p. 2, 2012.

[4] S. A. Marashi and A. Jahanian, "TrueFlex: A flexible and efficient evaluation platform for networked automotive systems," in *Electrical Engineering (ICEE), 20th Iranian Conference on*, 2012, pp. 645-650.

[5] S. A. Mustafa, "Interference Estimation and Mitigation in Wireless Networks," *International Journal of Computing and Digital Systems* vol. 4, pp. 233-243, 2015.

[6] J. Matsumura, Y. Matsubara, H. Takada, M. Oi, M. Toyoshima, and A. Iwai, "A simulation environment based on omnet++ for automotive can–ethernet networks," *Analysis Tools and Methodologies for Embedded and Real-time Systems,* p. 1, 2013.

[7] M. Morelli, F. Cremona, and M. Di Natale, "A system-level framework for the evaluation of the performance cost of scheduling and communication delays in control systems," in *5th International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems*, 2014, p. 13.

[8] Z. Zhang, E. Eyisi, X. Koutsoukos, J. Porter, G. Karsai, and J. Sztipanovits, "A co-simulation framework for design of time-triggered automotive cyber physical systems," *Simulation Modelling Practice and Theory,* vol. 43, pp. 16-33, 2014.

[9] Y. Zhou, J. Baras, and S. Wang, "Hardware Software Co-design for Automotive CPS using Architecture Analysis and Design Language," *arXiv preprint arXiv:1603.05069,* 2016.

[10] A. Cervin, M. Ohlin, and D. Henriksson, "Simulation of networked control systems using TrueTime," in *Proc. 3rd International Workshop on Networked Control Systems: Tolerant to Faults*, 2007.

[11] D. O. Da Penha and G. Weiss, "Integrated timing analysis in the model-driven design of automotive systems," *Proceedings of NiM-ALP*, p. 40, 2013.

[12] D. Henriksson, A. Cervin, and K.-E. Årzén, "TrueTime: Simulation of control loops under shared computer resources," in *Proceedings of the 15th IFAC World Congress on Automatic Control*, 2002.

[13] D. Henriksson, A. Cervin, and K.-E. Årzén, "TrueTime: Real-time control system simulation with MATLAB/Simulink," in *Proceedings of the Nordic MATLAB Conference*, 2003.

[14] A. Mechraoui, Z. H. Khan, J.-M. Thiriet, and S. Gentil, "Co-design for wireless networked control of an intelligent mobile robot," in *International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, 2009, p. 7.

[15] M. Ohlin, D. Henriksson, and A. Cervin, "TrueTime 1.5 Reference Manual," *Department of Automatic Control, Lund University, Sweden,* 2007.

[16] M. Sharma;, H. Elmiligi;, and F. Gebali, "Performance Evaluation of Real-Time Systems," *International Journal of Computing and Digital Systems,* vol. 4, pp. 43-52, 2015.

[17] G. Farias, K.-E. Årzén, A. Cervin, S. Dormido, and F. Esquembre, "Teaching embedded control systems," *International Journal of Engineering Education,* vol. 26, p. 938, 2010.

[18] M. Ghazal;, R. Hamouda;, and S. Ali, "A Smart Mobile System for the Real-Time Tracking and Management of Service Queues " *International Journal of Computing and Digital Systems,* vol. 5, pp. 305-313, 2016.

[19] M. Ghazal, R. Hamouda, and S. Ali, "A Smart Mobile System for the Real-Time Tracking and Management of Service Queues," *Int. J. Com. Dig. Sys, http://dx.doi. org/10.12785/ijcds /050402,* vol. 5, p. 4, 2016.

[20] A. Alqudah; and A. Qasaimeh, "Design Techniques Study of PID Controllers," *International Journal of Computing and Digital Systems,* vol. 4, pp. 101-109, 2015.

[21] K. J. Åström and T. Hägglund, *Advanced PID control*: ISA-The Instrumentation, Systems, and Automation Society; Research Triangle Park, NC 27709, 2006.

[22] H.-J. Cha, W.-H. Jeong, and J.-C. Kim, "Control-Scheduling Codesign Exploiting Trade-Off between Task Periods and Deadlines," *Mobile Information Systems,* 2016.

Azeem Ahmad Khan obtained his MSc from Punjab University in 2010 and MS (Electronics) in 2016 from Riphah International University, Islamabad. His research interests include modeling & simulation, embedded control and diagnosis.

Sajjad Ahmad Khan obtained his MSc in Physics from Punjab University in 2009 and MS (Electronics) in 2016 from Riphah International University, Islamabad. His research interests include real time embedded control, Digital design and computer controlled systems.

Zeashan Khan is presently working as an Assistant Professor and head of control and robotics research group at Riphah International University, Islamabad, Pakistan. He obtained his Ph.D. in Electrical Engineering from University of Grenoble, France in 2010. He has published about 20 conference papers, 8 journal papers and two books so far. His research interests include fault tolerant control, computational intelligence and embedded control systems. He is a member IEEE and PEC.