PERINTIS eJournal

PERINTIS eJournal, **2021**, Vol. 11, No. 1, pp. 40-60

ANALYSIS IN A NETWORKED CONTROL SYSTEM OF A SERVO MOTOR USING MATLAB TRUETIME TOOLBOX 2.0

Wasi Hyder[1], Ali Jafari[1], Ahmad Athif Mohd Faudzi[1,2], Luis Ferreira[1], Norr Iswaniza Md Siam[1], Nurul Adilla Mohd Subha[1]

[1]School of Electrical Engineering, Faculty of Engineering, Universiti Teknologi Malaysia, 81310 Johor Bahru, Johor, Malaysia
[2]Centre for Artificial Intelligence and Robotics (CAIRO UTM), Universiti Teknologi Malaysia, 54100 Kuala Lumpur, Malaysia

* Corresponding author: athif@utm.my

ABSTRACT

This paper proposes a comparative analysis to understand the effects of Networks and Scheduling techniques on Network Control System. The main contribution of this work is to see effects on stability of the control system by using Ethernet and Controller Area Network. And then further analyze it with creating variation via scheduling methods. This is achieved by using an example of network-based servo control system from the Matlab TrueTime toolbox 2.0. This model is designed in Simulink and each system is initialized using m.file in Matlab. The model used in this work consists of three network nodes; an interference generator, a plant (Servo motor with an actuator and a sensor) and a controller, all connected by a network. Four different case studies were developed and discussed such as Ethernet, Controller Area Network, Scheduling, Scheduling on control node. It helped to establish the superiority of the Controller Area Network, effects on data loss and use of right scheduled algorithm for stable real time control system. This comparative study will help a control engineer to decide a precise parameter, while designing a Real time network control system.

Keywords: *Real Time Control System; Scheduling; Network Control System; Ethernet; Controller Area Network*

1.0 INTRODUCTION

Real time control system (RTC) is a criterion of control operation to be fulfilled by industrial systems. In real-time control system the performance of the system does not depend only on the logical results but the time it is delivered is the important aspect and can cause the failure of the system if it is not considered [1,2]. RTC are the tasks or processes in computerized control systems in industry or devices that satisfy three criteria of reliable operation execution, set operation deadline and predictable operation results. One of the important applications of RTC in real world is the mobile robots and Autonomous Ground or Air vehicles. Real time response is mandatory in the execution of operation special gesture controller mobile robots [3]. The networked controlled system also has ideal application in the field of industrial sensor application especially non-invasive techniques [4]. On the other hand, industry is the bread and butter of a mankind. Actuators have very important role in the industry for executing multiple tasks [5]. Industrial Pneumatic actuators are highly nonlinear characteristics and uncertainties make it difficult to achieve high performances. Different control strategy is applied in a combination with a modified PID controller to a pneumatic pusher mechanism to achieve real time response [6].

Time deadline is very important for RTC systems. Missing the deadlines can cause fatal errors or undesirable outputs, it is considered as a hard real-time operation. In contrast, if finishing the tasks on time is desirable but not essential, it is a soft real-time operation [7]. True-Time 2.0 simulator is not an official toolbox for Matlab and still beta version. Its has been developed at Lund University since 1999. True-Time 2.0 helps to verify the analysis of control system under influence of different scheduling schemes, task execution, network delay, dropout, and sampling time. A Network Control System (NCS) is a closed loop control system that uses a network to communicate between the elements. The four basic elements in a network control system are sensors, controller, actuator and communication network. NCS enables executing of multiple tasks from long distance. It also reduces the hard wiring resulting in less complexity of the system and lower the implementation cost. Modification, adding elements such as sensors and actuators to the system will be easier using the NCS [8]. Different network types are used in control systems. In this work we focus on two of them, the Ethernet bus, with carrier sense multiple access with collision detection (CSMA/CD) and Controller Area Network (CAN) bus [9].

The CSMA/CD mechanism is used in Ethernet. The CSMA/CD protocols are based on the IEEE 802.3 network standard [9]. The advantage of Ethernet is its simple algorithms in network operation so there will be no delay on the low network loads [10]. Ethernet does not support message prioritization. When there is a high network load message, collision can affect data through put and time delay which can be unbounded [10-11]. Because of the required minimum valid frame size, Ethernet uses a large message size to transmit a small amount of data.

A Controller Area Network (CAN) is a serial protocol mainly used in automotive industry, but it can offer a good performance in other industrial applications. It is designed for short messages and uses a CSMA/Arbitration on message priority (CSMA/AMP) medium access method [10]. The message priority is determined for a CAN network. Higher priority messages always gain access to the medium so no delay for them can be guaranteed. The main disadvantage of CAN compared to other networks is its slow data rate that in maximum is 500 Kb/s [12]. Scheduling is the function of selecting the task, which will be done by processor as next. Special algorithms are required to schedule a set of tasks. There are two main classes of scheduling algorithms of Static Scheduling Algorithms and Dynamic Scheduling Algorithms [13]. The complete information about the number of tasks, deadlines, priorities, periods, etc. are required for a static scheduling. The scheduling should be designed, and the problem should be solved prior to the schedule is executed. This class is also called clairvoyant.

The dynamic scheduling is the case in which, changes in the configuration can be done at run time and feasibility can be determined. Off-line planning is a must for static schedules while the dynamic scheduling can be planed either offline with online implementation or online if the future is unknown or will be changed. The effectiveness of a real-time scheduling algorithm can be determined by its ability to meet all the deadlines. The system is considered overloaded if any deadline will be missed [6-9].

2.0 SYSTEM OVERVIEW

The system analyzed in this study is an example given by the True-time toolbox, called 'Network'. The example is illustrated in Simulink block diagram in Figure 1.

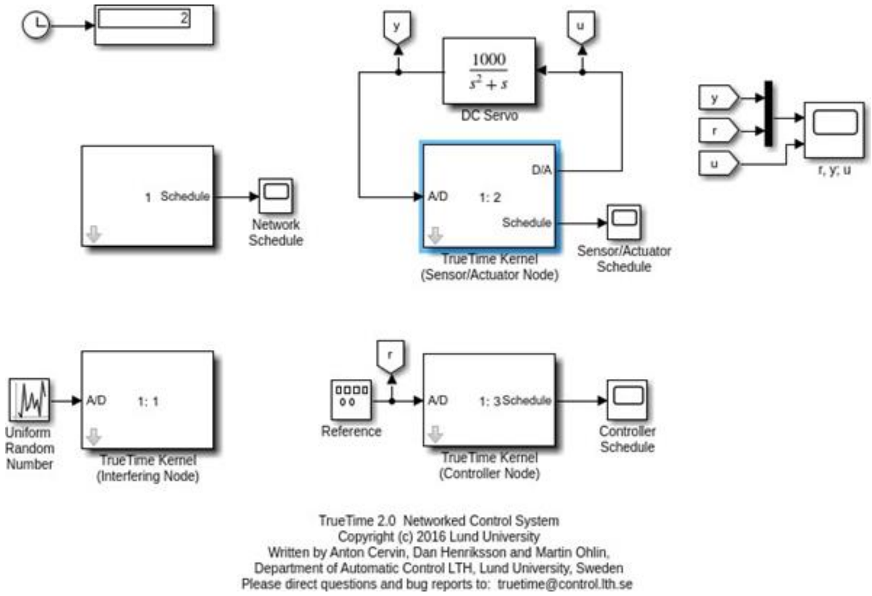

Figure 1 Networked System

As it can be observed in the Figure 1, the objective is to control a Servo motor in a distributed control system where there are three nodes and one network.

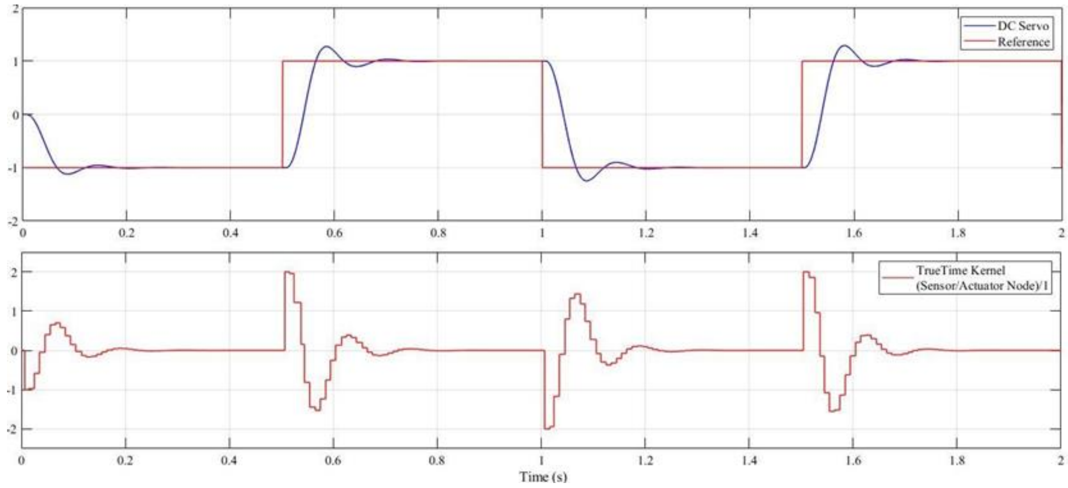

Figure 2 Time response and control signal of Networked system

A first run, illustrated in Figure 2, shows that with the current configuration the system is stable in the upper half of the graph, the DC servo system (Blue Legend) follows the reference (Purple Legend) and it's also realizable in the control signal. In the bottom half of the graph the control signal maintains itself within acceptable current (Ampere) values. The classical control design is appropriate. Looking into the scheduling there is little to no interference between the messages in the network.

2.1 Network

An overview over the network tells that the default medium access protocol is Carrier Sense Multiple Access with Collision Detection (CSMA/CD), a common protocol for the Ethernet. It also has data rate of 80,000 bit/sec and a minimum frame size of 80-bit, meaning that the minimum time a message occupies the network is 1 ms.

2.2 Node 1 (Interference)

The Node number 1 is an interference generator that publishes in the network random noise that is generated; having only one task. The kernel has a fixed priority assignment and number. Period is same as the deadline, is defined to 1 ms and it publishes an 80-bit message to the network occupying the network for 1 ms.

2.3 Node 2 (Plant)

The Node number 2 has both actuator and sensor. The node has two tasks: the sensor with a deadline that is equal to its period of 10 ms, a total execution time of 0.9 ms, and publishes messages of 80-bit occupying the network for 1ms to the third node. The actuator, which is event based with a deadline of 10 s, an execution time of 0.5 ms, and receives messages from the third node. The default scheduling is the Deadline-Monotonic Priority Assignment (DMPA) so the sensor always has a higher priority than the actuator.

2.4 Node 3 (Controller)

The Node number 3 is the PID controller. Only has one event-based task that activates with the reception of messages from the second node and has a deadline of 10 ms. Then, publishes 80-bit message to the network with the second node as a destination occupying the network for 1 ms.

3.0 CASE STUDIES

3.1 Case Study 1 (NETWORK)

In this section, the objective is to demonstrate different types of networks that can influence the stability of a system. Even with a stable control if we take into consideration real-time constraints. The only change to the original system is the size of the control message from 80-bit to160-bit. This will result in the control signal to occupy 2 ms per frame in the network, elaborated the configuration in section 3.

3.1.1 CSMA/CD (Ethernet)

The Carrier Senses Multiple Accesses with Collision Detection (CSMA/CD) protocol do not define any priority of messages in the network. Nodes try to access the network only when its idle, if a collision is detected the nodes stop trying to access the network and wait. A random defined interval in between before trying to broadcast the message again. With the new configuration, for a CSMA/CD protocol as used in Ethernet the system is unstable as seen in Figure 3. This is a direct representation of one of the disadvantages of CSMA/CD, when the network is too loaded there are too many collisions, and some messages are lost.

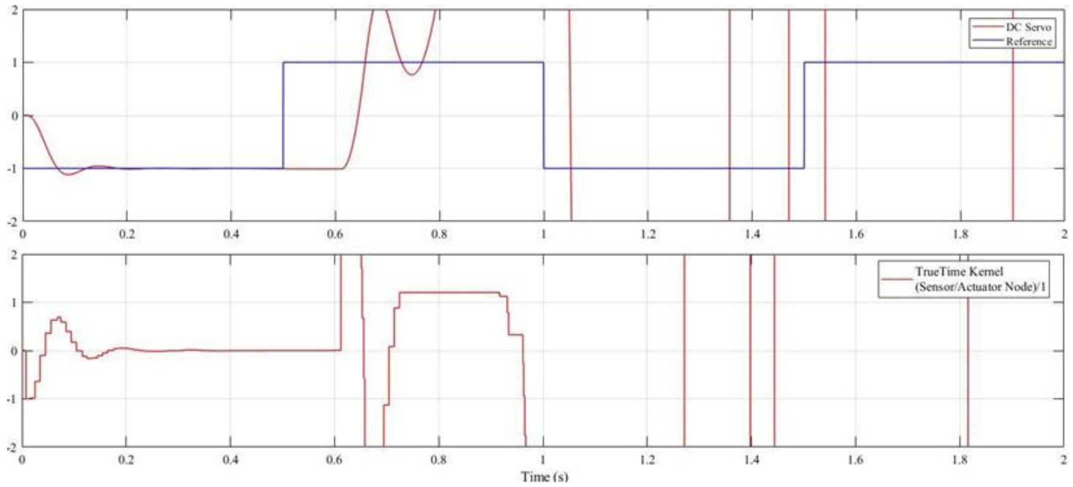

Figure 3 Time response and control signal with CSMA/CD protocol

As it can be seen in Figure 4, the interference between nodes trying to access the medium, results in a loss of sensor messages (Blue Legend) and results in the system falling to instability by delayed control signals (Red Legend).

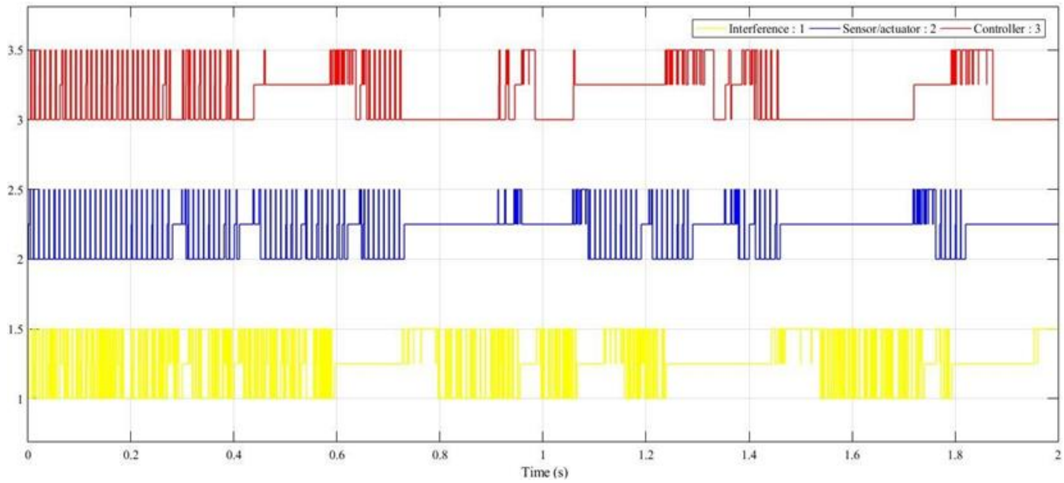

Figure 4 Network scheduling with CSMA/CD protocol

3.1.2 CSMA/AMP (CAN)

On the other hand, the Carrier-Sense Multiple Access with Arbitration on Message Priority (CSMA/AMP) deals with collisions in another way. The nodes only try to publish when the network is idle, but if two nodes collide there will be an arbitration for priority. The node with the highest priority takes over and starts transmitting the data. In this system the priority is as follows node ID number 1, node ID number 2, node ID number 3.

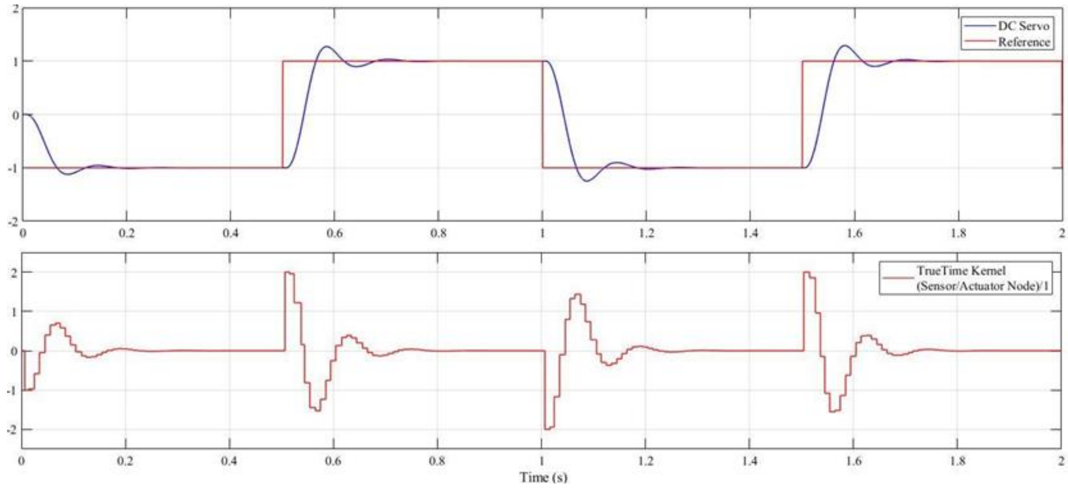

Figure 5 Time response and control signal with CSMA/AMP protocol

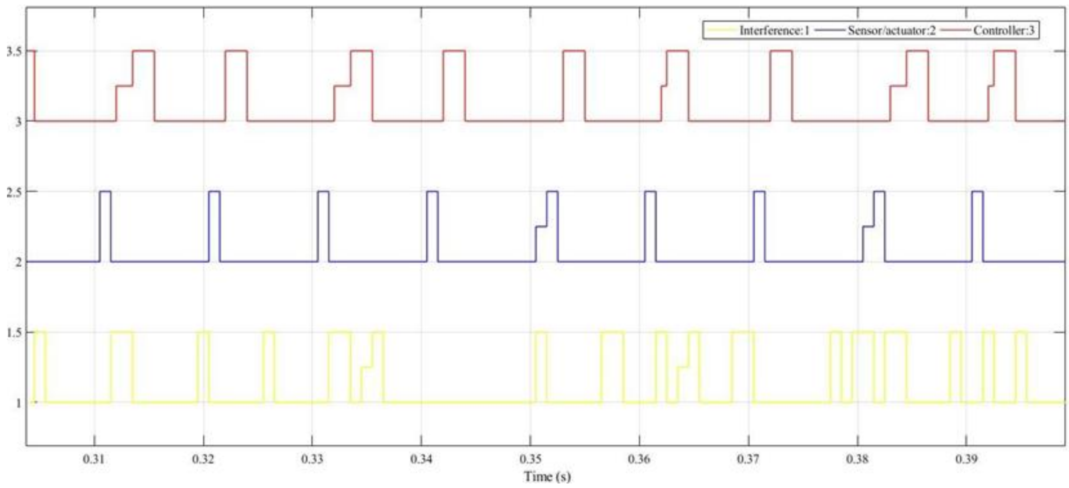

Figure 6 Network scheduling with CSMA/AMP protocol

As seen in Figure 5, the system is stable if the protocol for medium access is changed to CSMA/AMP as used in Controller Area Network (CAN). Unlike CSMA/CD there is no back off time, so there are no messages lost if there are multiple collisions, having always one of them transmitting the message as seen in Figure 6. This allows CSMA/AMP to make a better management for a loaded network.

3.1.3 CAN vs Ethernet

The CSMA/AMP protocol shows clear superiority to the CSMA/CD protocol for a loaded network with real time constraints, since the CSMA/CD can have an unbounded delay [10-11]. However, this results only demonstrate the advantages of CSMA/AMP protocol over CSMA/CD protocol, although this protocols are usually associated with CAN and Ethernet. It is an unfair comparison of the networks. For that the normal network speed and messages size would have to be considered and both are very different and hardware dependent. That will be discussed in section 4.1.

3.2 Case Study 2 (CAN)

The Controller Area Network bus is often used in distributed control systems, to test that the values of CAN bus were inserted in our simulation. Since each control message is only a float number, the data size was assumed to be 4-byte, i.e. each CAN frame would be 76-bit. The speed of CAN depends on the type of physical system it is being used, either ISO11898-2 or ISO11898-3.

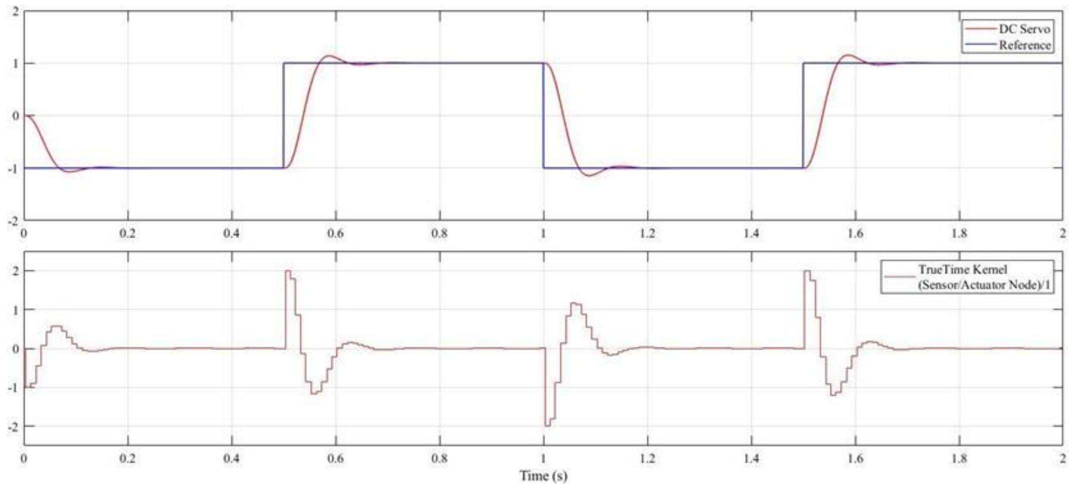

Figure 7 The response with CAN 1SO 11898-2

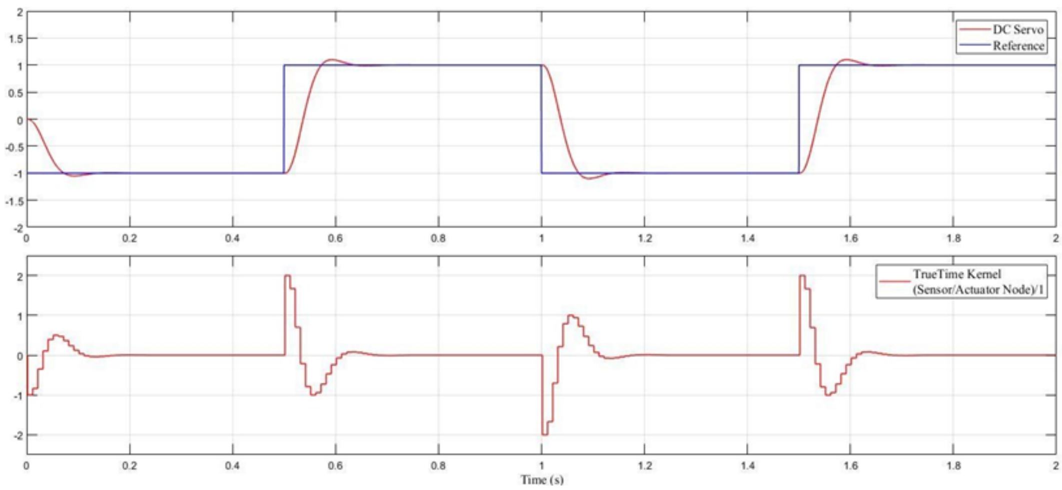

Figure 8 The response with CAN 1SO 11898-3

It can be seen that in both cases the system response is great. With a message size of 76-bit and a CAN bus speed of 1 Mbps as in Figure 7 or 125 Kbps as in Figure 8, this small network with only three nodes has no problem in taking the load. However, high-speed CAN (ISO 111898-2) can be less reliable than the low-speed CAN, also called fault tolerant. Since it operates with voltages that are closer to each other.

In the end raising data loss to 10 % is enough to oscillate the previously fast stabilizing system as seen in Figure 9. As the loss reached to 20 %, even with a speed that would be almost ten times greater than the other CAN physical system, the system enters instability.

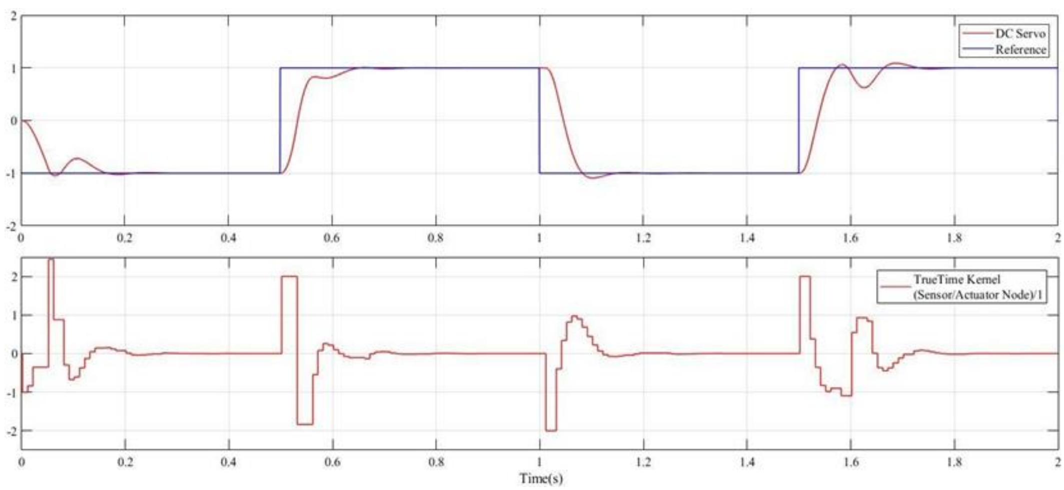

Figure 9 The response of CAN 1SO 11898-2 with 10% data loss

PERINTIS eJournal, **2021**, Vol. 11, No. 1, pp. 40-60

3.2.1 CAN vs Ethernet

As discussed in previous section, the medium access protocol CSMA/AMP has advantages over the CSMA/CD for loaded networks. However, it was an unfair comparison of CAN and Ethernet. So here, taking into account that the CAN network already behaves properly as seen in Figure 7 and Figure 8 respectively. Lets test the Ethernet with real life values. First, the minimum frame size for the Ethernet protocol is 512-bit. Second, the speed of Ethernet is tested for two distinct cases such as 1 Mbps and 500 Kbps. Notice that above the 1 Mbps there is actually no difference in the control system, since all messages are delivered with almost no interference in the network making the system response more dependent on periodicity of the messages rather than the network.

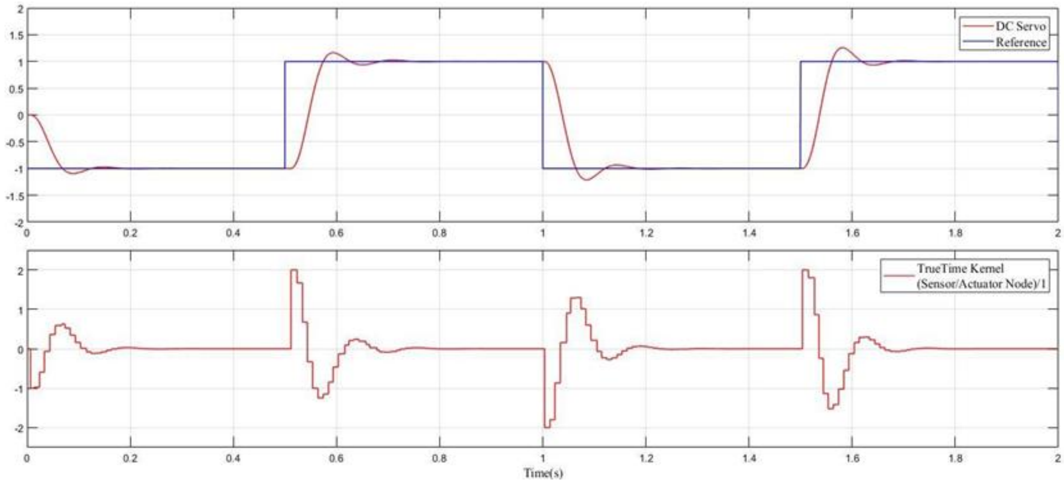

Figure 10 Time response with Ethernet 500 Kbps

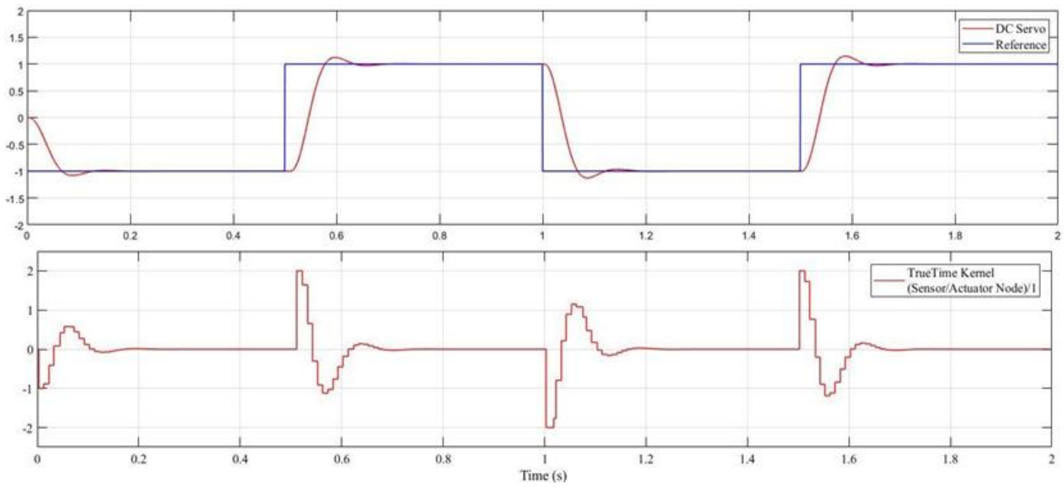

Figure 11 Time response with Ethernet 1 Mbps

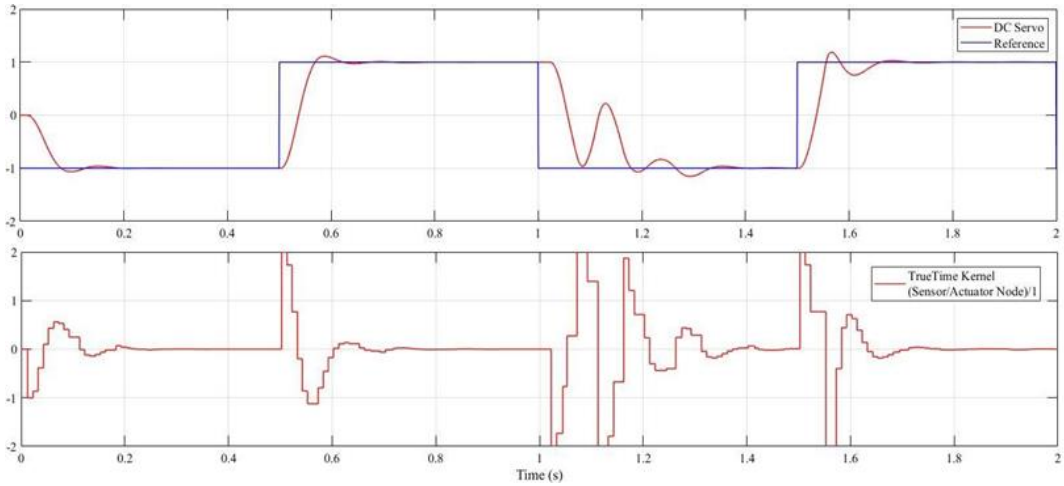

Figure 12 The response of Ethernet 1 Mbps with 10% data loss

In the Figure 10 and Figure 11, the system response starts to take a toll due to the CSMA/CD protocol, with a loaded network the number of collisions increases resulting in delayed control signals and oscillations. With lower speed values the system goes to unstable. In a comparison between Ethernet and CAN at 1 Mbps with 10% data loss, as seen in Figures 9 and 12, the system is performance is affected. However, there is no indication that one protocol is more resistant to data loss than the other. In conclusion of this comparison, in this particular system, both networks work smoothly since this networked system is not complex.

3.3 Case Study 3 (SCHEDULING)

Using the CSMA/AMP protocol in Case study 2 the configuration of the system is changed to see the impact of the scheduling in the system. A scheduling algorithm affects how the tasks in each node are scheduled in the processor, each node has a different scheduling algorithm. Since node 2 is the only one with two tasks. The scheduling is changed there to see the task conflict, obtaining the processor usage. The new configuration of the system is different from the original system in actuator execution time 2ms, actuator deadline 4ms, sensor execution time 2ms, and sensor deadline 5ms, lets name it configuration B.

3.3.1 Deadline-Monotonic Priority Assignment (DMPA)

The Deadline-Monotonic Priority Assignment (DMPA) scheduling method assigns priority based on the deadline; the lower deadlines have the higher priority. This is an example of an offline priority assignment. Therefore, in this case the actuator task will have a higher priority over the sensor task. With the configuration B and a DMPA scheduling, it can be clearly observed that in Figure 13 the actuator will always have priority over the sensor taking the execution time in the processor. This configuration results in a system relatively stable, with damped oscillation and overshoot of 2.019 as shown in Figure 14.

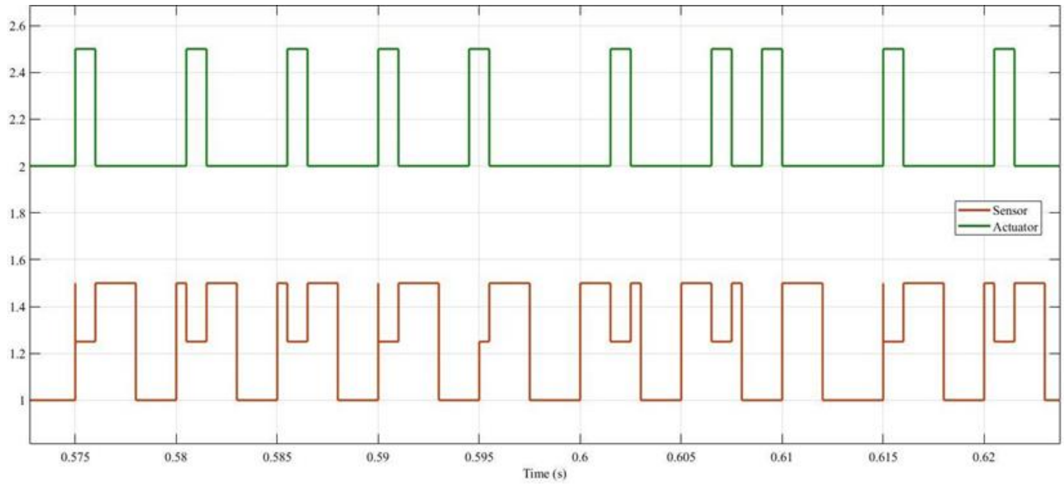

Figure 13 DMPA scheduling in Node number 2 with configuration B

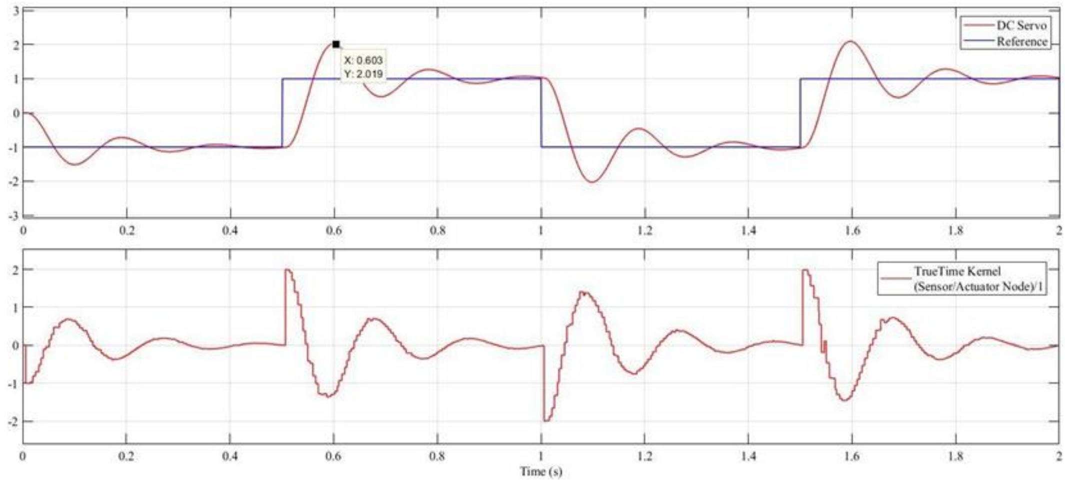

Figure 14 System time response using DMPA scheduling in node number 2 with configuration B

3.3.2 Earliest Deadline First (EDF)

The Earliest Deadline First (EDF) scheduling method assigns priority with an online method, the task with the deadline that will be met first has the highest priority. With the configuration B and a EDF scheduling, the obtained results are the following. It can be clearly observed that in Figure 15 the actuator will have priority over the sensor as well as the sensor will have over the actuator, depending on the earliest deadline. This configuration results in a system relatively stable, with damped oscillation and overshoot of 2.025 as shown in Figure 16.

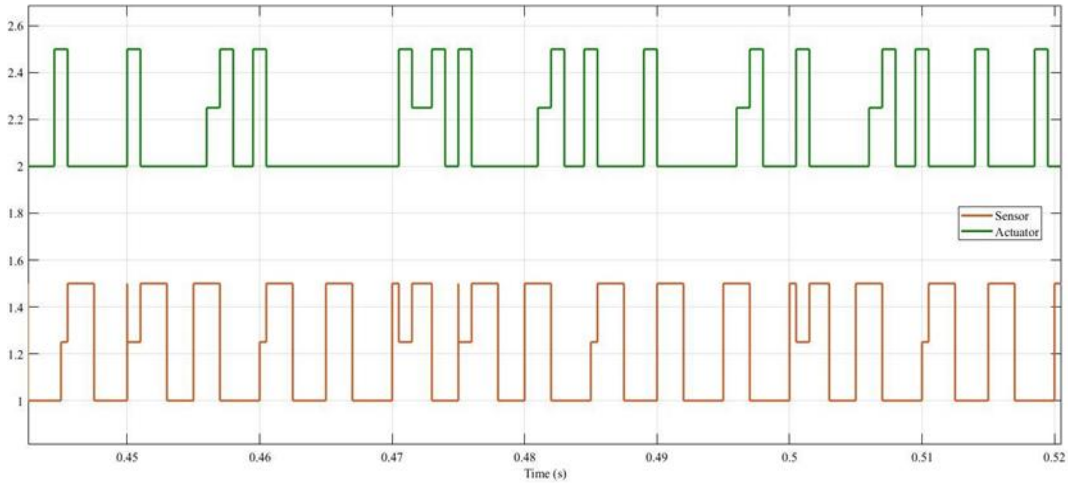

Figure 15 EDF scheduling in node number 2 with configuration B

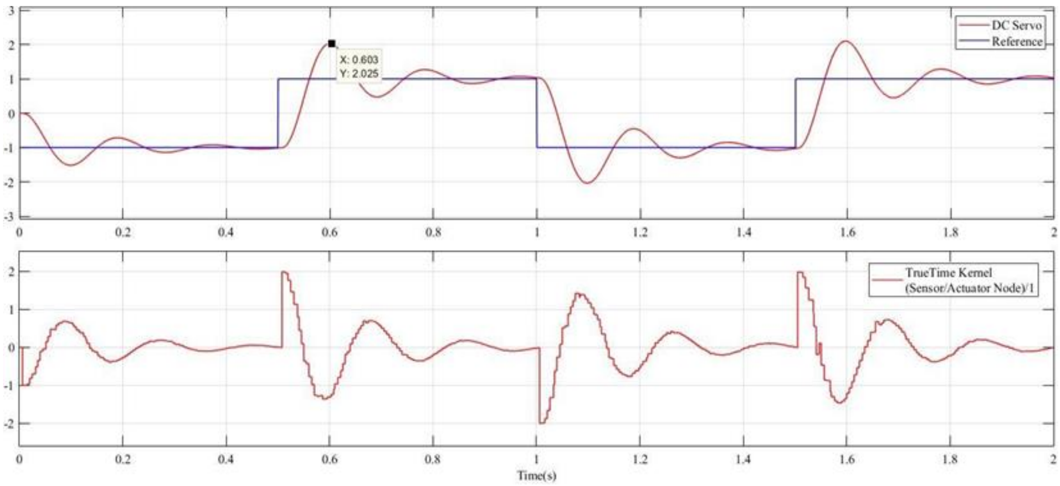

Figure 16 System time response using EDF scheduling in node number 2 with configuration B

3.3.3 DMPA vs EDF

In the first experiment it can be thought that EDF could be prejudicial for the system since the overshoot was increased for Figure 16 in comparison to Figure 14. However, in this system, the tasks are dependent on each other the actuator only executes if the controller sends a message and the controller only executes if the sensor sends a message, so the scheduling is not comparable. It was tested that, with a network with unlimited speed, the system does not vary on its behavior.

3.4 Case Study 4 (SCHEDULING ON CONTROLLER NODE)

In the previous section, the tests were not enough to demonstrate the effect of scheduling in the control system. In this section it is decided to demonstrate it with a hypothetical case where the controller node in the network is a central controller with four tasks to control four different plants. For that, four dummy tasks are created in the controller node, with execution times and periods, which are equal to the deadlines, defined in each scenario.

3.4.1 Scenario 1

The new node tasks namely as configuration C1 have the following values for execution time and deadline as shown in Table 1.

Table 1 Configuration C1

C_{servo} = 1.0ms	D_{servo} = 10ms
C_{d1} = 3.5ms	D_{d1} = 7.0ms
C_{d2} = 6.0ms	D_{d2} = 15ms
C_{d3} = 0.66ms	D_{d3} = 1.1ms

With the said values and the other entire configuration unchanged, if we assume an ideal network, the period of the controller would be the same as the period of the sensor, i.e., the value of 10 ms. For that we calculate the total usage of the CPU.

PERINTIS eJournal, **2021**, Vol. 11, No. 1, pp. 40-60

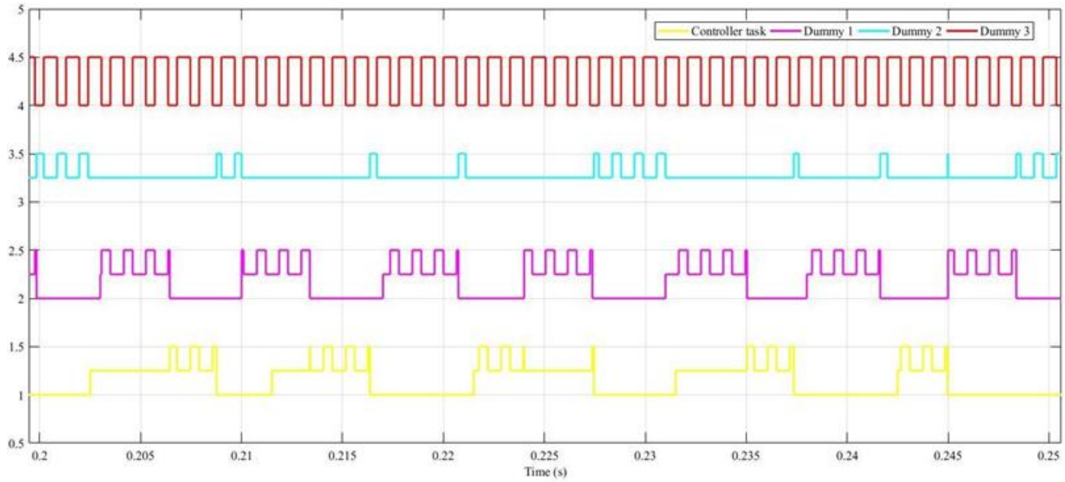

Figure 17 Scheduling using DMPA in Node number 3 with configuration C1

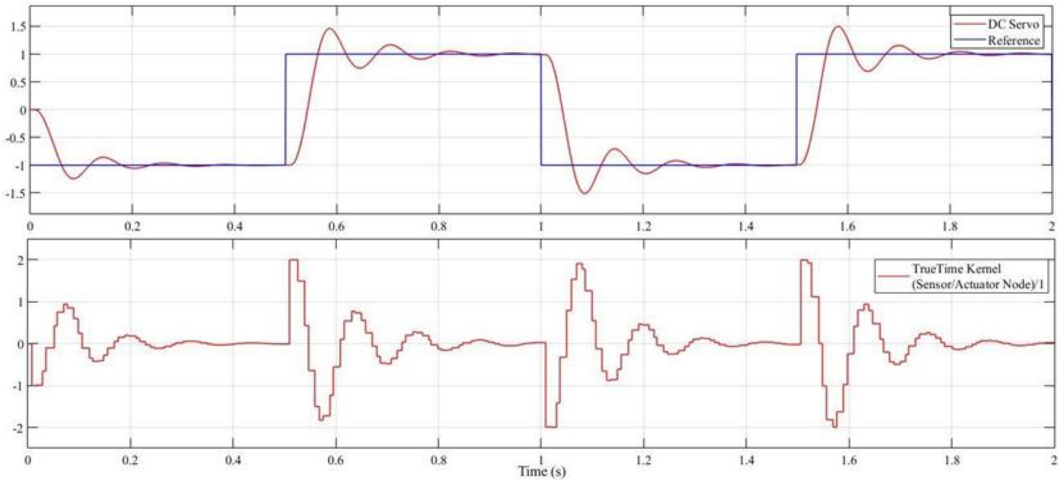

Figure 18 System time response using DMPA scheduling in node number 3 with configuration C1

For this configuration, as seen in Figure 18, the system shows a little oscillation on the time response if using DMPA, but otherwise stable. On the other hand, when using the EDF scheduling algorithm, the system goes unstable shown in Figure 20.

PERINTIS eJournal

PERINTIS eJournal, **2021**, Vol. 11, No. 1, pp. 40-60

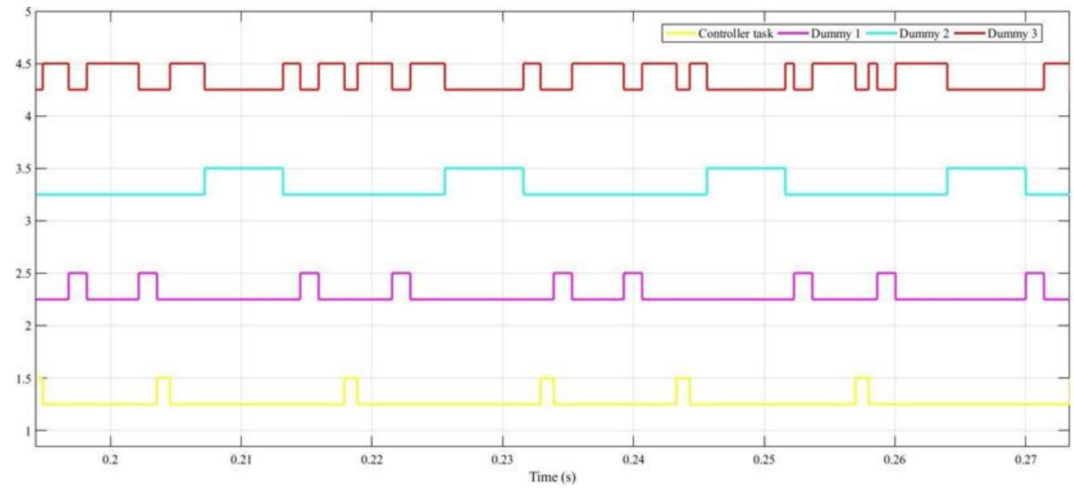

Figure 19 Scheduling using EDF in Node number 3 with configuration C1

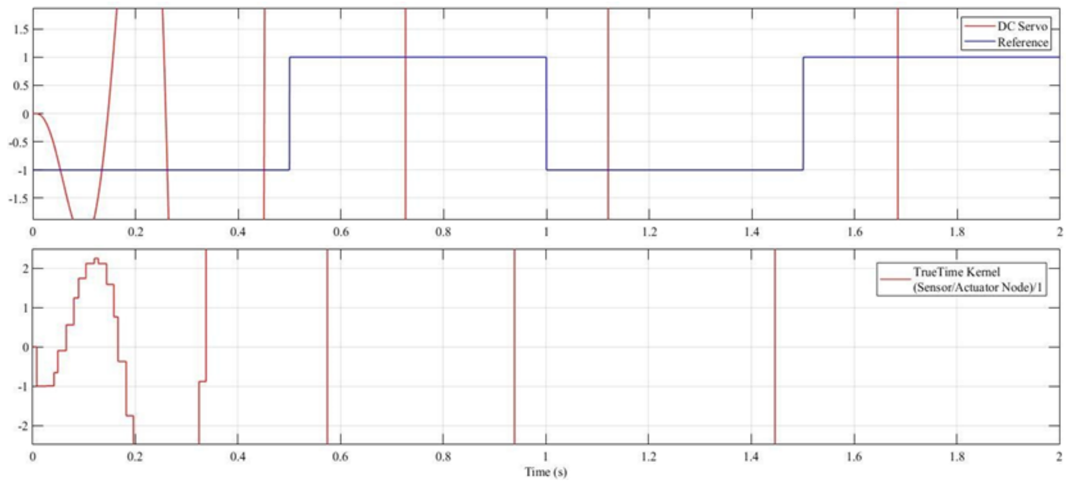

Figure 20 System time response using EDF scheduling in node number 3 with configuration C1

This is due to a heavy task, d2 in bright blue that occupies most of the processor time if allowed to run. Then all the other tasks are constrained and the processor itself is always running on delayed tasks. In this kind of situation, a control engineer would have to find a middle ground between the stability of the different plants depending on time response required and the importance of different tasks. For example, in a car system, the priority is higher in life and death situations than in commodities control.

3.4.2 Scenario 2

The new node tasks namely Configuration C2 have the following values for execution time and deadline as shown in Table 2.

Table 2 Configuration C2

C_{servo} = 1.0ms	D_{servo} = 10ms
C_{d1} = 1.4ms	D_{d1} = 7.0ms
C_{d2} = 2.4ms	D_{d2} = 8.0ms
C_{d3} = 0.44ms	D_{d3} = 1.1ms

Unlike configuration C1, in this there is not a node that occupied the network massively, so in the scheduling algorithm of EDF, all nodes should somehow publish a message. As observed in Figure 21, the system oscillates, but otherwise it is stable. This implies that some deadlines are being missed, but the system does not have a hard deadline and therefore it does not impact the system heavily.

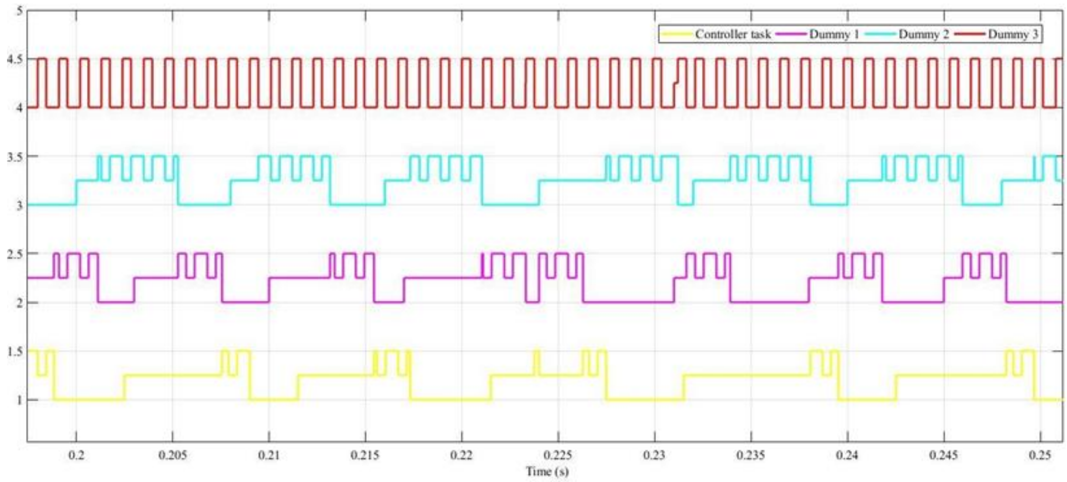

Figure 21 Scheduling using EDF in Node number 3 with configuration C1

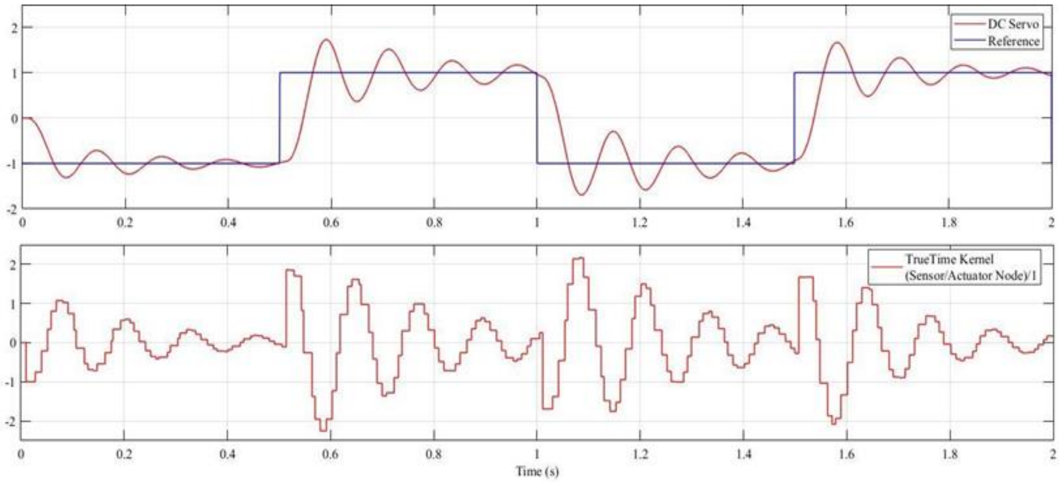

Figure 22 Time response using EDF scheduling in node number 3 with configuration C2

On the other hand, if using the DMPA approach, the controller will have the lowest priority and therefore it will be the one with the worst performance. The worst-case scenario of the controller was calculated with iterations until the stopping condition is above the deadline. This indicates that for DMPA, there will be missed deadlines. Although our system has a soft deadline there is a limit of tolerance. As seen in Figure 23, the servo controller barely executes itself, making the delay on the control signal critical leading to an unstable system shown in Figure 24.

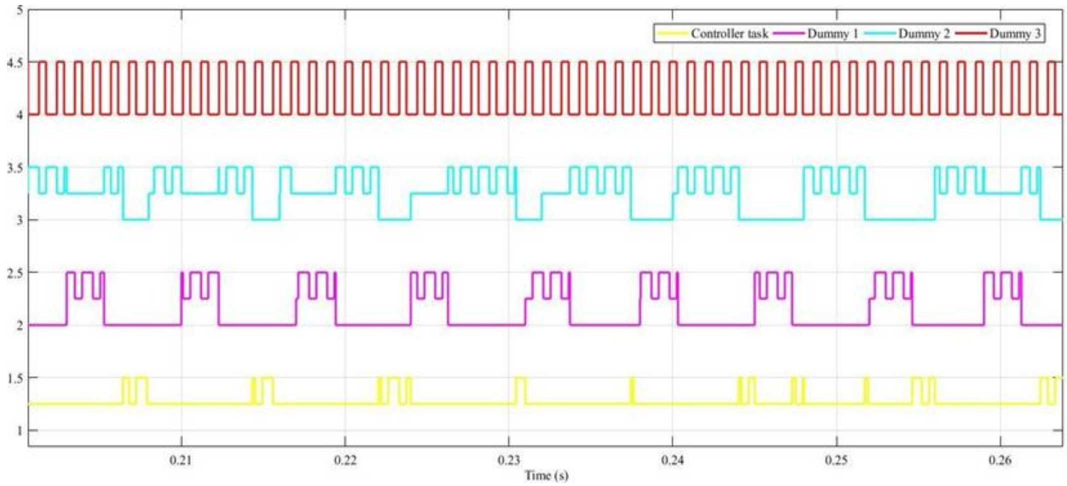

Figure 23 Scheduling using DMPA in Node number 3 with configuration C2

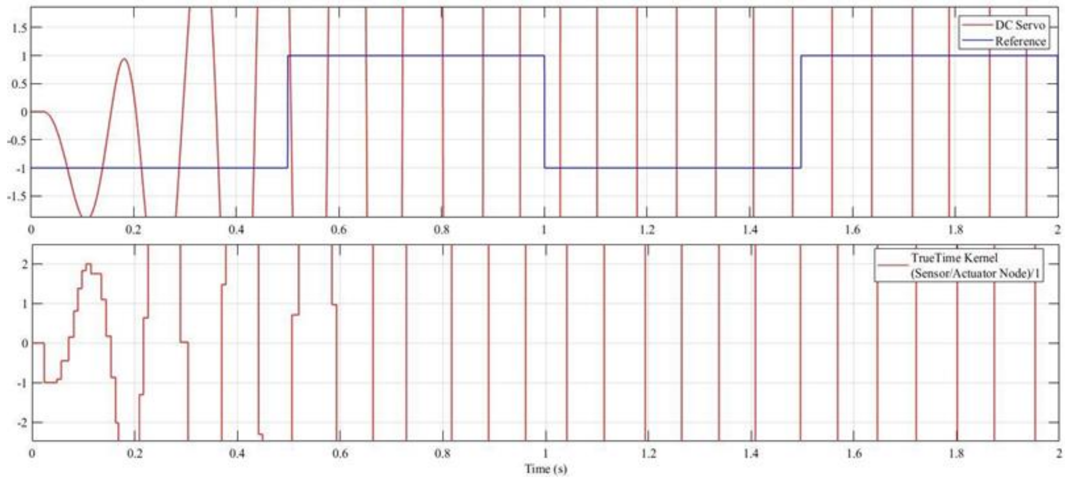

Figure 24 Scheduling using DMPA in Node number 3 with configuration C2

4.0 CONCLUSION

This work analyzes the impact of the network and scheduling algorithms in a distributed network control system. Iterations regarding network protocols, types, and passed onto tweaking with scheduling algorithms within the nodes. In Case study 1, changing the protocol on the network from CSMA/CD to CSMA/AMP, seeing the effect in a control system. With a higher priority on the sensor, the CSMA/AMP shows superiority in maintaining the system stable. These two protocols are not common in Ethernet and CAN respectively. But the comparison is unfair, as demonstrated in Case study 2. If the true values of the CAN and Ethernet networks are used, both systems display a good performance. Only if we consider a high-risk network with lots of data loss (more than 20%), the system becomes unstable. The other half of the paper, Case study 3 and Case study 4, analyzed the impact of the scheduling in the stability of the control system. It can only be concluded that the scheduling should be done depending on the tasks needs. For example, a task with a hard deadline should have a higher priority than a task with a soft deadline or a system with a slower time response could have a bigger period between control signals. So, choosing between EDF and DM or any other kind of scheduling algorithm, is in the hands of the real-time control engineer. This particular system has a good performance with any of the CAN ISO recommendations, without changing any other parameters.

ACKNOWLEDGMENT

The authors would like to express their appreciation to Kurita Water and Environment Foundation (KWEF) for providing Kurita Oversea Research Grant 2020 (20Pmy140-K31) which makes this research viable and effective. The authors would also like to acknowledge Universiti Teknologi Malaysia (UTM) for support and providing facilities to complete this research.

REFERENCES

[1] Astrom, K. J., and Wittenmark, B. (2013). Computer control. *Computer-Controlled Systems: Theory and Design, Third Edition.* USA: Dover Publications, Inc.

[2] Sivakumar, P., Vinod, B., Devi, R. S., and Rajkumar, E. J. (2015). Real-time task scheduling for distributed embedded system using MATLAB toolboxes. *Indian Journal of Science and Technology*, 8(15), 1-7.

[3] Faudzi, A.A.M., Ali, M. H. K., Azman, M. A., and Ismail, Z. H. (2012). Real-time hand gestures system for mobile robots control. *Procedia Engineering*, 41, 798-804.

[4] Ibrahim, S., Hyder, W., Yunus, M. A. M., Wahap, A. R., Rahim, R. A., and Pusppanathan, J. (2020). An ultrasonic system for detecting defects in wood. *Journal of Tomography System and Sensor Application*, 3(1), 122-130.

[5] Shoani, M. T., Ribuan, M. N. and Faudzi, A. A. M. (2019). Extendable Hybrid Linear Actuator: A Mini-Review. *PERINTIS eJournal*, 9(2), 48-57.

[6] Rahmat, M. F., Sunar, N. H., Salim, S. N. S., Abidin, M. S. Z., Faudzi, A. A. M., and Ismail, Z. H. (2011). Review on modeling and controller design in pneumatic actuator control system. *International Journal on Smart Sensing and Intelligent Systems*, 4(4), 630-661.

[7] Zhang, P. (2010). Industrial control systems. *Advanced Industrial Control Technology*. UK: Elsevier Ltd.

[8] Jithish, J., and Sankaran, S. (2017). Securing networked control systems: modeling attacks and defenses. *IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, 7-11.

[9] F Lian, F. L., Moyne, J. R., and Tilbury, D. M. (2001). Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet. *IEEE Control Systems Magazine*, 21(1), 66-83.

[10] Wheels, J. D. (1993). Process control communications: token bus, CSMA/CD, or token ring. *ISA Transactions*, 32(2), 193-198.

[11] Khanna, V. K., and Singh, S. (1994). An improved "Piggyback Ethernet" protocol and its analysis. *Computer Networks ISDN Systems*, 26(11), 1437-1446.

[12] Olufowobi, H., Young, C., Zambreno, J., and Bloom, G. (2019). SAIDuCANT: specification-based automotive intrusion detection using controller area network (CAN) timing. *IEEE Transactions on Vehicular Technology*, 69(2), 1484-1494.

[13] Abdul-Hussain, R. N., and Awad, O. A. (2019). Studying the effect of sampling time and network load on wireless networked control systems. *Journal of Al-Qadisiyah for Computer Science and Mathematics*, 11(3), 31-42.