Schedulability Analysis of Messages in a CAN Network Applied to an Unmanned Airship

Max Mauro Dias Santos* , Marcelo Ricardo Stemmer**

Department of Production and Systems Engineering*

Department of Automation and Systems**

Federal University of Santa Catarina, BRAZIL

marcelo@das.ufsc.br, maxmauro@unilestemg.br

Francisco Vasques

Department of Mechanical Engineering

University of Porto

PORTUGAL

vasques@fe.up.pt

Abstract - This paper presents the application of a CAN network to the integration of sensors and actuators in an unmanned airship called HELIX, based on the architecture of distributed processing proposed by [4]. Initially, a study of the viability of several communication networks used in vehicles (such as CAN, VAN, C2D and FIP, etc.) was accomplished [6]. The preliminary study concluded that the behavior of CAN was better for this application when compared to other types of networks analyzed. In the present work, a detailed study of the employability of a CAN network in the HELIX project was accomplished, including situations not considered in the preliminary analysis, such as the presence of Jitter in the group of messages, the high load of messages over the network, and the consideration of presence of transmission errors. As the CAN networks offers good properties for transmission of messages with temporal requirements, a schedulability analysis was accomplished for the transmission of messages and a numeric example is presented in an illustrative way to the present problem.

I. INTRODUCTION

The control of vehicular systems such as automobiles, airships, space ships and trains, is generally accomplished by a set of sensors, actuators and processing units. Initially the control of such vehicles used to be conceived as a centralized system, in agreement with the existent technological limitations. With the evolution of the microprocessors and of the network communication systems, the philosophy of distributed processing took a prominence place in function of the processing capability, safety and reliability.

The modern embedded systems in vehicles are distributed and the processing is accomplished by a set of sensors and actuators that communicate through a network. The distributed and networked control systems have three important characteristics when compared to centralized ones: a) volume reduction; b) weight reduction and c) increased processing capability.

In the present work we will discuss the networked control of an unmanned airship, the helicopter HELIX [6],

conceived and projected by the Gyron Company [7]. This application has hard real time communication constraints.

The HELIX control system was initially projected with a centralized processing, and in [6] a distributed processing proposal was presented, in function of the advantages that this architecture possesses [1]. In [6], a viability study of several networks existent in the market, such as VAN, CAN and FIP, among others, was accomplished. CAN [8] provided a better performance. CAN (Controller Area Network) was originally projected by BOSCH, with the objective of satisfying the requirements of electronic systems in automotive control systems.

In this work, a detailed study of the use of CAN in the HELIX airship, considering conditions of Jitter on the messages, overload and noises on the net (aspects that were not considered in previous works), was accomplished.

The following section describes the HELIX project. Section 3 presents a distributed architecture model for HELIX. Section 4 describes the main characteristics of CAN, as well as its temporal properties under conditions of normal operation and under errors. Section 5 shows the application of CAN to the HELIX system, and some numeric results are presented. Section 6 describes a general conclusion of the accomplished study.

II. THE UNMANNED HELICOPTER HELIX

In 1991, after the accomplishment of deepened studies and market estimates on RPVs (Remotely Piloted Vehicles), GYRON began the development of an aerial robot, which received the denomination of Project HELIX. The HELIX project consisted of an unmanned helicopter, a control station on earth and a support vehicle with capacity to transport up to 2 airships.

The helicopter was conceived to accomplish tasks in that the human being took life risk, such as, inspection of transmission lines, surveillance of fire in forests, analysis of metallic structures, among others.

* lecturer at UNILESTE, PhD student at UFSC with a scholarship from FGPA.

Figure 1: Communication between the helicopter and the control station on earth

It has a capacity to drive useful loads of up to 5 kg. The basic model transports a video camera for the transmission of images in real time, and a communication system among the control station on earth and the airship was accomplished through a radio frequency system, as can be seen in Figure 1.

The original control system of HELIX consisted of a point-to-point architecture for the interconnection of sensors, actuators and control modules, with the objectives of monitoring and control of the airship.

The necessary intra-vehicle data in the system HELIX is classified in three different types, in agreement with its functionality: a) data for the control of the airship; b) data for monitoring objectives; and c) data related to the application, including to the embarked camera and eventual additional sensors.

Figure 2: Original processing structure of HELIX

The initial point-to-point architecture was composed of 2 RISC microprocessors and a transputer [6] with a totally centralized processing, as can be seen in Figure 2.

III. THE DISTRIBUTED REAL TIME ARCHITECTURE

In [6] a distributed computation model for the HELIX system was proposed, using a diffusion network and supporting bi-directional communication. The proposed distributed architecture is shown in the Figure 3, with the respective details of the tasks and messages.

Figure 3: A distributed architecture for the HELIX system

In figure 3 the messages exchanged among the nodes are represented by f_i , with i identifying the message. A detailed description of each message being transmitted is presented ahead.

The study described in [6] was accomplished considering different network technologies used in automotive systems and industrial automation, such as: a) CAN; b) FIP; c) VAN; d) J1850; e) MIL-STD-1553B; f) C2D and g) CSMA/DCR.

The above-mentioned study concluded that CAN was the best solution for the application in focus. In the present work, the usage of CAN is assumed as a project decision and the temporal behavior of the network is analyzed in more detail.

IV. TEMPORAL ANALYSIS OF CAN

An analysis of the latency of messages is presented in the worst case for messages in real time, which is similar to the analysis made in [2]. Some modifications have been considered: the deadline of a message (D_m) should not be larger than the period (T_m) and the bus controller should not transmit a message of low priority if exists a high priority message pending.

CAN adopts a MAC (Medium Access Control) protocol known as CSMA/NBA (Carrier Sense Multiple Access with Non-destructive Bit-wise Arbitration). The Non-destructive bit-wise arbitration allows 100% utilization and message priority based on an 11-bit packet identifier. In this protocol, any node can access the bus when it is silent. Similar to Ethernet, each node attempts to transmit when the bus is free. Unlike Ethernet, there is no collision. If two or more nodes start transmitting simultaneously, bus conflict is resolved by bit-wise arbitration using the IDENTIFIER field. A “0” is dominant on wire and overrides a “1”. When a node transmits a “1”, but hears a “0”, it immediately stops transmitting. The “winning” node continues to transmit its message to completion. This mechanism guarantees that neither information nor time is lost. The value of the IDENTIFIER defines priority during arbitration (lowest IDENTIFIER “wins” arbitration). This means two nodes cannot share the same IDENTIFIER. All nodes check the consistency of the message being received and will acknowledge a consistent message and flag an inconsistent message in the ACK slot. CAN provides automatic error detection, signaling, and retries. The data portion of packet can be 0 to 8 bytes long.

The CAN network operates with a fixed priority scheduling policy, but does not support pre-emption of messages.

A. Analysis of CAN response time

According to [1], the transmission time of a message m in the worst case is given by:

$$R_m = J_m + w_m + C_m \quad [1]$$

The term C_m represents the biggest time required to transmit a message m over the physical bus. The CAN frame contains 47 bits of overhead and a bit-stuffing width of 5 bits. Only 34 of the 47 bits suffer stuffing. The equation that represents C_m is:

$$C_m = \left(\left\lfloor \frac{34 + 8s_m}{5} \right\rfloor + 47 + 8s_m \right) \tau_{bit} \quad [2]$$

The term S_m indicates the size of the message m in bytes and the term τ_{bit} is the time spent to transmit a bit over the bus (for the 1Mbps bus we would have $1\mu s$).

The term w_m represents the delay in the queue in the worst case (the biggest time spent between putting a message in a priority ordered queue and the beginning of the transmission) and is given by:

$$w_m = B_m + \sum_{\forall j \in hp(m)} \left\lceil \frac{w_m + J_j + \tau_{bit}}{T_j} \right\rceil C_j \quad [3]$$

The term B_m is the blocking time in the worst case of the message m and is given by:

$$B_m = \max_{\forall k \in lp(m)} (C_k) \quad [4]$$

where $lp(m)$ is the set of messages in the system with lower priority than message m .

The term J_j is the Jitter and T_j is the period of the set of messages with higher priority than message m . Since the term w_m appears in both sides of the equation, a recurrent solutions is required, as shown bellow:

$$w_m^{n+1} = B_m + \sum_{\forall j \in hp(m)} \left\lceil \frac{w_m^n + J_j + \tau_{bit}}{T_j} \right\rceil C_j \quad [5]$$

Finally, the term J_m represents the message Jitter (average waiting time of the message at the transmission queue before scheduling) and is empirically determined. In the present work we use the value 0.1 ms.

B. Analysis of CAN response time under error conditions

Ken Tindell and Allan Burns [1] proposed a model for the calculation of the worst-case response time of messages in CAN under error conditions. The response time can be calculated as:

$$R_m = J_m + q_m + C_m \quad [6]$$

The terms J_m and C_m have the same meanings as in the previous section. The term q_m corresponds to the time spent by a message in the waiting queue under error conditions

and is given by the following recurrent equation:

$$q_m^{n+1} = B_m + \sum_{\forall j \in hp(m)} \left\lceil \frac{q_m^n + J_j + \tau_{bit}}{T_j} \right\rceil C_j + E_m(q_m + C_m) \quad [7]$$

$E_m(t)$ is a function of the error recovery overhead and gives the superior limit of all error recovery overheads that can occur in a time interval t . It is given by the following equation:

$$E_m(t) = \left(n_{error} + \left\lceil \frac{t}{T_{error}} \right\rceil - 1 \right) \left(29 + \max_{\forall k \in hp(m) \cup \{m\}} (C_k) \right) \quad [8]$$

where n_{error} is the number of errors that can occur in a small arbitrary interval and T_{error} is the period of error occurrence. The number of errors in an interval of duration t is given by:

$$n_{error} = \left\lceil \frac{t}{T_{error}} \right\rceil - 1 \quad [9]$$

In every error, the error recovery overhead can be increased in 29 bits, followed by a 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:

$$\max_{\forall k \in hp(m) \cup \{m\}} (C_k) \quad [10]$$

In the following sections, the equations above will be used in a schedulability analysis of CAN applied to the unmanned helicopter HELIX, considering the conditions of messages with Jitter, presence of error and a very elevated message load in the net.

V. APPLICATION OF CAN TO THE HELIX AIRSHIP

A. Model of Tasks and Messages

The HELIX control structure is decomposed in 8 different subsystems: Controller, Inertial, Position, Actuators, RF-Links, Attitude, Monitoring and Camera. Each one includes a group of real time message generating tasks.

The main temporal properties of the messages generated in the HELIX system are presented in Table 1, where T is the period, D is the deadline and P/S stands for Periodic/Sporadic.

In relation to Jitter, we will consider that it is totally dependent of the latency of the application. The latency is the maximum elapsed time between the moment an application task asks for the transmission of a message and the placement of the message in the transmission buffer. We can consider a Jitter of 1ms for any flow of messages, because $1ms \ll D$ as presented in Table 1.

Another important characteristic of this application is the length of the signal #13. Signal #13 is the image obtained by the camera and is transmitted from T1camera to Rcontrol every 1000 ms. Originally, signal #13 had a length of 1064 bits. As it is very long, it must be divided in many smaller messages of the maximum size acceptable for CAN frames. The CAN frame is capable to transport up to 8 bytes of data. This way the messages with a size superior to 8 bytes should be broken into fragments and mounted in frames with the same temporal properties inherited by the original message.

An alternative for the minimization of the load of messages over the net is to insert a data compression algorithm in the node T1Camera and a decompression algorithm in the node Rcontrol. That was already accomplished as shown in Table 1 and resulted in a gain of 40% in size when the image is in JPG or FIG format.

#	Signal Description	Size [bits]	Length [# of msg]	T [ms]	D	J [ms]	From	To
1	PWM signals for servos s0 to s7	8 x 16	2	25	25	1	T1Control	Ractuator
2	Angular acceleration	3 x 16	1	50	50	1	T2Inercial	Rcontrol
3	Angular speed	2 x 32	1	50	50	1	T1Inercial	Rcontrol
4	Position references Xref, Yref, Zref and Href	4 x 48	3	50	50	1	T1RFLinks	Rcontrol
5	Swashplate motor rotation	1 x 16	1	100	100	1	T1Monit	Rcontrol
6	□ (yaw) angle and its derivate	2 x 32	1	100	100	1	T1Atitude	Rcontrol
7	Operation codes	3 x 48	3	200	200	1	T2RFLinks	Rcontrol
8	Linear speed (x, y and z)	3 x 48	3	1000	1000	1	T2Posição	Rcontrol
9	VDC, motor temperature	3 x 16	1	1000	1000	1	T2Monit	Rcontrol
10	Latitude, longitude and altitude (x, y, z)	3 x 48	3	1000	1000	1	T1Posição	Rcontrol
11	Adjustment Reference	4 x 48	3	1000	1000	1	T3RFLinks	Rcontrol
12	Telemetry e monitoring	24 x 48	18	1000	1000	1	T2Control	RFLinks
13	Digitalized Signal from camera	640	10	1000	1000	1	T1Camera	Rcontrol

Table 1: Description of the main communication signals

B. Schedulability Analysis of Messages

Table 2 shows the results of the scheduling test over the group of messages presented in Table 1 (where each message stream is sub-divided according to the related number of messages), considering different transmission rates at the net. Two cases will be analysed: a) supposing

that there are no transmission errors; and b) considering that the system under analysis is subject to the following error conditions: $n_{error}=5$ and $T_{error}=10ms$.

In Table 2, the grey cells represent, for a given network data rate, message streams that do not respect the deadline restriction, *i.e.*, that can have messages with a response time longer than the related deadline.

#	Message Streams' Properties				Response Time (CAN network under normal operation)				Response Time (CAN network subject to error conditions)			
	C [bits]	J [ms]	T [ms]	D [ms]	w 125Kbit/s	w 250Kbit/s	w 500Kbit/s	w 1Mbit/s	Q 125Kbit/s	q 250Kbit/s	q 500Kbit/s	q 1Mbit/s
1.1	64	1	25	25	6.7	3.3	1.7	0.8	*	23.1	8.7	4.4
1.2	64	1	25	25	12.1	6.1	3.0	1.5	*	*	11.5	5.0
2	48	1	50	50	16.4	8.2	4.1	2.1	*	36.2	12.6	5.6
3	64	1	50	50	21.9	11	5.5	2.7	*	39.0	13.9	6.3
4.1	64	1	50	50	27.4	13.7	6.8	3.4	*	44.6	15.3	6.9
4.2	64	1	50	50	43.9	16.4	8.2	4.1	*	47.3	16.7	7.6
4.3	64	1	50	50	49.3	19.2	9.6	4.8	*	*	18.1	8.3
5	16	1	100	100	*	22.8	11.4	5.7	*	89.0	19.9	9.2
6	64	1	100	100	*	21.9	10.9	5.5	*	88.2	19.4	9.0
7.1	64	1	200	200	*	25.6	12.8	6.4	*	94.6	22.7	9.9
7.2	64	1	200	200	*	33.8	14.2	7.1	*	97.4	24.0	11.3
7.3	16	1	200	200	*	34.7	14.6	7.3	*	98.3	27.2	11.5
8.1	64	1	1000	1000	*	40.2	17.3	8.7	*	145.6	30.0	12.9
8.2	64	1	1000	1000	*	42.9	18.7	9.4	*	148.3	32.7	13.6
8.3	16	1	1000	1000	*	43.8	19.2	9.6	*	149.2	33.2	13.8
9	48	1	1000	1000	*	45.9	20.2	10.1	*	186.7	34.3	14.3
10.1	64	1	1000	1000	*	48.7	21.6	10.8	*	189.4	35.6	15.0
10.2	64	1	1000	1000	*	51.4	23.0	11.5	*	195.0	37.0	15.7
10.3	16	1	1000	1000	*	70.9	23.4	1.7	*	195.9	37.4	15.9
11.1	64	1	1000	1000	*	73.7	24.8	12.4	*	198.7	38.8	16.6
11.2	64	1	1000	1000	*	76.4	28.9	13.1	*	249.6	41.6	17.3
11.3	64	1	1000	1000	*	84.7	30.3	13.8	*	287.7	43.0	18.0
12.1	64	1	1000	1000	*	87.4	31.7	14.5	*	293.3	44.3	18.7
12.2	64	1	1000	1000	*	90.1	33.0	15.1	*	296.0	45.7	19.4
12.3	64	1	1000	1000	*	92.9	34.4	15.8	*	298.8	47.1	20.8
12.4	64	1	1000	1000	*	95.6	35.8	16.5	*	343.3	48.5	21.4
12.5	64	1	1000	1000	*	98.4	37.2	17.2	*	346.1	49.8	22.1
12.6	64	1	1000	1000	*	101.1	38.5	17.9	*	348.8	63.3	22.8
12.7	64	1	1000	1000	*	126.1	39.9	18.6	*	386.9	64.7	23.5
12.8	64	1	1000	1000	*	37.4	16.0	8.0	*	142.8	28.6	12.2
12.9	64	1	1000	1000	*	134.3	41.3	19.3	*	389.7	66.1	24.2
12.10	64	1	1000	1000	*	137.1	42.6	19.9	*	395.2	67.4	26.2
12.11	64	1	1000	1000	*	139.8	44.0	20.6	*	398.0	68.8	26.9
12.12	64	1	1000	1000	*	142.6	45.4	21.3	*	448.9	71.6	27.6
12.13	64	1	1000	1000	*	145.3	46.8	22.0	*	487.0	73.0	28.3
12.14	64	1	1000	1000	*	148.1	48.1	22.7	*	489.8	74.3	29.0
12.15	64	1	1000	1000	*	150.8	49.5	23.4	*	495.4	78.5	29.7
12.16	64	1	1000	1000	*	172.1	60.2	24.1	*	498.1	79.8	31.1
12.17	64	1	1000	1000	*	174.9	61.5	26.1	*	539.8	82.6	31.8
12.18	64	1	1000	1000	*	183.1	62.9	26.8	*	545.4	84.0	32.4
13.1	64	1	1000	1000	*	185.9	64.3	27.5	*	548.2	85.4	33.1
13.2	64	1	1000	1000	*	188.6	65.6	28.2	*	586.2	86.7	33.8
13.3	64	1	1000	1000	*	191.3	67.0	28.7	*	589.0	88.1	34.5
13.4	64	1	1000	1000	*	194.1	68.4	29.6	*	594.6	89.5	35.2
13.5	64	1	1000	1000	*	196.8	69.8	30.2	*	597.3	92.2	35.9
13.6	64	1	1000	1000	*	199.6	71.1	30.9	*	648.3	93.6	36.6
13.7	64	1	1000	1000	*	236.4	72.5	31.6	*	686.4	95.0	37.2
13.8	64	1	1000	1000	*	239.2	73.9	32.3	*	689.1	96.4	37.9
13.9	64	1	1000	1000	*	241.9	75.3	33.0	*	694.7	97.7	38.6
13.10	64	1	1000	1000	*	244.7	79.4	33.7	*	697.4	99.1	39.3

Table 3: Response time analysis of messages in the CAN network

The results show that the net running at the speed of 125 Kbps cannot tolerate the assumed rate of errors, with all the messages not satisfying their respective deadlines. When CAN runs at the speed of 250 Kbps, only the message 1.2 doesn't satisfy its deadline, resulting that the whole group of messages is considered unschedulable. For the speeds of 500 Kbps and 1 Mbps the system can tolerate the presumed error rates, since the group of messages is totally schedulable, with a certain rest.

VI. CONCLUSION

A functional description of the HELIX embarked system was accomplished, consisting of a distributed processing model with a net of the type CAN. For the analyzed group of messages, mathematical models were presented for verification of the warranty of temporary requirements.

The main contribution of this work consists of a detailed analysis of message schedulability for a real process, considering real conditions such as error rate, Jitter, and response time for each generated message.

REFERENCES

- [1] Tindell, K. and Burns, A.: "Guaranteed Message Latencies for Distributed Safety Critical Hard Real-Time Networks", Technical Report YCS 229, Department of Computer Science, University of York, England, May 1994.
- [2] Tindell, K., Burns, A. and Wellings, A.: "Calculating Controller Area Network (CAN) Message Response Time". In *Control Engineering Practice*, Vol. 3, N° 8, pp. 1163-1169, 1995.
- [3] Tindell, K.; Burns, A.: "Analysing Real-Time Communications: Controller Area Network (CAN)". *Proceedings 15th IEEE Real-Time Systems Symposium*, pp. 259-265, December 1994.
- [4] Pinho, L. M.: "A Framework for the Transparent Replication of Real-Time Application", PhD Thesis, FEUP – Univ. Porto, 2001.
- [5] Vasques, F. and Pinho, L. M.: "Atomic Multicast Protocols for Reliable CAN Communication", *Simpósio Brasileiro de Redes de Computadores - SBRC'2001*, Florianópolis – Brasil. pp. 194-209, Maio 2001.
- [6] Fascioni, L. C.: "Avaliação de Arquiteturas de Comunicação Distribuídas para Interligação de Sensores/Atuadores em um Helicóptero não Tripulado". *Dissertação de Mestrado*, Setembro, 1996.
- [7] Gyron. Projeto HELIX: Definição e descrição geral. Documento Interno, 1992.
- [8] Gyron. HELIX Software Reference Manual. Documento Interno, Abril 1996.
- [9] Lawrenz, W.: *CAN System Engineering: From Theory to Practical Applications*. New York: Springer-Verlag, 1997.