Design of CAN-Based Distributed Control Systems with Optimized Configuration

Eduardo P. Godoy
epgodoy@yahoo.com.br

Rafael V. de Sousa
rafael@cnpdia.embrapa.br

Arthur José V. Porto
ajvporto@sc.usp.br
EESC – São Carlos School of Engineering
USP – University of São Paulo
Department of Mechanical Engineering
Av. Trabalhador São Carlense, 400
13566-590 São Carlos, São Paulo, Brazil

Ricardo Y. Inamasu
ricardo@cnpdia.embrapa.br
EMBRAPA – Brazilian Agricultural Research Corporation
CNPDIA – Agricultural Instrumentation
Rua XV de Novembro, 1452
13560-970 São Carlos, São Paulo, Brazil

A current trend in distributed control systems is the application of communication network technologies such as CAN – Controller Area Network. A recent utilization approach of these technologies is the networked control systems (NCS). The fundamental challenges in the development of NCS are the analysis of the network delay effects and the prediction of the timing behavior of the distributed control system. The common parameters that impact the performance of NCS include response time, network utilization and network delays induced by the communication of messages between the devices. In addition, the performance of a NCS is highly dependent on these messages sampling times. A significant emphasis has been put on development and application of methodologies to handle the network delay effect in these systems and improve their performances. This paper presents a detailed timing analysis and a mathematical model to calculate these network delays in CAN-based networks. With the results of this model, the application of a methodology is proposed to minimize the effects of these delays and to achieve the optimization (network operation and utilization) of a CAN-based network. A case study of a CAN-based distributed control system in a mobile robot is described to demonstrate the application of the optimization methodology and the utilization of the CAN mathematical model systemized.

***Keywords**: CAN protocol, mathematical model, network time delays, optimization methodology*

Introduction

Recent applications of distributed control systems demonstrate a new approach for the use of industrial networks. In this approach, called Networked Control Systems (NCS), the controller and the plant are physically separated and connected through a communication network (Yang, 2006). Network control systems can replace the traditional centralized and point-to-point control systems and provide several benefits such as reduced amount of wiring, increased reliability, improved capacity for system reconfiguration and ease of maintenance (Moyne and Tilbury, 2007).

Extensive researches and developments have released several protocols for NCSs (Moyne and Tilbury, 2007). A preferred option is the Controller Area Network (CAN) (Bosch, 2006) which was originally developed to be used in automotive area to interconnect electronic control units, but recently has also been considered in many other networked control applications (Othman et al., 2006). This preference is due to its low cost of development and large acceptance in the industrial and academic areas. Currently, CAN-based networks are applied in distributed systems in several areas such as robotic control, automated manufacturing and process control environments, and used in industrial proprietary architectures such as Device Net and CAN Open (CIA, 2006).

Despite the advantages and potentials, the existence of communication networks make the analysis and design of a NCS complicated (Baillieul and Antsaklis, 2007). One of the fundamental challenges in this area is the development of analysis to predict the timing behavior of the networked control system (Cervin et al., 2003). The temporal behavior of the NCS must be confirmed during the design phase by performing a timing analysis such as a schedulability test and response time of tasks and messages. Many works in recent years has been developing this analysis for different types of networked control systems and protocols (Lian, Moyne and Tilbury, 2001). In the case of CAN, this type of analysis and discussion first appears in Tindell, Burns and Wellings (1995). From this paper, several others studies used, modified and incremented the equations proposed in different researches. Jeon et al. (2001) use the equations to analyze the communication delays and latency in CAN networks and Punnekkat, Hansson and Norstrom (2000) provide additional terms to the equations to analyze CAN networks in presence of message transmission errors. In addition, the work described in Davis et al. (2006) revises the original equations and considers some modifications to deal with the priority assignment policy in CAN networks, and Nolte (2007) investigates the effects of the bit stuffing in CAN networks and adapts the equations to obtain more realistic values for the bus utilization rate.

Regardless of the type of network used, the overall NCS performance can be affected by network delays. The network induced delays and data packet dropouts occur when sensors, actuators, and controllers exchange data across the network. Delays can degrade the stability and the performance of a control system (Tipsuwan and Chow, 2003). Other problems related to the NCS are the correct choice of the network configuration parameters such as the network bandwidth, messages data length and messages sampling time. These parameters also influence the network performance (Lian, Moyne and Tilbury, 2002). The NCS must take sample and transmit data at a sampling time appropriate to achieve required performance metrics. However, if this sampling time is higher than the network bandwidth available, the network becomes overloaded, originating additional network delays and jitter, and causing packet losses and errors transmissions (Al-Hammouri, Branicky and Liberatore, 2008). Networked systems generally must meet two main criteria: bounded time delay and guaranteed transmission, that is, a message should be transmitted successfully within a bounded time delay. Unsuccessfully transmitted or large time-delay messages from a sensor to an actuator may lead to deterioration of system performance or make it unstable. For a NCS, performance is a function of not only the messages sampling times, but also of the traffic load on the network.

Recently, much research work has been done on the performance analysis and design of methodologies for NCSs (Tipsuwan and Chow, 2003; Hespanha, Naghshtabrizi and Xu,

Paper accepted February, 2010. Technical Editor: Glauco A. de P. Caurin.

2007; Goodwin, Quevedo and Silva, 2008). A methodology is required to handle network time delays in control systems over a network, such as CAN and improve their performance. The work of Tipsuwan and Chow (2003) demonstrates and describes several of these control methodologies. Among them, the robust control methodology Goktas (2000) designs a networked controller in the frequency domain using robust control theory that does not require any kind of information about the network delays in the control system. The fuzzy logic modulation methodology (Almutairi, Chow and Tipsuwan, 2001) considers a controller for an NCS with a linear plant and a modulated PI controller to compensate the network delay effects. In this methodology, the PI controller gains are externally updated at the controller output with respect to the system output error caused by network delays. The optimal stochastic methodology (Hu and Zhu, 2003) presents a controller that has good performance to treat the effects of random network delays in an NCS. The sampling time scheduling (Hong and Kim, 2000) presents a procedure to appropriately select a sampling time for the messages of an NCS over a CAN-based network such that the effects of the network delays in the performance of the system are minimized.

Therefore, to design a NCS with a certain networked methodology, a designer has to clearly know about the data requirements to apply the methodology under a selected network protocol. Based on these facts, this work presents the application of one of the previously cited methodologies, called Sampling Time Scheduling (STS), to handle the induced network delays in the performance of a CAN-based distributed control system proposed in a mobile robot case study. A CAN mathematical model systemized is used to calculate the required data for the methodology application. In the case study, the methodology is applied to ensure the correct choice of the messages sampling times of the devices connected in the robot CAN network for the optimization (network operation and utilization) of the distributed control system studied.

Analysis of CAN-Based Networks

As described in Farsi, Ratcliff and Barbosa (1999), CAN is a serial communication protocol developed mainly for applications in the automotive industry, but is also capable of offering good performance in other time-critical industrial applications. In a CAN-based network, data are transmitted and received using message frames that carry data from a transmitting node to one or more receiving nodes. An identifier that is unique throughout the network labels each message and its value defines the priority of the message to access the network.

The CAN protocol is optimized for short messages and uses a CSMA/CD with NDBA (Carrier Sense Multiple Access / Collision Detection with Non-Destructive Bitwise Arbitration) arbitration access method. The bit stream of a transmission is synchronized on the start bit, and the arbitration is performed on the following message identifier, in which a logic zero is dominant over a logic one. The CAN protocol supports two message frame formats: standard CAN (version 2.0A, 11-bit identifier) and extended CAN (version 2.0B, 29-bit identifier), as shown in Fig. 1.

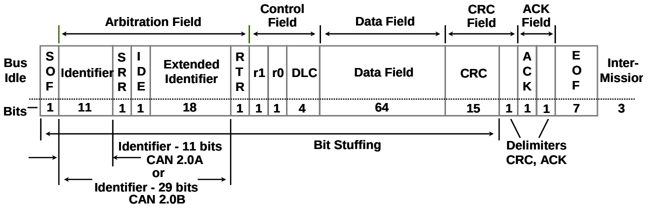


Figure 1. Message frame formats defined by the CAN protocol (Godoy, 2007).

Timing Analysis

The time delays of a CAN-based network can be analyzed by studying their timing parameters (Lian, Moyne and Tilbury, 2002). Figure 2 shows a general timing diagram of the initialization and ending of the task of sending a CAN message over the network. The total time delay of a message, *Tdelay*, is defined as the difference between the time when the source node begins the process of sending a message, *Tsrc*, and the time when the destination node completes reception of this message, *Tdest* (i.e., *Tdelay = Tsrc – Tdest*).

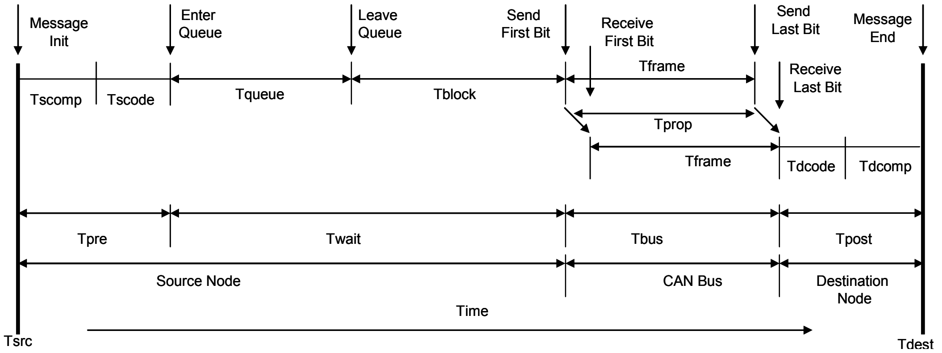


Figure 2. Timing diagram of the transmission of a CAN message (Modified from Lian, Moyne and Tilbury, 2001).

The total time delay, *Tdelay*, can be broken into three parts: time delays at the source node, *Tsrc*, on the CAN network, *Tbus*, and at the destination node, *Tdest*, as shown in Fig. 2. The *Tsrc* node includes the preprocessing time, *Tpre*, which is the sum of the computation time, *Tscomp* with the encoding time, *Tscode*. The waiting time, *Twait*, is the sum of the queue time, *Tqueue*, and the blocking time, *Tblock*. Depending on the amount of data that the source node must send and the traffic on the network, the waiting time may be significant. The queuing time, *Tqueue*, is the time that a message waits in the buffer at the source node while previous messages in the queue are sent. It depends on the blocking time of previous messages in queue, the periodicity of messages, and the processing load. The CAN network time delay, *Tbus*, includes the total transmission time of a message, *Tframe*, and the propagation delay of the network, *Tprop*. This will depend on message size, data rate, and the length of the network cable. The *Tdest* is the post processing time, *Tpost*, which is the sum of the decoding time, *Tdcode*, and the computation time, *Tdcomp*, at the destination node.

The *Tpre* and the *Tpost* times are typically constant and small. These times are totally related and depend on the characteristics of the software and hardware used in the network. Because of these facts, a relation of them with the network *Jitter* (variation of waiting time of the message at the transmission queue before scheduling) was adopted. Finally, the Eq. (1) can express the total time delay that can be found in the transmission of a message over the CAN network.

$$\begin{aligned} T_{delay} &= T_{src} - T_{dest} \\ &= T_{pre} + T_{wait} + T_{bus} + T_{post} \\ &= \underbrace{T_{pre} + T_{post}}_{J_m} + \underbrace{T_{wait}}_{Q\ m} + \underbrace{T_{bus}}_{C\ m} \end{aligned} \tag{1}$$

Mathematical Model Systematization

According to the review, the equations and analysis of CAN networks firstly appear in Tindell, Burns and Wellings (1995). However, several other papers have used, incremented and adapted the proposed equations to include more characteristics of CAN networks and to do your own researches (Punnekkat, Hansson and Norstrom, 2000; Davis et al., 2006; Nolte, 2007). Godoy (2007) presents a revision about these equations and systemizes a CAN mathematical model to calculate performance parameters such as CAN bus utilization rate, network delays, and message transmission times. In the paper (Godoy, 2007), the CAN mathematical model is implemented in simulation software to ease the analysis of CAN networks and the model is verified and validated. The verification and validation processes consisted in the performance parameters comparison between the data obtained by the application of the simulation software and the data obtained using a real CAN network in laboratory.

The parameters shown in Eq. (1) are described for CAN-based networks resulting in a relation between the Eq. (1) and the Eq. (2) (Godoy, 2007).

$$R_m = J_m + Q_m + C_m \tag{2}$$

where the term J_m represents the message *Jitter* and is empirically determined. In the present work, we use the value 0.1 ms (millisecond). The term Q_m corresponds to the time spent by a message (*m*) in the waiting queue under error conditions and the term C_m represents the transmission time needed to physically transmit a message (*m*) over the CAN network.

The CAN data frame of a message (*m*) contains an amount of *O* bits of overhead and a bit-stuffing width of 5 bits. Only an amount of *T* of the *O* bits suffers stuffing (bit stuffing area shown in Fig. 1). The values of the variables *T* and *O* are different for the CAN message types. Godoy (2007) defines for CAN 2.0A, the values of 34 and 47 for *T* and *O*, and for CAN2.0B, the values of 54 and 67 respectively. Finally, the Eq. (3) represents C_m.

$$C_m = \left(\left\lfloor \frac{T + 8S_m - 1}{4} \right\rfloor + O + 8S_m \right) . \tau_{bit} \tag{3}$$

T = the amount of overhead bits subject to bit stuffing;

O = maximum number of overhead bits per message;

S_m = the size of a message *m* (8 bytes maximum);

τ bit = time spent to transmit one bit over the bus (founded with the CAN bus speed: bits per second).

The waiting time of a CAN message (*m*) in the transmission queue depends on the lower priority blocking time of the message that occupied the bus in the moment (B_m) and of the time needed to retransmit messages that presented error transmission (E_m). It depends too on the higher priority messages transmission times. This time (Q_m) is given by the recurrent relation presented in Eq. (4) with initial value $Q_m^0 = 0$ and iterations until convergence (i.e., $Q_m^{n+1} = Q_m^n$).

$$Q_m^{n+1} = B_m + \sum_{\forall j \in hp(m)} \left\lceil \frac{Q_m^n + J_j + \tau_{bit}}{T_j} \right\rceil . C_j + E_m(Q_m + C_m) \tag{4}$$

hp (m) = set of messages with higher priority than *m*;

B_m = the longest time that the given message *m* can be delayed by lower priority messages, given by Eq. (5);

Tj = the sampling time of a given message *j*;

Jj = the Jitter of a message *j*.

$$B_m = \max_{\forall k \in lp(m)} (C_k) \tag{5}$$

lp (m) = set of messages with lower priority than message *m* (if a message *m* is the lower priority message, B_m is equal to zero). The values of *Cj*, *Ck* are the same of *Cm* calculated by the Eq. (3).

In the Eq. (4) the term *E*$E_m(t)$ is defined as a function of the error recovery and gives the superior limit of all overhead of error recovery that can occur in a time interval *t*. It is given by the Eq. (6).

$$E_m(t) = \left(n_{error} + \left\lceil \frac{t}{T_{error}} \right\rceil - 1 \right) . (31 . \tau_{bit} + \max_{\forall K \in hp(m) \cup \{m\}} (C_{k)}) \tag{6}$$

n_{error} = number of errors that can occur in an arbitrary interval;

T_{error} = the period of error occurrence.

The values of n_{error} = 1 and T_{error} = 100 ms are defined as described in Tindell, Burns and Wellings (1995) for utilization in Eq. (6). In every error that occurs in the network, the overhead of the error recovery can be increased in 31 bits, followed by a CAN message retransmission. Only messages with higher priority than message *m* can be retransmitted and cause a delay in message *m*. The biggest of those messages is given by Eq. (7).

$$\max_{\forall K \in hp(m) \cup \{m\}} (C_k) \tag{7}$$

Other parameter used to evaluate the performance of the CAN-based network is the network utilization rate (Lian, Moyne and Tilbury, 2002). The equation that defines this parameter is presented in Eq. (8).

$$U = \sum_{i=1}^{N} \frac{Ci}{Ti} \tag{8}$$

Ci = transmission time of the message *i* and given by (3);

Ti = sampling time of the message *i*;

N = the total number of messages in the CAN network.

The set of equations presented constitutes the mathematical model systemized and validated (Godoy, 2007) that can be used to analyze and evaluate performance of CAN-based networks.

Optimization of CAN-Based Networks

Relation between Sampling Time and Performance

Data exchange by sensors and actuators through one networked control system induce unavoidable data latency and might degrade the control performance. In particular, Lian, Moyne and Tilbury (2002) define the relationship between the sampling times and the performance of a networked system as one of the main concepts to study and analyze NCS such as CAN. As presented in Moyne and Tilbury (2007), Fig. 3 shows a diagram to assist the selection of these design parameters and to visualize overall CAN-based system performance at different sampling and transmission. To understand the diagram, Moyne and Tilbury (2007) explain that the worst, unacceptable, acceptable, and best regions can be defined based on required control system specifications such as response time and/or phase margin. For the networked control case, point 1 can be determined by further investigating the characteristics and statistics of network-induced delays and device processing time delays. Smaller sampling times may be needed to guarantee a certain level of control performance, but the network will present high level of idleness. As the sampling time gets faster, the network traffic load becomes heavier and the idleness reduces, the possibility of more contention time or data loss increases in a bandwidth-limited network, and longer time delays result. Point 2 is the situation when the network is becoming saturated.

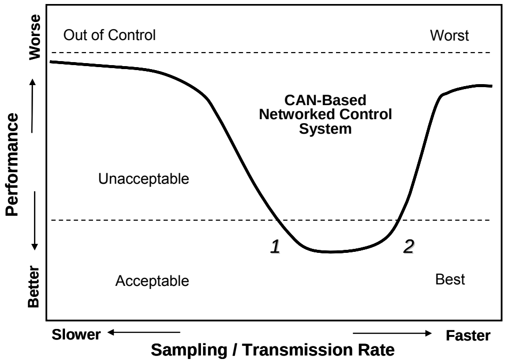


Figure 3. Relation diagram of control performance and sampling time in CAN networked control systems (Modified from Moyne and Tilbury, 2007).

Selected Methodology: Sampling Time Scheduling

Methodologies for optimization of the distributed control systems have to maintain the stability of the system in addition to controlling and maintaining the system performance as much as possible (Tipsuwan and Chow, 2003; Hespanha, Naghshtabrizi and Xu, 2007; Goodwin, Quevedo and Silva, 2008). The methodologies have been formulated based on several types of network behaviors and configurations in conjunction with different ways to treat the delay (Yang, 2006). The concepts of one of these methodologies called Sampling Time Scheduling (STS), to achieve optimization and communication performance in a CAN networked control system, were chosen to be applied in the case study presented in this paper.

As described in Hong (1995), the STS methodology is used to appropriately select a sampling time for the devices of a distributed control system such that network delays do not significantly affect the control system performance and the system remains stable. This methodology was originally developed for multiple systems on a periodic delay network. However, it was also modified to apply to random delay networks such as CAN (Hong and Kim, 2000).

The STS methodology defines the number of electronic control units on the network as *M*. The sampling times of all *M* devices on the network can be calculated from the sampling time of the most sensitive device based on the analysis of its worst-case delay bound. The most sensitive device, denoted as *ECU$_1$*, has the shortest delay bound defined as φ_1. The STS methodology is formulated from the window concept illustrated in Fig. 4 where *L* and σ are the transmission times of a pure data message and its overhead, respectively; T_1 is the sampling time of *ECU$_1$* and *r* is the number of data messages of the *ECU$_1$* that can be served by the network during the worst-case network traffic. The sampling time T_1 is computed from Eq. (9).

$$T_1 = \frac{\varphi_1 + L}{3} \tag{9}$$

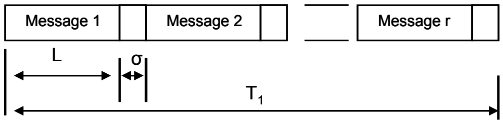


Figure 4. Window of data transmissions of the sampling time T1 in the methodology (Tipsuwan and Chow, 2003).

As described in the methodology, the other devices on the same network have to be indexed from the worst-case delay of the systems in an ascending order as *ECU2, …, ECUM*. The sampling times of *ECU2, …, ECUM* are determined from T_1 using different rules with respect to network conditions. In a generic case, all other sampling times are multiples of T_1 as expressed by Eq. (10).

$$T_i = k_i T_1, k_i = \wedge\left[\frac{\varphi_i - (T_1 - L)}{2T_1}\right] \rightarrow i = 2,3...M \tag{10}$$

where $a = \wedge(b)$ indicates that $a = 2^{v_i}, v_i \in \{0,1,...\}$, which is the closest to, but does not exceed *b*.

In addition, with the correct definition of the sampling times values of the devices, the optimality operation of the network can be achieved by this methodology, which is an advantage over other methodologies (Tipsuwan and Chow, 2003; Li and Fang, 2005). The condition for this optimality is given by Eq. (11).

$$2\sum_{i=1}^{M} \frac{T_1}{T_i} \cong r, r = \left\lfloor \frac{T_1 - M\sigma}{L} \right\rfloor \tag{11}$$

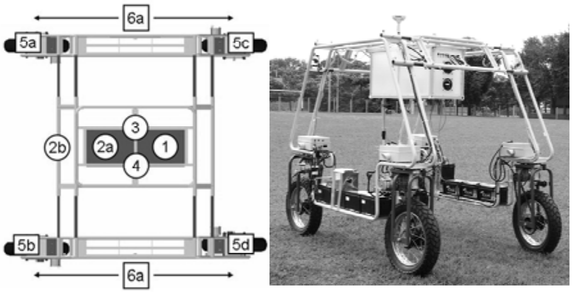
According to Hong (1995), the condition for optimality, in Eq. (11), is the situation in which the maximum utilization of network resource could be achieved by proper choice of T_i (messages sampling times) so that the number of messages to be transmitted do not exceed *r* (maximum number of messages that can be served by the network during the worst-case network traffic). In this optimum condition, the network utilization is increased (and the idleness is reduced) without violating the messages timing requirements or deadlines. On the other hand, if the number of messages to be transmitted exceeds *r,* the messages miss their deadlines and the network becomes overloaded. This situation causes overlapping and loss of messages on the network.

CAN-Based Network Case Study

For a NCS, performance is a function not only of the messages sampling times, but also of the traffic load on the network. In a network, as the sampling times decrease, performance improves until network saturation is reached .The proposal of this case study is to use the CAN mathematical model systemized to calculate the required data for the application of the STS methodology presented. The STS methodology will be used for the optimization of a CAN-based network applied to the control of a mobile robot. The main idea is to ensure the correct choice of all the messages sampling times of the robot devices to achieve the largest possible value of the CAN network utilization and to obtain a good communication performance (minimizing the network idleness) for the CAN-based distributed control system studied.

Characteristics of the Robot Control Network

In this section, network parameters are defined for the CAN-based network used for the distributed control of the mobile robot showed in Fig. 5. The Fig. 5 presents the mechanical structure of the robot and the devices connected by the embedded CAN network. In applications of CAN-based networks, the electronic system that provides the interconnection between a device and the communication network is commonly called by electronic control unit (ECU).



1	Embedded PC
2a	Vision System PC
2b	Monochromatic Camera
3	DGPS (Differential Global Positioning System)
4	Electronic Digital Compass
5a,b,c,d	Set composed by 2 motor controllers (propulsion and guidance), potentiometer sensor and encoder
6a,b	Ultrasonic sensors

Figure 5. Robot Structure and Devices of the CAN-Based Network.

Table 1. Characterization of the Set of Messages of the Robot CAN Network.

Nº	ECU	Message Description	Priority	DL (Bits)	ST (ms)
1	1	control - propulsion motor 1	1	16	50
2		velocity - propulsion motor 1	2	16	50
3		control - guidance motor 1	3	16	100
4		angular displacement - guidance 1	4	16	100
5	2	control - propulsion motor 2	5	16	50
6		velocity - propulsion motor 2	6	16	50
7		control - guidance motor 2	7	16	100
8		angular displacement - guidance 2	8	16	100
9	3	control - propulsion motor 3	9	16	50
10		velocity - propulsion motor 3	10	16	50
11		control - guidance motor 3	11	16	100
12		angular displacement - guidance 3	12	16	100
13	4	control - propulsion motor 4	13	16	50
14		velocity - propulsion motor 4	14	16	50
15		control - guidance motor 4	15	16	100
16		angular displacement - guidance 4	16	16	100
17	5	right side ultrasonic sensors	17	64	500
18		left side ultrasonic sensors	18	64	500
19	6	GPS position	19	64	1000
20	7	digital compass localization	20	32	500
21	8	movement control of the camera	21	48	1000
22	9	control of the embedded PC	22	64	1000
23		management data	23	64	1000
24		monitoring data	24	64	1000
25		network administration	25	64	1000

A set of messages is proposed in Table 1 referred to the devices connected to the CAN bus in the robot of Fig. 5. The column DL represents the data length and the column ST the sampling time of the CAN messages. The column ECU shows the messages transmitted by each electronic control unit connected to the CAN network. For example, the device Nº 3 is responsible for measuring the velocity of the propulsion engine 1 of a encoder and send this information in the CAN network in a message with priority 3, 16 bits of data and sampling time of 100 ms.

Description of the Optimization Process

The data presented in Table 1 is used as data input for the mathematical model systemized for the CAN network. The mathematical model was implemented in a computational program more detailed described in Godoy (2007). This implementation represents one useful task that ease the analysis of the output data obtained with the utilization of the model systemized. With the required data obtained by the use of the simulation software, the application of the STS methodology completes the optimization process for the control network analyzed. The flowchart of Fig. 6 synthesizes this optimization process.

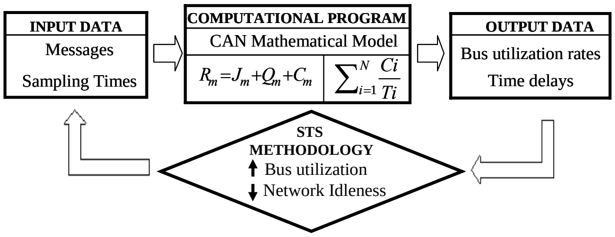


Figure 6. Utilization flowchart for optimization of the CAN-based distributed control system.

Based on the flowchart of Fig. 6, first the values of Table 1 are used as data input for the simulation program (Godoy, 2007). With the application of the computational program, the output data of the bus utilization rates and time delays can be analyzed to verify the optimization possibility for the system. If the value for the CAN bus utilization is too low and the temporal requirement of the messages are satisfied, then the control system can be optimized (the temporal requirement demands that the time delay of the messages be smaller than its message sampling time). In this case, the network presents high level of idleness and the messages sampling times were not chosen correctly. If the bus utilization rate is greater than 100% or the temporal requirement of the messages are not satisfied, then the network will be saturated and messages can be lost or overlapped in the bus. Thus, in accordance with the STS methodology, an increase (or decrease) in the values of these messages sampling times can be achieved for the optimization of the CAN network. With the optimization, the largest possible value for the bus utilization rate is achieved without problems to the system.

Results and Discussions

The following parameters are assumed for the case study proposed: bus speed at 250 Kbits/s, CAN 2.0B message frame, σ = 0.05 ms, L = 0.9 ms and φ equal to the sum of the Rm of the messages of the ECU analyzed. The first results obtained with the input data in Table 1 for the mathematical model program (Godoy, 2007) are shown in Table 2 for the mobile robot control.

Table 2. Messages time delay – Rm and Bus utilization rate – BU for the CAN network.

Message	Rm (ms)	Message	Rm (ms)	Message	Rm (ms)
1	1.1	9	4.86	17	8.54
2	2.06	10	5.26	18	9.18
3	2.46	11	5.66	19	9.82
4	2.86	12	6.06	20	10.3
5	3.26	13	6.46	21	10.86
6	3.66	14	6.86	22	11.5
7	4.06	15	7.26	23	12.14
8	4.46	16	7.66	24	12.78
Bus utilization rate - BU			10.33 %	25	13.18

The results of Table 2 show that the value of the bus utilization rates is too low and the network presents high level of idleness. Slower values selected to the sampling times, before Point 1 in the diagram of Fig. 3, give an unacceptable performance to the robot control system. Thus, as described in the STS methodology, an increase in values for the messages sampling times can be achieved. Now, to demonstrate the utilization of the STS methodology and the optimization of the CAN-based network proposed in the case study, other applications (columns 1, 2, 3 in Table 3) of the computational program are done with the correct input data values. The results in columns 1 and 2 of Table 3 represent two attempts to optimize the control system without the use of a control methodology, manually selecting other sampling times to the ECUs in the CAN bus.

Table 3. Results of the application of the control methodology for the CAN network.

Message	ECU	1		2		3	
		ST (ms)	Rm (ms)	ST (ms)	Rm (ms)	ST (ms)	Rm (ms)
1	1	10	1.66	5	1.66	5	1.66
2		10	2.06	5	2.06	5	2.06
3		25	2.46	5	2.46	5	2.46
4		25	2.86	5	2.86	5	2.86
5	2	10	3.26	10	3.26	10	3.26
6		10	3.66	15	3.66	10	3.66
7		25	4.06	20	4.06	10	4.06
8		25	4.46	25	4.46	10	4.46
9	3	10	4.86	5	* 6.46	15	6.46
10		10	5.26	10	7.26	15	6.86
11		25	5.66	5	* 7.66	15	7.26
12		25	6.06	20	8.46	15	7.66
13	4	10	6.46	10	8.86	20	8.06
14		10	6.86	10	9.26	20	8.46
15		25	7.26	10	* 13.66	20	8.86
16		25	7.66	15	14.46	20	9.26
17	5	300	8.54	300	18.54	300	13.34
18		300	9.18	300	19.18	300	13.98
19	6	100	13.02	100	27.82	100	14.62
20	7	250	13.5	250	28.30	250	18.3
21	8	250	14.06	25	* 29.34	30	18.86
22	9	250	14.7	100	37.58	50	19.5
23		250	15.34	150	38.22	50	26.54
24		250	15.98	150	38.86	50	27.18
25		250	15.98	150	38.86	50	27.18
BU (%)	--	--	47.31	--	84.96	--	74.91

In column 1 of Table 3, the results show that the CAN network presents a low bus utilization rate (<50%) and according to the methodology an increase in the sampling times values is still necessary and recommended. In column 2 of Table 3, note that the bad values selected for the sampling times are those whose some messages miss their deadlines (* in lines 9, 11, 15, 21 of Table 3). These messages will not be transmitted in the CAN network harming the operation of the robot (because, for example, the propulsion and guidance engines 3, which are commanded by messages 9 and 11, will not be controlled). This fact causes overlapping, saturation of the network and loss of messages on the CAN bus. In addition, an unacceptable performance for the control system is achieved with these faster values selected to the sampling times, after Point 2 in the diagram of Fig. 3. To correct this problem, the devices sampling times have to be chosen in agreement with the STS control methodology presented. The correct selection of the sampling times, with values between Points 1 and 2 in the diagram of Fig. 3, give an acceptable performance to the control system. Thus, the final parameters and the results obtained are shown in the column 3 of Table 3. The values for the sampling times of the ECUs 5, 6 and 7 cannot be selected as defined in the STS methodology. This fact can be explained because of the time requirements (related to the hardware) of these devices, that imposes a minimum possible value for each sampling time (for example the minimum sampling time of the compass is 250 ms). The sampling times selected in Table 3 represent these minimum values.

The condition for the optimality presented in Eq. (11) is used to demonstrate that the optimization of the CAN-based control system for the mobile robot was achieved.

$$2\sum_{i=1}^{M}\frac{T_1}{T_i} \cong r \to 4{,}8674 \cong 5$$

The little difference between the results can be explained by the sampling times of the ECUs (5, 6 and 7) that cannot be selected as defined in the STS methodology. The correct definition of the messages sampling times for the CAN-based control system, according to the methodology, are that network delays do not significantly affect the control system performance and the CAN-based system remains stable.

Conclusion

Due to the difficulties encountered with the determination of performance parameters such as transmission times, time delays, message sampling times and network utilization rates, predict the behavior of distributed control networks can be a challenging work. Since the behavior of NCS is determined by these design parameters, assigning the messages sampling times and calculating the network delays are important issues in NCS development. In addition, the network time delays can degrade the control performance and destabilize the NCS. To control this problem, an application of a network methodology is required to diminish the network delay effect and to maintain the performance and stability of these systems.

The detailed timing analysis presented and the mathematical model systemized to calculate performance parameters can provide information about the performance and operational behavior of CAN-based distributed control system and should be useful for designers of NCS. The computational implementation of the CAN mathematical model systemized simplifies the determination of the network parameters and the performance analysis tasks of NCS, generating the required data and allowing the application of the selected sampling time scheduling (STS) methodology. The STS methodology was applied in a case study for the design of an embedded CAN-based network in a mobile robot to obtain an optimized configuration for the operation of the distributed system.

The results show that the optimization of the CAN-based system proposed in the case study can be achieved and the sampling times of the messages were chosen correctly. This condition is that network delays in the distributed communication system do not significantly affect the control system and ensure a better operational performance (network utilization increased and idleness minimized) for the CAN-based distributed control system.

References

Al-Hammouri, A., Branicky, M.S and Liberatore, V., 2008, "Co-simulation Tools for Networked Control Systems", Springer-Verlag, pp. 16-29, Berlin.

Almutairi, N.B., Chow, M.Y. and Tipsuwan, Y., 2001, "Network-based controlled DC motor with fuzzy compensation", Proceedings of the 27th Annual Conference of the IEEE Industrial Electronics Society, Vol. 3, Denver, CO, pp. 1844-1849.

Baillieul, J. and Antsaklis, P.J., 2007, "Control and Communication Challenges in Networked Real Time Systems", Proceedings of IEEE Technology of Networked Control Systems, Vol. 95, No. 1, pp. 09-28.

Bosch, 2006, "CAN 2.0 Specification Version", Available: http://www.can.bosch.com.

Cervin, A., Henriksson, D., Lincoln, B. and Eker, J., Arzen, K., 2003, "How does control timing affect performance? Analysis and simulation of timing using Jitterburg and Truetime", *IEEE Control System Magazine,* Vol. 3, pp. 16-30.

CIA, 2006, "Applications Fields of Controller Area Network", Available: http://www.cia-can.org.

Davis, R.I., Burns, A., Brill, R.J. and Lukkien, J.J., 2006, "Controller Area Network (CAN) Schedulability Analysis: Refuted, Revisited and Revised". Technical Report, University of York. Available in: <www.cs.york.ac.uk/ftpdir/reports/YCS-2006-408.pdf>. Access in: July. 2006.

Farsi, M., Ratcliff, K. and Barbosa, M., 1999, "An overview of controller area network", *Computing & Control Engineering Journal*, Vol. 10, No. 3, pp. 113-120.

Godoy, E.P., 2007, "Desenvolvimento de uma Ferramenta de Análise de Desempenho de Redes CAN para Aplicações em Sistemas Agrícolas", Dissertação de Mestrado, Universidade de São Paulo, São Carlos, 93p.

Goktas, F., 2000, "Distributed control of systems over communication networks", Ph.D. Thesis, University of Pennsylvania, 129 p.

Goodwin, G.C., Quevedo, D.E. and Silva, E.I., 2008, "Architectures and coder design for networked control systems", *Automatica*, Vol. 44, pp. 248-257.

Hespanha, J.P., Naghshtabrizi, P. and Xu, Y., 2007, "A Survey of Recent Results in Networked Control Systems", Proceedings of the IEEE Technology of Networked Control Systems, Vol. 95, No. 1, pp. 138-162.

Hong, S.H. and Kim, W.H., 2000, "Bandwidth allocation scheme in CAN protocol", *IEEE Proceedings of Control Theory and Applications*, Vol. 147, No. 1, pp. 37-44.

Hong, S.H., 1995, "Scheduling algorithm of data sampling times in the integrated communication and control systems", *IEEE Transactions on Control Systems Technology*, Vol.3, pp. 225–230.

Hu, S.S. and Zhu, Q.X., 2003, "Stochastic optimal control and analysis of stability of networked control systems with long delay", *Automatica*, Vol. 39, pp. 1877-1884.

Jeon, J.G., Kim, D.W., Cho, Y.J. and Lee, B.H., 2001, "An analysis of Network-Based Control System Using CAN", Proceedings of the Robotics & Automation Conference, Vol. 4, Korea, pp. 3577-3581.

Lian, F.L., Moyne, J.R. and Tilbury, D.M., 2001, "Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet", *IEEE Control Systems Magazine*, Vol. 21, No. 1, pp. 66-83.

Lian, F.L., Moyne, J.R. and Tilbury, D.M., 2002, "Network Design Consideration for Distributed Control Systems", *IEEE Transactions on Control Systems Technology*, Vol. 10(2), pp. 297-307.

Moyne, J.R. and Tilbury, D.M., 2007, "The Emergence of Industrial Control Networks for Manufacturing Control and Safety Data", Proceedings of IEEE Technology of Networked Control Systems, Vol. 95, No. 1, pp. 29-47.

Nolte, T., 2007, "Reducing pessimism in CAN response time analysis". Mälardalen University, Sweden. Available in: <http://www.mrtc.mdh.se/publications/0384.pdf>. Access in: May, 2009.

Othman, H.F., Aji, Y.R., Fakhreddin, F.T. and Al-Ali, A.R., 2006, "Controller Area Networks: Evolution and Applications", Proceedings of the 2nd International Conference on Information and Communication Technologies – ICTTA, Vol. 2, pp. 3088-3093, 24-28 April.

Punnekkat, S., Hansson, H. and Norstrom, C., 2000, "Response Time Analysis under Errors for CAN", Proceedings of the Real-Time Technology and Applications, Vol. 2, Washington, USA, pp. 258-266.

Tindell, K., Burns, A. and Wellings, A., 1995, "Calculating Controller Area Network (CAN) Message Response Time", *Control Engineering Practice*, Vol. 3, No. 8, pp. 1163-1169.

Tipsuwan, Y. and Chow, M.Y., 2003, "Control Methodologies in Networked Control Systems", *Control Engineering Practice*, Vol. 11, No. 3, pp. 1099-1111.

Yang, T.C., 2006, "Networked control system: a brief survey", IEEE Proceedings of Control Theory and Applications, Vol. 153, No. 4, July, pp. 403-412.