Article ECORFAN Journal Democratic Republic of Congo
December 2020 Vol.6 No.11 17-23

Heuristic study of a full can implementation on a network of digital controllers based on experimental data

Estudio heurístico de una implementación full can dentro de una red de controladores digitales a partir de datos experimentales

LUJÁN-RAMÍREZ, Carlos Alberto†, SANDOVAL-GÍO, Jesús*, FLORES-NOVELO, Agustín Alfonso and OJEDA-ARANA, Juan Alberto

Tecnológico Nacional De México / I.T. Mérida, Mexico.

ID 1st Author: *Carlos Alberto, Luján-Ramírez* / **ORC ID:** 0000-0002-8978-9188, **Researcher ID Thomson:** T-6838-2018, **arXiv Author ID:** clujan, **CVU CONACYT ID:** 296592

ID 1st Co-author: *Jesús, Sandoval-Gío* / **ORC ID:** 0000-0001-5847-3669, **Researcher ID Thomson:** V-1930-2018, **arXiv Author ID:** jesus.sandoval, **CVU CONACYT ID:** 297308

ID 2nd Co-author: *Agustín Alfonso, Flores-Novelo* / **ORC ID:** 0000-0002-9691-8056, **Researcher ID Thomson:** T-3024-2019, **arXiv Author ID:** afloresnovelo, **CVU CONACYT ID:** 624909

ID 3rd Co-author: *Juan Alberto, Ojeda-Arana* / **ORC ID:** 0000-0002-7239, **arXiv Author ID:** JuanOjeda, **CVU CONACYT ID:** 580342

DOI: 10.35429/EJDRC.2020.11.6.17.23 Received July 20, 2020; Accepted December 30, 2020

Abstract

Over time, the CAN (Controller Area Network) communication bus has been implemented in different technological sectors, within which, depending on the application, the bus implementation may change. On the other hand, the design and implementation of digital controls based on experimental data is a well-known topic in the automation industry where the acquisition system is of great importance. In this document, a heuristic study of the behavior of a Full CAN network is reported to implement digital controllers in two interconnected control loops. This study takes into account the access time to the bus and the amount of data sent when observing the response to disturbances. The design of two digital controllers is presented based on the parametric identification of two plants: a DC motor with an electromagnetic brake and a pneumatic levitator. Using PSoC® microcontrollers, a Full CAN network is implemented, where the digital controllers exchange data by randomly accessing the bus. A specially designed interface allows visualizing the speed and amount of data transferred under different operating conditions of the control loops. At the document end, the experimental data obtained are discussed.

Full CAN, Digital Control, PSoC

Resumen

A lo largo del tiempo, el bus de comunicaciones CAN (Controller Area Network), ha sido implementado en diferentes sectores tecnológicos, dentro de las cuales, dependiendo de la aplicación, la implementación del bus puede cambiar. Por otra parte, el diseño e implementación de controles digitales a partir de datos experimentales es un tema muy conocido en la industria de la automatización en donde el sistema de adquisición tiene una gran importancia. En este documento, se reporta un estudio heurístico del comportamiento de una red Full CAN para implementar controladoreses digitales en dos lazos de control interconectados. Dicho estudio toma en cuenta el tiempo de acceso al bus y la cantidad de datos enviados al observar la respuesta ante perturbaciones. Se presenta el diseño de sendos controladores digitales a partir de la identificación paramétrica de dos plantas: un motor de CC con freno electromagnético y un levitador neumático. Usando microcontroladoes PSoC® se implementa una red Full CAN, donde los controladores digitales intercambian datos accediendo al bus aleatoriamente. Una interfaz especialmente diseñada permite visualizar la rapidez y cantidad de datos transferida bajo distintas condiciones de operación de los lazos de control. Finalmente se discuten los datos experimentales obtenidos.

Full-CAN, Control Digital, PSoC

Citation: LUJÁN-RAMÍREZ, Carlos Alberto, SANDOVAL-GÍO, Jesús, FLORES-NOVELO, Agustín Alfonso and OJEDA-ARANA, Juan Alberto. Heuristic study of a full can implementation on a network of digital controllers based on experimental data00. Journal - Democratic Republic of Congo. 2020. 6-11:17-23.

---

* Correspondence to Author (Email: jesus.sg@merida.tecnm.mx)

† Researcher contributing as first author.

Introduction

The CAN communications bus was created for the interconnection of several devices without the need for dense cabling, to have a robust and economical communication with high immunity to noise. Although originally this protocol was developed for automotive applications, today it has ventured into various areas of engineering. Since the appearance of the CAN communication protocol at the end of the 80s, the bus has evolved thanks to technological advances, which allowed the inclusion of this communication bus in various areas, such as industrial, home automation, biomedical, the oil company, among others (Lawrenz, 1997). In applications destined to process control, the CAN bus has gained a lot of strength in the industrial sector, due to its low implementation costs and its reliability in high interference environments, communicating different common devices in the implementation within industrial processes (Guo, 2011) (Bailey, 2015).

The CAN communication bus is based on the OSI reference model (Open Systems Interconnection Model) where it makes use of three layers of the reference model: in the physical layer, there is a transceiver where the H and L lines are connected. of the communications bus (Bailey, 2015); Said transceiver must have the necessary electrical characteristics in accordance with the standard used (ISO11898, ISO11519, ISO11992, SAEJ2411, among others.). In the data link layer is the CAN module embedded in the controller; This module will be in charge of the logical link control and the link control to the medium. Lastly, the application layer. In this layer are the resources available by the controller for the manipulation of actuators, sensors or some interface for the final application (Bailey, 2015).

The CAN communication bus can be classified according to certain characteristics, one of them is the configuration of the buffer that receives the data of the frame, also called a mailbox (Mailbox). This configuration can have the following implementations: Basic CAN, Full CAN, FIFO, Enhaced Full CAN (Lawrenz, 1997). The most typical implementation on the bus, especially in SCDs (Distributed Control Systems), is Basic CAN (Guo, 2011) (YunxiaJiang, 2011).

Knowing the performance of the bus when processing control signals is of great importance since inherent delays and loss of information degrade the operation of the control loop. Few tools currently exist to study these phenomena in industrial applications. In this work, a heuristic study of the performance of a Full CAN (FCAN) implementation within a digital control loop, shared by two systems, is made. It describes the design of the controllers, the implementation of the bus. At the end of the document the results obtained when the control loops are subjected to disturbances are discussed.

Method

FCAN implementation in two interconnected digital control systems

For the development of this work, a CAN communications bus is implemented, where each bus node will perform a different task: one will process the digital control algorithms of each proposed system. Two systems will be studied: System 1 (S1), the control of a motor before disturbances and System 2 (S2) the control of a pneumatic levator. The implementation of the nodes and the interface of the sensors, actuators and transceivers was carried out in the CY8CKIT-059® kit, based on the CY8C5888LTI-LP097® microcontroller, from CYPRESS®.

Description of the FCAN bus implemented

The 4-node FCAN network is proposed. Each node will perform the following tasks:

Node 1. Will send and receive data from S1 to the bus. It will be used to activate sensors and relays of the assigned system. This node will be of high priority.

Node 2. Will send and receive data to the bus from S2. This node will manage the sensors and actuators of S2.

Node 3. In this node the processing of the digital controls for the S1 and S2 systems will be performed.

Node 4: This node will receive and send data from the bus to the interface through USB-HID communication.

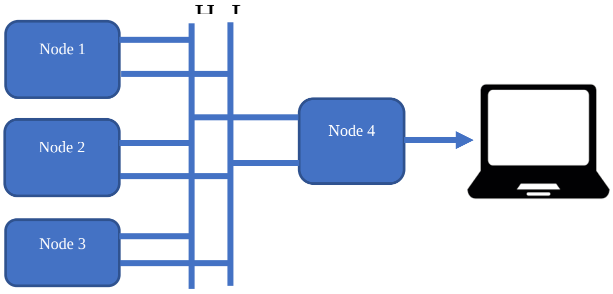


Figure 1 Structure of the network to study
Source: Self made

Systems modeling

The transfer function of each system will be found by parametric identification. The information sent or received will be continuous, which will generate a congestion of information on the bus.

The parameters for designing the controls for each system are then established.

System 1. Disturbed DC motor.

The appropriate control was designed so that at a certain disturbance (mechanical braking) of the system, it maintains its speed. S1 is made up of a 24 VDC direct current motor which, from a magnetic clutch, will induce its partial braking. The parameters to satisfy in the control will be that the system responds in no more than 1.5 seconds with less than 5% overdraft and a steady state error of 1%.

System 2. Pneumatic levator.

The S2 system is formed by a pneumatic levator. This "levitates" a disk inside a transparent tube until it reaches a desired level. The control objective is to design a controller so that the disc maintains the set position when there is a disturbance such as variation in the area of the upper air outlet window. The control that was designed complies with the following: response time of two seconds with a steady-state error less than 2%, and the overdraft must be less than 5%.

Control layout for S1.

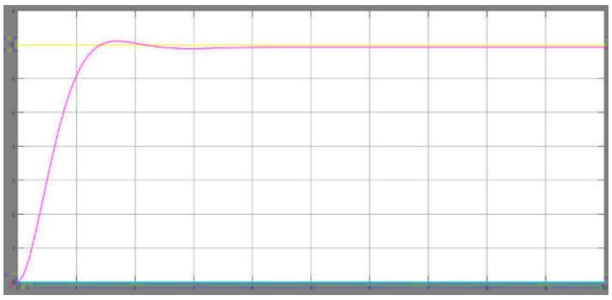
As a first step, the transfer function of the system to be designed is obtained, by means of parametric identification.

As mentioned (Novelo, 2014) there is an extensive study of control systems, where the behavior of systems when they are excited by a specific signal is already known (Yunxia Jiang, 2011). The choice of the appropriate sensors for the acquisition of data from the system to be identified is of great importance, since a bad reading can generate an error in the determination of the transfer function. In this case, a step signal was applied to the input, setting the motor speed at 25 RPS (Revolutions Per Second) when it is being braked (disturbance). Subsequently, the brake is released and a graph of the behavior of the motor is constructed.

With the information obtained from the graph of the reaction curve, the transfer function is found. According to (Novelo, 2014), the behavior of the system corresponds to a first-order overdamped system without delay. Applying the techniques exposed by (Novelo, 2014) the transfer function of the system is:

$$G_S = \frac{1}{(.4S+1)(.4S+1)} \quad (1)$$

Graphic 1 Behavior of the engine before disturbance
Source: Self-made



Graphic 2 Behavior of the designed control
Source: Self-made

With the transfer function of Eq. 1, proceed to design the control for the system. Following the techniques established in (M. García Juárez, 2015) (Novelo, 2014), a P-I control results.

$$\frac{s+1.7}{s+0.017} \quad (2)$$

Next, the characterization of the control in Matlab is presented.

$$\frac{s+1.7}{s+0.017} \tag{3}$$

The control design was carried out in the continuous medium, so it is necessary to convert it to the discrete medium in order to implement it in a microcontroller. To convert from the continuous medium to the discrete medium, the Tustin discretization method (Novelo, 2014) is used with the sampling time of 1 second. Subsequently, it is checked whether the discretization was correct by graphically comparing the behavior of the control in continuous medium against the discrete equivalent using simulation software such as Matlab.

Transfer function:

$$\frac{.0265z+.02243}{z^2-1.558z+.6065} \tag{4}$$

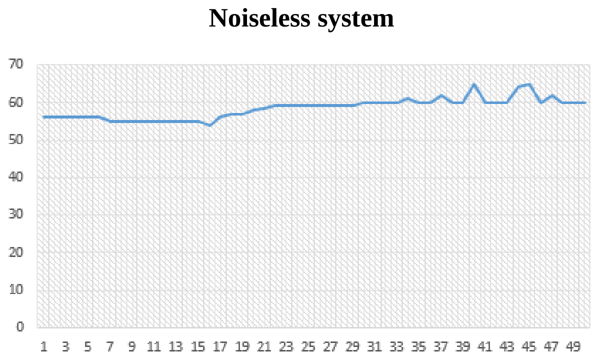
P-I control:

According to the techniques used for digital control (Novelo, 2014) (M. García Juárez, 2015), the difference equation of the designed control is obtained.

$$G_s(z)=\frac{U_z}{E_z}=\frac{z-0.8301}{z-0.9983} \tag{5}$$

$$U_{(k)}=0.9983\,U(k-1)+E(k)-0.8301\,E(k-1) \tag{6}$$

Obtaining the transfer function of the control 2. For the calculation of the digital control, it is necessary to know the transfer function of the pneumatic levitator. According to (Novelo, 2014) a reference is set so that a disturbance is subsequently added to the system, the disturbance is represented in Graphic 3.



Graphic 3 System behavior S1
Source: Self-made

Using the techniques proposed in (Novelo, 2014), the resulting transfer function is:

$$G_S=\frac{1.015}{(.8S+1)(.45S+1)} \tag{7}$$

Therefore, it is necessary to design a control whose response is at least 2 seconds with a steady-state error of 2%. Using the techniques for the design of controllers in the continuous medium (Novelo, 2014) the following control is obtained:

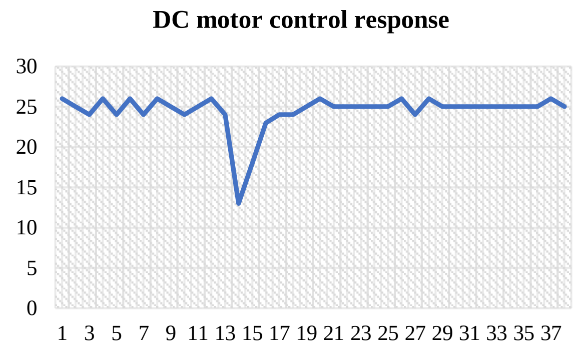
$$\frac{s+1.2}{s+0.025} \tag{8}$$

Subsequently, the control obtained is discretized to obtain the difference equation. This equation will be the one that will be programmed into the microcontroller.

$$\frac{z-0.8801}{z-0.9975} \tag{9}$$

Experimental results

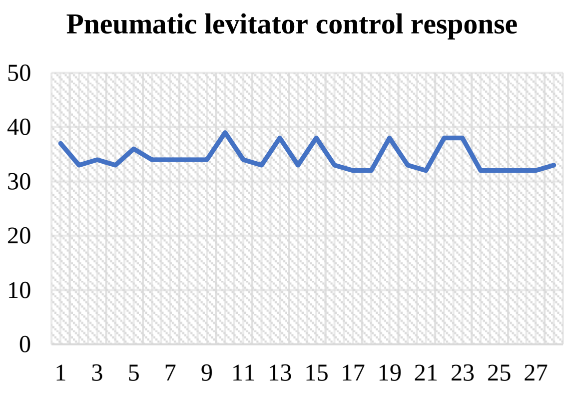
Next, the behavior of the controls in the different systems will be presented, and a comparison will be made of the number of data sent successfully on the bus with FCAN implementation of the nodes related to the control of each system. The communication bus can be classified according to the Mailbox implementation or by the type of frame and even by the type of hardware embedded in the chip. This work focuses on the Mailbox implementation, especially the Full-CAN implementation. The microcontroller used to implement the interface of the sensors, actuators and transceivers was the CY8C5888LTI-LP097® in the CY8CKIT-059® kit from the CYPRESS® company. For the visualization of the systems, an interface was developed in Visual C # ® with the possibility of saving the information in an Excel® spreadsheet.



Graphic 4 Response to system disturbance S1
Source: Self-made

The PSoc Creator® facilitates the configuration of CAN modules, since it has a wizard, which reduces the amount of code to write. That said, the FCAN implementation and the assignment of the identifiers for each node was carried out successfully. One of the advantages of choosing the Full-CAN implementation in the PSoC is saving time and code, since when selecting the Mailbox as Full, values are automatically established that will make it work. The identifier, in this case, is set by the user. In a Basic-CAN implementation, the interrupts and identifiers would have to be configured via code in the PSoC Creator®.

For the configuration of the transfer speed in CAN communication at 1Mbps, it is important to have an external clock with an error less than or equal to .2%; if not, there is a risk that several messages cannot be transmitted.



Graphic 5 Response to system disturbance S2
Source:Self-made

System control 1.

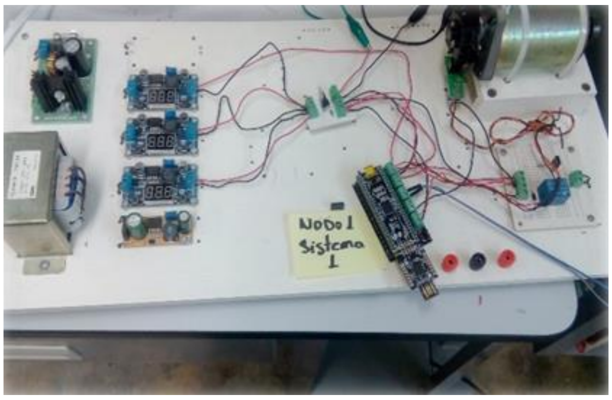


Figure 2 Control hardware 1
Source: Own elaboration

Figure 2 shows the hardware used for the implementation of the proposed system, the behavior of which is represented in Graphic 4.

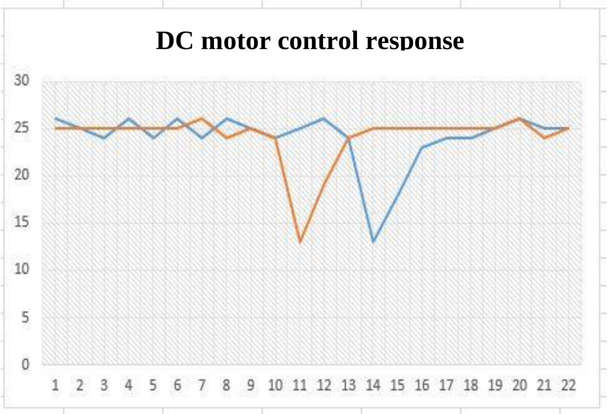
The procedure followed was as follows: previously a set-point of 25 RPS is set with the brake deactivated. Subsequently, braking is activated (action represented in second 13), observing in Graphic 4 that the motor it recovers its speed in approximately 2 seconds.

System control 2.

The hardware and structure of system two can be seen in Figure 3. The testing procedure was as follows, the disk was fixed at a stable height of 36 cm, after applying a disturbance the result is shown in Graphic 5. Given the above, the system has a variation of 2 cm from the reference point.

Monitoring of data transfer on the FCAN bus

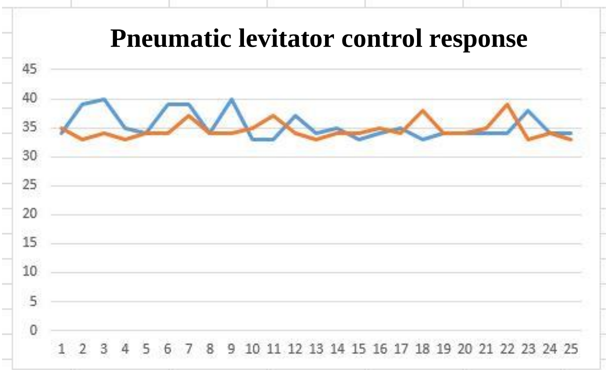
The interface created has the possibility of graphing in each second the amount of data transmitted in a specific node. This is in order to know if there is any loss of information when several nodes try to transmit at the same time. Next, in Graphic 6, the behavior of S1 is presented when only node 1 and 3 are transmitting data. As mentioned previously, node 1 oversees receiving the data from the sensors to later send it to the processing node. After being processed, the data is returned to node 1 to execute a control action on the actuator.



Graphic 6 S1 system control stability
Source: Self-made

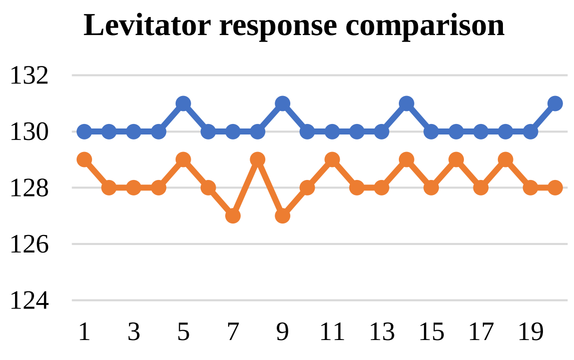
The behavior of the motor in both signals is the same since it only presents a variation of 1 revolution per second. Therefore, the system is not being affected if all the elements of the CAN network request the bus to communicate with each other.

Graphic 7 shows the comparison of the control applied to system 2 when all nodes transmit versus when only nodes 2 and 3 work. The way in which the nodes communicate is as follows: node 2 is in charge of receiving information from the infrared sensor, which later, node 2 sends the data to node 3 (which applies the designed control), after the node 3 processes the information, the node returns information to node 2, this information will be used to control the actuator (motor). In the comparison of the signals of the Graphic, it is observed that in the signal where all the nodes are working (blue signal) there is greater instability.



Graphic 7 S2 system control stability
Source: Self-made.

Graphic 8 presents a comparison of the data successfully transmitted by node 2 in the same cases as the previous Graphic. The blue line represents the data sent when only system 2 is working on the bus.

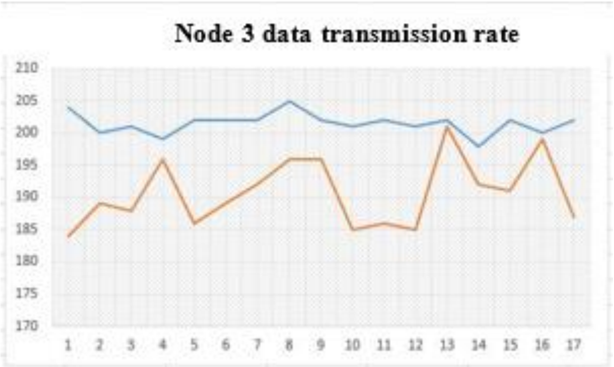


Graphic 8 Data sent on node 2
Source: Self-made

The orange signal represents the data sent when all the nodes are working. It is observed that the orange signal obtained less data sent to the processing node, it is also observed that it presents greater instability when sending data. The difference in sent / received data between the two signals is ± 4 messages.

Processing node monitoring

Graphic 9 shows the data successfully transmitted from node 3 of the CAN network. This node is in charge of processing the digital control for each system, therefore, it receives and sends information to nodes 1, 2 and 4. Graphic 9 shows two signals, one in blue and the other in orange. The first corresponds to the data sent from node 3, when only the levator system is working. The second corresponds to the data sent from node 3, when the two systems are operating. Obviously, a decrease in data sent to the nodes that will activate the corresponding actuators of each system is observed. The difference of data sent from both signals is ± 16.



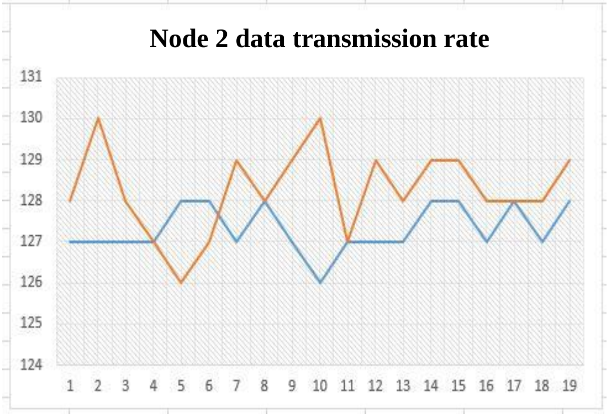
Graphic 9 Data sent from the processing node
Source: Self-made

Node	ID Previous	ID new
Node 1	0x001	0x002
Node 2	0x002	0x001
Node 3	**0x004**	**0x004**

Table 1 Full CAN identifiers
Source: Self-made

Monitoring of bus access in two nodes

Next, a comparison of the data sent is presented by changing the identifier in the FullCAN bus, where, according to (Lawrenz, 1997), the identifier 0x001 is the one with the highest priority. Graph 10 shows the comparison of the data sent by node 2. Said graph shows 2 signals. The first (blue signal) corresponds when the levator system is in operation; the second (orange signal) shows the data sent by node 2 when all the proposed systems are in operation. Comparing Graphic 9 with the current one, it is verified that there is less loss of transmitted messages when all nodes are in operation.



Graphic 10 Data sent on system 2 by changing the priority of nodes
Source: Self-made

Discussion

When observing the behavior of the designed controls and the availability of the nodes when communicating with each other, it was observed that, in certain nodes, the amount of data sent changes. In the levitator system, it was observed that, when all the nodes are in use, the number of messages sent decreases, said decrease, does not greatly affect the control. In contrast, Figure 10 presents the same system with modified identifiers (Table 1). The explanation for the difference in data sent from node 2 is that at the moment when two nodes want access to the bus to send data, the node with the smallest identifier wins access (Bailey, 2015) (Lawrenz, 1997) . The above mentioned is confirmed by the signals presented in Graphic 9, which correspond to node 3 whose identifier is 0x004. This graphic shows that node 4 is the one that loses the most data when all nodes are up and running. Therefore, if you want to implement several control systems in a network with Full-CAN implementation, it is recommended to assign the smallest identifiers to the nodes that need to send information more quickly.

Conclusions

Digital controllers were designed for two systems, one for controlling a DC motor and the other for a pneumatic levator. The effectiveness of the controllers designed by simulation and an experimental implementation of the same was verified. The design of the controllers was based on a parametric identification in two experiments to obtain the reaction curve of each system.

A Full CAN network was implemented where the digital controllers were embedded in a bus node. The microcontroller used to implement the interface of the sensors, actuators and transceivers was CY8C5888LTI-LP097 in the CY8CKIT-059 kit from CYPRESS. A study was carried out on the transfer of data through the bus when the nodes request access and its repercussion on the performance of the control loop.

References

Bailey, A. (2015). CAN BUS. Washington: Washington State University.

Guo, L. C. (2011). Design of the Distributed Control System Based on CAN Bus. En Computer and Information Science (págs. 83-89). Xi'an: Canadian Center of Science and Education.

Lawrenz, W. (1997). CAN system engineering from theory to practical applications. New York: Springer.

Lepkowski, J. (2004). EMI/ESD Protection. Arizona: On semiconductor.

Luo, C. L. (2013). A Co-Simulation-and-Test Method for CAN Bus System. Journal of Communications Vol. 8, 681-689.

M. Garcia Juarez, J. F. (2015). Estudio de estrategias de control PI disparado por eventos para sistemas basados en red. Congreso Nacional de Control Automatico, 447-453.

Novelo, D. A. (2014). Obtención de la Función de Transferencia de Sistemas mediante la Identificación Paramétrica a partir de datos Experimentales. En Apuntes de control. Mérida Yucatán, México.

Yunxia Jiang, B. X. (2011). Design and Implementation of CAN-Bus Experimental System. IEEE, 655-659