VOL. 17, NO. 11, JUNE 2022 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

ASYNCHRONOUS ALGORITHM FOR THE TRANSMISSION OF DATA PACKETS IN MULTI-ROBOT SYSTEMS

Holman Montiel Ariza[1], Fredy H. Martínez S.[1] and Faiber Robayo Betancourt[2]
[1]Facultad Tecnológica, Universidad Distrital Francisco José de Caldas, Bogotá D. C., Colombia
[2]Departamento de Ingeniería Electrónica, Facultad de Ingeniería, Universidad Sur colombiana, Neiva, Huila, Colombia
E-Mail: hmontiela@udistrital.edu.co

ABSTRACT

This paper proposes a strategy that increases the efficiency of the communication established in a system of multiple robots with an asynchronous algorithm that detects the occupation level of the transmission bus. Multiple-robot systems with several operational levels sometimes implement mechanisms to exchange information to avoid conflict or damage between these levels during operation. Normally, these mechanisms synchronize all robots, tools, or accessories in the system and are composed of a physical layer, a sender, and a receiver; where the message to be sent from one robot to another is encrypted using a protocol to increase the security level of the established link. Although in ideal conditions the transmission of messages is satisfactory and there are no losses of information, in reality, the mechanisms of data transmission that have a high transit of messages do not reach an efficiency of one hundred percent, because, the transmission buses or communication channels are limited by the bandwidth, the environmental conditions or the speed of processing of each individual robot.

Keywords: robot, controller area network, multiple robot systems, communication network, central pattern generators, zigbee.

1. INTRODUCTION

On an industrial level, robotic systems have become an indispensable tool in manufacturing processes, since the multifunctional and adaptive nature of a robot allows it to execute one or more tasks. In addition, these systems offer advantages, such as; reducing the time of manufacture of a product, the amount of personnel required in a manufacturing line and the margin of error during the execution of repetitive tasks, which increases the productivity of a process [1-2].

In addition, the massification of robotic systems has allowed the development of robots for almost all types of applications, as well as research into problems such as the movement of loads, the delivery of mail, the construction of routes or the detection of anti-personnel mines. However, the applications of robots are still being explored, because there is no technology that is one hundred percent efficient and has an absolute level of autonomy [3-4].

Considering that the level of autonomy of a single robot limits its ability to solve a problem, strategies have been devised to increase this capacity. One of these strategies includes the creation of Multi-Agent Systems (MAS), which are systems based on individual programs with a certain level of knowledge to interact among them or with their environment. It is worth saying that, Multiple Robot Systems (MRS) are one of the representations of MAS in robotics [5-8].

Analogous to the definition of MAS, an MRS is a set of individual robots that interact with each other or with their environment to achieve a goal. This type of robot has also become popular and diversified. So much so that, one of the events that has driven the development of MRS is the Robot Soccer (or RoboCup) competitions, in which developments are presented that range from simulation (individual agents recreate game skills), to games between humanoid robots (which recreate game techniques) [9].

As you can see, MRSs are not only assemblies of robots assembled in the same structure, but they are also assemblies of individual robots. For example, swarms of mobile robots that are platforms with individual wheels that synchronize with each other to go from one point to another, where a leading robot guides the others along its path, or there are also modular chain-type robots that have two or more robots in their structure, whose kinematic chain changes depending on the number of robots available in their structure [7-9].

Coordinating a set of robots or tools in an MRS is not an easy task, since, as the number of elements in the MRS increases, so does the complexity of generating a coordinated motion pattern. Although, this issue has been partially solved with the development of Central Pattern Generators (CPG) or by recreating patterns with phases similar to those of human motion, there are other limitations associated with the operation of MRS, such as; the reduction of the operating speed by increasing the robustness of the control algorithm or the loss of information when there is a high transit of messages during communication between the robots or tools that make up the MRS [10-11].

This article addresses one of the limitations mentioned above and proposes a partial solution with an asynchronous message or information packet flow control algorithm, which allows each robot in an MRS to coordinate its own control scheme and information sending and receiving functions. This contribution is presented in the following sections, which are organized as follows: section 2 gives a general description of the concepts related to MRS communications, section 3 describes the development of the algorithm presented, and section 4 presents the results obtained with this algorithm.

2. METHODOLOGY

Packet transit in MRS is an extensive subject and covers a wide range of concepts. Next, this topic is presented in general and focused on modular chain-type robots, because it is the type of robot in which the algorithm tests were performed.

2.1 Structure of a Communications Network

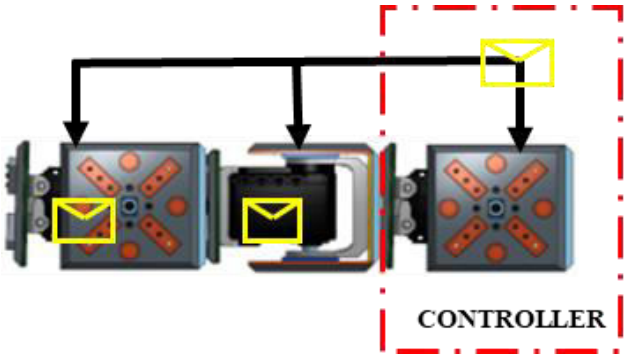
A communications network is a set of equipment arranged in a way that allows them to share information with each other. In addition, this concept is widely used in telecommunications during the construction of computer networks, which are classified depending on their topology, among which are Ring, when the equipment is connected one after the other and at one point the first and last equipment are connected. Star, when all computers are connected to a single receiving device in parallel, that is, each computer has at least one connection point available on the receiving computer. In bus, when all the equipment is connected to a receiving device in serial form, that is, all the equipment is connected using a single port to the receiving equipment [12-14].

Similarly, MRSs implement topologies and adopt network building mechanisms like those contemplated for telecommunications networks. Although the parameters and criteria for evaluating network efficiency are the same in both cases, it must be taken into account that the transmission speed between robots or MRS tools is lower than that of a computer network, since the speed of the control devices of a robot is typically much lower than that of a computer [7-9, 12].

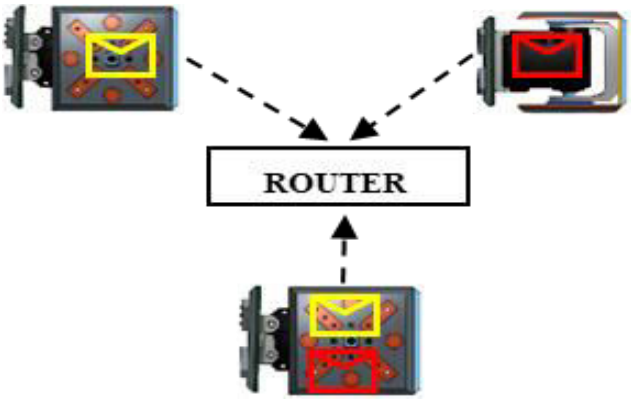
Typically, an MRS implements a wired network with a bus-shaped topology (see Figure-1a), i.e. all control devices are connected to the same pair of conductors to share information among them. Likewise, networks with star topologies (see Figure-1b) are common among MRSs, since there are robot configurations that communicate wirelessly and connect through a router, which oversees addressing the information in the network. However, both network topologies have advantages and disadvantages, e.g. wired networks give the maximum available performance, i.e. transmission speed is not affected by environmental conditions. However, as the length of the data bus increases, so does the likelihood that the signal will be attenuated [9, 11, 14].

As you can see, there are several ways to network devices on an MRS, resulting in ways to set up a network and control each device. On the one hand, the most basic elements of an MRS network (one or more control devices, the connection terminals, one or more devices to be controlled, and the physical transmission medium (wired or wireless)) must have; an address to know the route that the information available in the network must follow, a network connection port, and a protocol for sending and receiving information. On the other hand, the information flow control depends on the control strategy used, since, there are centralized controllers where a single device controls the others and the decentralized controllers where one or more devices control the others (see Figure-1) [11, 15-16].

The proposed design consists of two functional blocks: an electromechanical locking system energized by means of a rechargeable lithium battery and an electronic control system based on the transfer of information via BLE. The proposed prototype seeks to eliminate the use of physical mechanical keys, see Figure-1, both for the opening or closing of the lock, optimizing, personalizing and making safer the device designed compared to other locks found in the market.



a) Network with bus topology and a centralized controller.



b) Network with star topology and a decentralized controller.

Figure-1. Representation of two network topologies and controller models.

Another feature of MRS network is media access control, which is essentially a protocol with which devices exchange information through a common transmission medium. There are several types and forms of media access control, but this article only addresses those based on the protocols; SERIAL, CAN and ZIGBEE [11, 17-18]. In the first case, the SERIAL protocol sends and receives the information sequentially and to establish communication between devices, three wires are shared: transmitter, receiver and ground, from which the transmitter and receiver of one device are connected to the receiver and transmitter of the other respectively, taking into account that the ground of both devices are connected. These wires are available at physical layer level as RS-232 or RS-485, which allow several devices to be connected on the same bus [17].

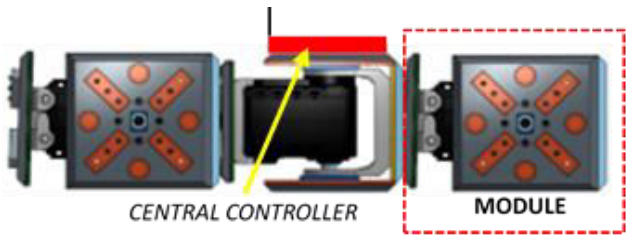
In the second case, the CAN protocol (Controller Area Network) is a protocol that uses a proprietary

controller to filter and condition the messages to be received or sent respectively, which increases the time available to the local controller to execute other tasks. Like the SERIAL protocol, the CAN protocol incorporates a two-wire physical layer where all devices share it and connect in parallel. In the third case, the ZIGBEE protocol is defined as a set of low power wireless communication protocols (based on the IEEE 802.15.4 standard), where each device has its own address and is considered a node (a network can have up to 65535 nodes) in the network. In turn, each network has at least one coordinator who is responsible for guiding the devices to establish a network, a router interconnects the available nodes and the end device is a network node [11, 18]. Finally, the relationship between these three communication protocols with the developed algorithm will be presented in detail in the following section.

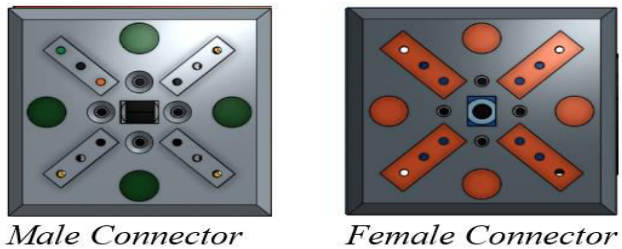
2.2 The EMERGE Robot

The EMERGE robot (see Figure-2a) is a prototype under development, whose elements necessary for its manufacture (mechanical and electrical diagrams, files for the manufacture of printed circuits and test programs) are available in a repository. This robot is modular, and its structure is built with homogeneous modules, which gives the user flexibility to build different robot morphologies. Furthermore, each robot module has its own local controller, which establish a network using the CAN protocol [10-11].

The topology of the network is in the form of a bus that is constructed when the pins (VCC, GND, CANH and CANL) come into contact and surface available on the faces of the modules. This contact is ensured by a male and female connector with magnets (see Figure-2b), which are responsible for securing the modules when the robot is built [10].



a) EMERGE Prototype



b) Connectors

Figure-2. EMERGE Robot.

As it can be seen, the EMERGE robot (see Figure-2) is just a case of an SMR and its typical configuration comprises chain type nickname robot structures, which perform sinusoidal movements using a CPG based on the coupled oscillator model of coupled oscillators. This model incorporates equations 1 and 2 where a central controller sets the parameters (Amplitude=x_iPhase=φ_i and offset=r_i) of oscillation of each module (θ_i or motor position) of the structure and each individual module is synchronized by tuning the weights of the value of the self-oscillation (w_i) and the structure (w_{ij}) and the local result (i) obtained (φ_i) is shared with its neighbors (j) and thus obtain all the coefficients (φ_j = phase of the neighbouring module and θ_{ij} = own exit angle) [11, 19-20].

$$\theta_i = x_i + r_i \cos(\varphi_i) \quad (1)$$

$$\varphi_i = w_i + \sum_i^j w_{ij} * \mathrm{sen}(\varphi_i - \varphi_j + \theta_{ij}) \quad (2)$$

The information generated by the CPG is packaged and sent using the CAN protocol and the packets sent by one module reach the others, from which the receiving module rejects the packets that have not been assigned their address. In addition, the modules occupy the communications bus with a randomly defined time interval (see Algorithm 1) [11].

Algorithm 1. Native message transmission scheme of a module

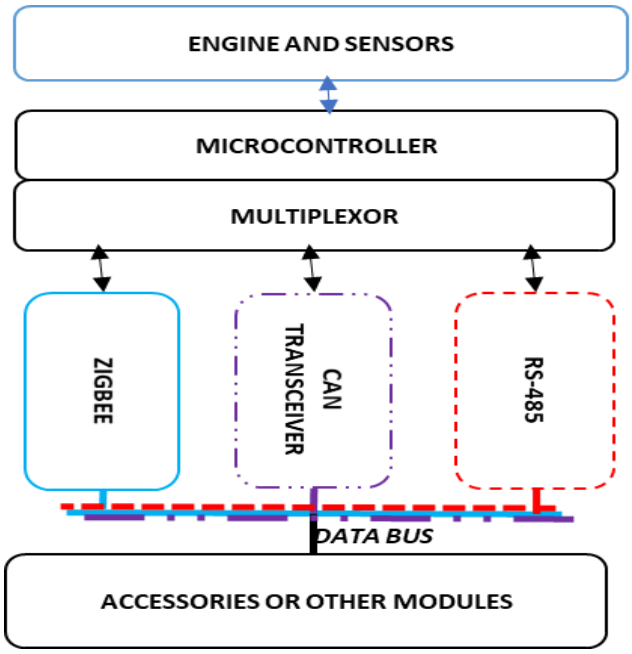
Main **Function** ()
Start local controller variables ()
Start CAN controller ()
Check the number of connected modules ()
Assign module address
Activate interruptions
Delay in milliseconds ~ U [0.1000]
$w_i = 0.5, w_{ij} = 0.8$
While Module on **do**
θ_i= CPG (r_i, φ_i, x_i, w_i, w_{ij})
Move actuator ()
Wait (Delay in milliseconds)
Send a message to the neighbors (CPG parameters, myaddressón + 1, myaddressón − 1)
Function Interrupt message reception ()
String=Scan Input Buffer ()
If String [0] == my address **then**
φ_i=Chain [1]
x_i=String [2]
r_i=Chain [3]
Update (r_i, φ_i, x_i)

The proposed algorithm incorporates to the CGP model a variant in its packaging form, which gives flexibility to form a network between the modules using the SERIAL, CAN and ZIGBEE protocols. This variant is described in detail in the following section.

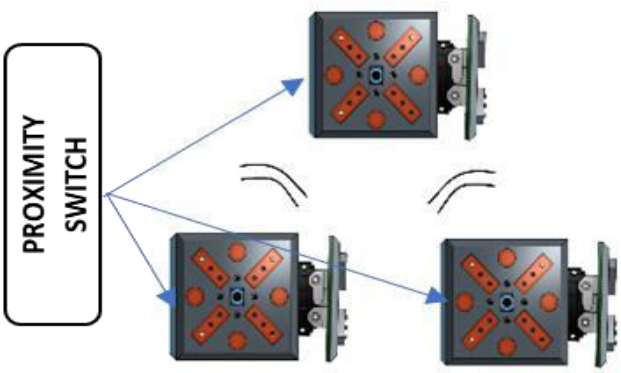
VOL. 17, NO. 11, JUNE 2022 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

3. IMPLEMENTATION

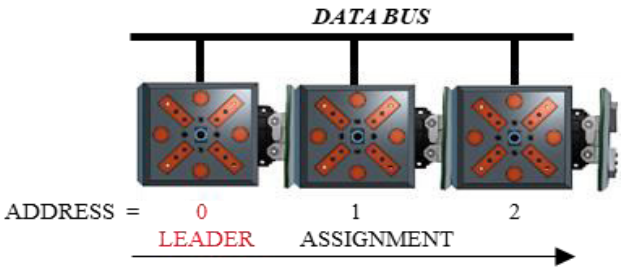
The communication algorithm between modules developed is flexible and was tested with the SERIAL, CAN AND ZIGBEE protocol, whose objective is to transmit the output value of the CPG between the modules without loss of information (Figure-3). Since, it was tested with different communication protocols from the native one, the electronic design of each module of the EMERGE robot was modified, to add a peripheral that establishes a RS-485 network (using the pins destined for the CAN communication) and a terminal to connect a ZIGBEE module.



a) Representation of the electronic diagram of each module



b) View of the wireless network with behind modules



c) View of the conventional network with a three-module chain

Figure-3. Topology of EMERGE robot networks.

Considering that the modification of the connection terminals does not alter the morphology of the robot when forming kinematic chains, Algorithm 2 was tested with a chain of three modules (see Figure-2c). This chain does not require an external centralized controller as conventionally proposed [11], but all modules incorporate a safety routine that gives them the ability to inherit the leader function.

Each module has a proximity sensor (see Figure-3b) on its faces and in this case, when the sensor on the front face is not active, the module automatically assigns itself a zero address (see Figure-3c) and sends a message to the neighboring module to update its address with respect to the previous module, which sends a repeat of the auto-addressing routine and so on until all the modules in the chain have auto-assigned an address with respect to the activation of the proximity sensor.

Once automatic addressing is completed, movement is initiated using the CPG model, the output of which is shared across the communications bus using a three-step sequence. The first step is to package the output of the CPG into a variable with the address of the recipient and the address of the sender. The second step consists of waiting until the data bus is available to send a message, i.e. the timed sending method was eliminated (see Algorithm 1). The third step consists in opening the received message and verifying if the recipient's address is the own address and the sender's address is the neighboring module.

In other words, the modules use the data bus only if it is available in order to limit the number of messages that can be sent simultaneously. However, communication is a totally stochastic process, since, when there are intervals of inactivity on the bus; a randomly selected module sends a message (see Algorithm 2).

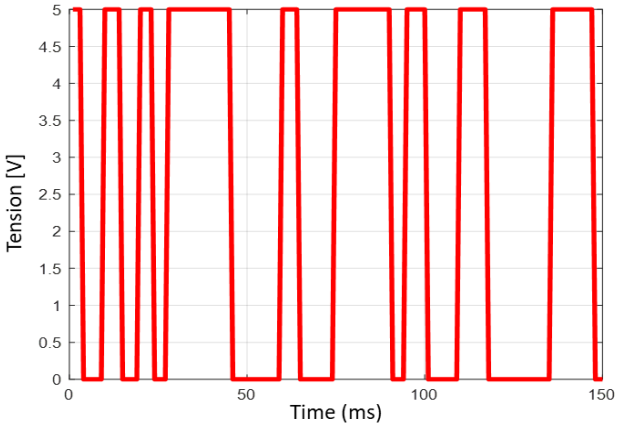
Algorithm 2. Modified message transmission scheme of a module

Main Function()
Start local controller variables ()
Start communication interface ()
Check proximity sensor status ()
My address = Auto-assign address ()
Activate interruptions ()
$w_i = 0.5, w_{ij} = 0.8$
Set overflow interruption to one second
While Module on **make**
θ_i= CPG (r_i, φ_i, x_i, w_i, w_{ij})
Move actuator ()
Function Bus interruption available ()
Message = Build message (my address,r_i, φ_i, x_i, θ_i)
Send Message
Function Overflow interruption ()
Indicator~U[0.1]
If Indicator<0.5 **then**
Message = Build message (my address,r_i, φ_i, x_i, θ_i)
Send Message
Function Interrupt message reception ()
String=Scan Input Buffer ()
If String [0] == my address **and** String [1] == my address +1 **then**
φ_i=String [2]
x_i=Chain [3]
r_i=Chain [4]
Update (r_i, φ_i, x_i)
Function Build message (my address,$\mathbf{r_i}$, $\boldsymbol{\varphi_i}$, $\mathbf{x_i}$, $\boldsymbol{\theta_i}$)
String [0] =my address+1
String [1] =my address
String [:] = Package CPG parameters
Send CPG parameters (String)

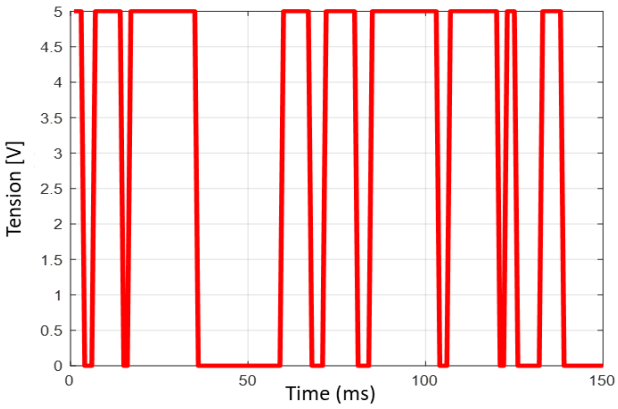
Finally, two hardware configurations were made to measure the effectiveness of Algorithms 1 and 2. In the first one, a data acquisition card (NI DAQ 6008) was connected to the data bus with a configuration that implements the CAN protocol to see the behavior of the messages circulating in the network. In the second one, an accessory (CAN SNIFFER, RS485 and XBEE at a transmission speed of 512kbps, 115200 bauds and 57600 bauds respectively) was connected to the robot's network to verify the number of messages sent to all the modules using the three mentioned protocols. The results obtained in the tests are presented in the following section.

4. RESULTS AND DISCUSSIONS

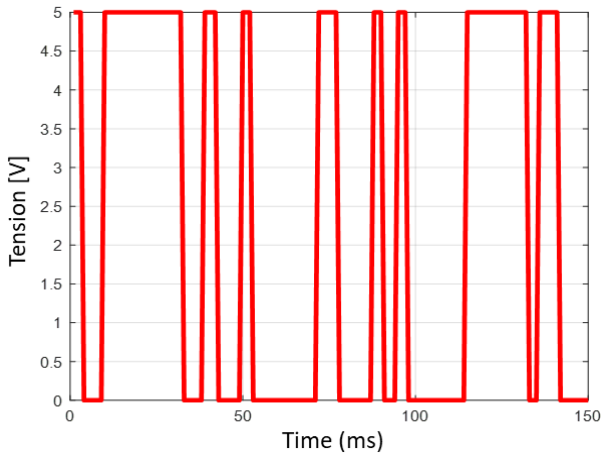
Initially, the communication bus terminals (in CAN configuration) of each individual module were connected to the NI DAQ, then each module was programmed with a message sequence and the shape of the signal representing the messages was recorded in a plain text file whose graphic by module is shown in Figure-4. Then, the kinematic chain of three modules was assembled and the NI DAQ was connected to the conformed data bus to record the shape of the data frame seen by the robot when implementing Algorithms 1 and 2 (see Figure-5). Given the transmission speed of the messages (512 Kilo bauds per second) and the sampling frequency of the NI DAQ (1 MHz), files were constructed with around 120000 samples during each recording session, so, the information presented corresponds to a segment of the collected information.



a) Module 1

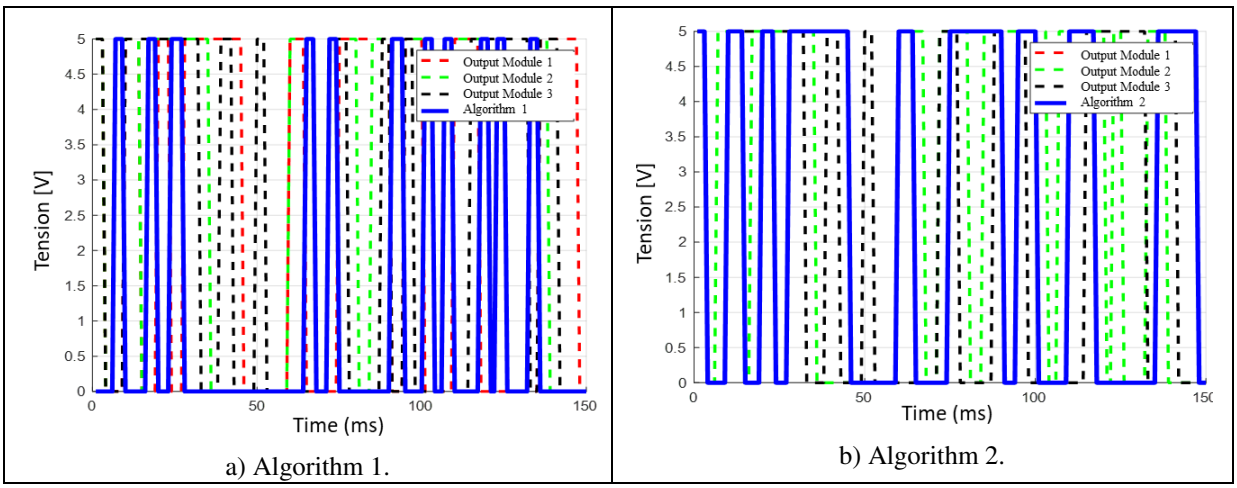


b) Module 2



c) Module 3

Figure-4. Segment of a sequence of messages sent by 3 modules through a CAN bus.



a) Algorithm 1. b) Algorithm 2.

Figure-5. Superposition of the graphics obtained with the NI DAQ individually and the output observed when incorporating algorithm 1 and 2 in the robot.

The second experiment consists of connecting the sniffer to the powertrain and monitoring the messages with information from the CPG that each module sends to the others. In this case during two (2) minutes and five (5) executions of Algorithm 1 and 2. In addition, the sniffer performs the following functions; it monitors the data bus, waits until each module sends 100 messages and records the number of messages received and lost by each module (Tables 1, 2 and 3). Finally, the interpretation of the results obtained is presented in the following section.

Table-1. Average and error margin of the number of sent, received and lost messages using algorithm 1 and 2 of module 1.

	Algorithm 1			Algorithm 2		
	Submitted	**Received**	**Lost**	**Submitted**	**Received**	**Lost**
SERIAL	100	65±4.04	35±4.04	100	80±2.04	20±2.04
CAN	100	70±3.06	30±3.06	100	82±1.55	18±1.55
ZIGBEE	100	55±6.08	45±6.08	100	62±7.05	38±7.05

Table-2. Average and error margin of the number of sent, received and lost messages using algorithm 1 and 2 of module 2.

	Algorithm 1			Algorithm 2		
	Submitted	**Received**	**Lost**	**Submitted**	**Received**	**Lost**
SERIAL	100	60±3.09	40±3.09	100	78±1.55	22±1.55
CAN	100	65±5.08	35±5.08	100	75±2.07	25±2.07
ZIGBEE	100	38±10.6	62±10.6	100	68±8.07	32±8.07

Table-3. Average and error margin of the number of sent, received and lost messages using algorithm 1 and 2 of module 3.

	Algorithm 1			Algorithm 2		
	Submitted	**Received**	**Lost**	**Submitted**	**Received**	**Lost**
SERIAL	100	45±10.9	55±10.9	100	72±6.08	28±6.08
CAN	100	35±7.88	65±7.88	100	79±4.67	21±4.67
ZIGBEE	100	28±9.53	72±9.53	100	62±9.75	38±9.75

5. CONCLUSIONS

Figure-2c presents the kinematic chain formed for the realization of the tests reported in this article, since, as the number of modules increases, so does the difficulty in collecting and filtering the information of the results in each experiment. However, the shape of each modified module and Algorithm 2 ensure that kinematic chains of more than three modules work properly since self-addressing supports up to 254 device configurations. In addition, incorporating a wireless network gives the robot a functional advantage, which is that each module can operate separately from the conformed structure.

The graph in Figure-4 shows a plot of the messages sent by three modules of the robot, which when superimposed on the results (Figure-5) obtained, two completely different phenomena are observed. On the one hand, in the graph of Figure-5a the solid line shows that when the messages cross (dotted lines) the signal attenuates or disappears, that is, when several modules try to transmit at the same time it is possible that; there is a failure in the communication that the received information is wrong or is lost. On the other hand, in the graph of the Figure-5b it is observed that the output of the algorithm follows the message of the module 1, which indicates that when controlling the flow of the messages the information is readable and available in the data bus for the receiver to interpret it. In addition, the reduction in information loss is because the only way two modules transmit at the same time depends on a random function defined in the overflow interruption.

Finally, as shown in Tables 1, 2 and 3 when the CAN protocol is implemented there is a 5 to 10 percent reduction in message loss compared to the other protocols. However, the robotic structure loses speed during the execution of the CPG routine, since the CAN protocol does not reach such a high speed as the SERIAL protocol, which indicates that in terms of performance the SERIAL protocol is still a good option for the MRSs. Although, in this case, the CAN and SERIAL protocols report the best results, they take away flexibility from the structure because the morphology of the robot depends on the electrical conductors that make up the data bus, which can still be scanned using ZIGBEE technology.

REFERENCES

[1] J. Guhl, S. Tung and J. Kruger. 2017. Concept and architecture for programming industrial robots using augmented reality with mobile devices like microsoft Holo Lens. 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Limassol. pp. 1-4.

[2] J. Golz, T. Wruetz, D. Eickmann and R. Biesenbach, 2016. RoBO-2L, a Matlab interface for extended offline programming of KUKA industrial robots. 2016 11th France-Japan & 9th Europe-Asia Congress on Mechatronics (MECATRONICS) /17th International Conference on Research and Education in Mechatronics (REM). pp. 064-067.

[3] J. Kim. 2020. Multi-Axis Force-Torque Sensors for Measuring Zero-Moment Point in Humanoid Robots: A Review- IEEE Sensors Journal. 20(3): 1126-1141.

[4] A. Saudabayev and H. A. Varol. 2015. Sensors for Robotic Hands: A Survey of State of the Art. IEEE Access. 3: 1765-1782.

[5] M. A. Mahmoud and M. S. Ahmad. 2015. A self-adaptive customer-oriented framework for intelligent strategic marketing: A multi-agent system approach to website development for learning institutions. 2015 International Symposium on Agents, Multi-Agent Systems and Robotics (ISAMSR). pp. 1-5.

[6] P. Dasgupta. 2012. Multi-agent coordination techniques for multi-robot task allocation and multi-robot area coverage. 2012 International Conference on Collaboration Technologies and Systems (CTS). pp. 75-75.

[7] H. Kang, H. Kim and Y. M. Kwon. 2019. RECEN: Resilient MANET Based Centralized Multi Robot System Using Mobile Agent System. 2019 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 1952-1958.

[8] Y. Tao, T. Li, S. Liu, T. Deng and D. Xin. 2018. Research of Universal Modular Cooperation Robot Control System. 2018 2nd International Conference on Robotics and Automation Sciences (ICRAS). pp. 1-5.

[9] S. Behpour and M. Golparyan. 2013. Simulation robocup by agent-oriented methodology: Prometheus. 2013 3rd Joint Conference of AI & Robotics and 5th RoboCup Iran Open International Symposium. pp. 1-5.

[10] Rodrigo Moreno, Frank Veenstra, David Silvera, Julian Franco, Oscar Gracia, Ernesto Cordoba, Jonatan Gomez and Andres Faina. 2018. Automated Reconfiguration of Modular Robots Using Robot Manipulators. Proceedings of IEEE Conference on Evolvable Systems, part of the 2018 Symposium Series on Computational Intelligence (IEEE-SSCI 2018). pp. 18-23. In Press

[11] Henry Hernández, Rodrigo Moreno, Andres Faina, and Jonatan Gómez. 2018. Design of a bio-inspired controller to operate a modular robot autonomously.

Proceedings of the 16th Ibero-American Conference on Artificial Intelligence. pp. 13-16.

[12] V. Kaptur and E. Mammadov. 2015. Methodology of selecting appropriate technologies for constructing telecommunication access networks. 2015 Second International Scientific-Practical Conference Problems of Info communications Science and Technology (PIC S&T). pp. 90-92.

[13] Li Xinzhan and Wang Li. 2009. A method to support auto adaptive access network. International Conference on Industrial Mechatronics and Automation. pp. 346-349.

[14] D. Martin and C. Dickey. 2014. Hardware-in-the-loop for power and telecommunications co-simulation with applications. ISGT 2014. pp. 1-5.

[15] K. Yaovaja and J. Klunngien. 2019. Teleoperation of an Industrial Robot using a Non-Standalone 5G Mobile Network. 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE). pp. 320-323.

[16] H. Leon-Rodriguez, T. Sattar and J. Park. 2013. Wireless climbing robots for industrial inspection. IEEE ISR 2013. pp. 1-4.

[17] S. S. B. K. Mohideen and S. B. M. Noor. 2009. Improvement of data refresh rate for dual serial port to Universal Serial Bus acquisition system. 2009 IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurements Systems. pp. 327-329.

[18] G. Pan, J. He, Q. Wu, R. Fang, J. Cao and D. Liao. 2018. Automatic stabilization of Zigbee network. 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD). pp. 224-227.

[19] Ceyue Liu, Jiangong Liu, Rodrigo Moreno, Frank Veenstra and Andres Faina. 2017. The impact of module morphologies on modular robots. ICAR 2017.

[20] M. Wang, X. Li, H. Dong and S. Yang. 2017. A PSO-based CPG model parameter optimization method for biomimetic robotic fish. 2017 Chinese Automation Congress (CAC). pp. 3445-3449.