Experimental Analysis of Outage Times for PROFIBUS Networks

José Augusto Carvalho
Escola Superior de Tecnologia e de Gestão
Campus de Santa Apolónia
5301-857 Bragança
Portugal
jac@ipb.pt

Adriano Silva Carvalho, Paulo José Portugal
DEEC, FEUP
Rua Dr. Roberto Frias
4200-465 Porto
Portugal
{asc, pportugal}@fe.up.pt

Abstract- Distributed control systems operation strongly depends on the communication system performance. Therefore, their analysis becomes an important aspect to be considered, particularly in situations where communications have real-time constrains and the network operates on faulty environments.
In this paper the behavior of PROFIBUS network in the presence of faults is analyzed. This analysis is performed in an experimental base supported by a real network. The network operation is disturbed through a set of fault injection experiments from which performance metrics related with PROFIBUS outage times are evaluated.

I. INTRODUCTION

Fiedbuses are nowadays a widely used communication system, oriented to automation and industrial applications. Their use allowed the replacement of the traditional centralized control architectures, based on analogue signals or proprietary communication protocols, and at same time the support of advanced functionalities [1].

Industrial environments are characterized by the existence of a high diversity of equipments that are source of large patterns of electromagnetic interference (EMI), which can induce faults in electronics circuits [2]. In the communication systems, these types of faults produce errors in the communication bus by corrupting data contents. To recover from these situations fieldbus networks implement several fault-tolerant mechanisms. However, this creates an overhead which introduces delays in the delivered messages and performance degradation in the control system operation [3,4].

In control systems with real-time requirements, message delays can disturb the system behavior by leading to their failure [3,4]. In this context, an evaluation of fieldbus networks must be performed, enabling the identification of the most important parameters from a performance viewpoint.

In this paper a performance evaluation of the PROFIBUS network is performed [5]. The behavior of the *Physical*, *Fieldbus Data Link* (FDL) in the presence of transient faults is analyzed in order to obtain the PROFIBUS temporal response in such conditions.

The paper is structured as follows. In section II is given a description of the PROFIBUS network. A brief discussion of the related work on performance degradation in PROFIBUS networks is given in section III. In section IV the PROFIBUS fault behavior and associated recovery mechanism is presented. The methodology used for the implementation of the experiments is presented in section V. The results obtained from the experiments are discussed in section VI. Finally the conclusions are presented in section VII.

II. PROFIBUS

The PROFIBUS [5] is a fieldbus designed for use at the low level of factory automation systems, where it performs high-speed data exchange between process controllers and field devices, such as sensors and actuators.

Two types of stations can be connected to the network: **(i)** Masters, usually performing automation and control tasks (e.g. PLCs). Their operation typically consists of polling a set of associated slave devices and executing control programs; **(ii)** Slaves, consisting on peripheral devices (e.g. I/O) exchanging data with the masters. Masters are referred as *active* stations, and slaves as *passive* ones. Communications can only be initiated by the active stations. Passive stations can only access to the medium in response to an active station request. The communication stack is organized according the OSI model, but using only with 3 layers: *Physical*, *Fieldbus Data Link* (FDL) and *Application*. The stations are interconnected according a bus topology.

The medium access control is achieved by a hybrid access medium method: a decentralized method accordingly to the principle of token passing is underlain by a central method according to the master-slave principle. In order to manage the bus access, active stations have to build and manage a logical ring. Each active station has its own *List of Active Stations* (LAS), which represents all active members of the ring. According to the LAS, the token is passed on the ring from active to active station on ascending station address way, except if the token holder is the station with the higher address value. In this case, the token is passed to the station with lowest address value.

The station that possesses the token is referred as *this station* (TS). After receiving the token from the station immediately below in LAS, referred as *previous station* (PS), it is able to initiate the message cycles with the slaves. The cycle duration is defined by the *token holding time* (T_{TH}) and by the number of messages to transmit. After a master has received the token, the measurement of the token rotation time begins. The time measurement ends at the next token receipt and results in the *real token rotation time* (T_{RR}). At the same instant a new measurement of the following rotation time starts. The T_{TH} is defined as the difference between the target rotation time (T_{TR}), which is a configuration parameter, and T_{RR}. When the message cycles finishes, it has to pass the token to the immediate station on the LAS referred as *next station* (NS). If the token holder is the only active station on the ring it has to pass the token to itself. Each active station has two main types of tasks: an active phase where the station communicates with the associated slaves and a manage-

ment phase, where the station performs maintenance ring operations. On the management phase the station has to keep its LAS updated in response either to a station insertion or removal from the ring.

III. RELATED WORK

Fieldbus networks are typically used to support distributed real-time control applications. Therefore, in order to fulfill real-time system requirements, the communications protocols have to present small and predictable message latency with a minimal jitter [1,3,4]. Although the referred requirements are concerns of fieldbuses protocols, these are influenced by faults conditions such as the ones that lead to the corruption of transmitted data.

In the case of PROFIBUS, errors in transmitted data lead to outage events. These outages are interruptions of the communications services which can affect either the entire ring or single stations. They correspond to token losses and station removals from the logical ring, respectively.

Outage events have a strong impact in message latency times, and consequently in their Worst Case Response Time (WCRT). However, usual PROFIBUS behavior and schedulability analyses are performed based on the performance characteristics of the protocol [6,7,8]. Therefore, these analyses are only correct if no faults occur during the network operation.

With PROFIBUS, a few analyses were performed to verify how it behaves in fault scenarios. In [9,10,11] the ring stability of PROFIBUS in error-prone links is analyzed. This work focuses at transient faults and proposes either a simulation model [10], based on a proprietary development environment, and an analytical model [11] (as an approximation to [10]), to evaluate the network behavior. Although the model presented in [9,10] had identified several outages events, their causes are not completely analyzed. Besides, in some cases the explanations of verified faults are contradicting with the observed networks operation.

In [12] the authors present an analysis of PROFIBUS network behavior. The analysis is supported by a real network that has its operation disturbed by a set of fault injection experiments. From these experiments different outage events are identified. According to the effects of these ones in the network operation events are classified as: *System Outage*, when a token loss occurs and all stations are inhibited to perform their communication services; *Station Outage*, when one or more stations are removed from the logical ring. The impact of these events in the network operation is quantified from a probabilistic basis.

In this paper the previous work is extended by evaluating the expected duration of these events and also the bus cycle time.

IV. PROFIBUS FAULT BEHAVIOR

This section performs a description of PROFIBUS networks on fault conditions and the corresponding recovery mechanisms.

The PROFIBUS protocol makes use of the following services in order to manage the network operation:

- **Message cycle**, which is used to exchange data between stations (active and slaves ones). It is performed by using *Action / Replay* frames.
- **Logical ring management**, which is used to establish the logical ring, allows inserting new stations and performs the rotation of the token among active stations. It is accomplished through using the *Request FDL Status action* frames and *Token* frames.

From data integrity viewpoint Action / Replay frames are protected by means of a FCS (*Frame Check Sequence*) field (1 byte), which enables the stations to detect frame errors. The token frame is composed by 3 UART characters and its integrity is assured only by the character's parity bits (Fig. 1). Therefore it is possible that an error in Source Address (SA) or Destination Address (DA) fields occurs without being detected -undetected errors- (e.g. 2 bit errors).

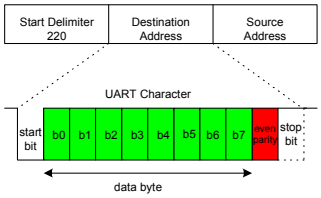

Fig. 1 - Token frame and UART character formats.

Due to the importance of the token frames in the management tasks, errors in the token frame can disturb the logical ring and affect the communication tasks. This kind of instability causes a disruption of the service (outages) which hampers temporally the network (*System Outage*) or the station (*Station Outage*).

A. System Outage

A *System Outage* event is caused by a token loss. When a token loss occurs all stations are inhibit to execute their message cycles until the generation of a new token.

The token loss has its origins on errors during token pass procedures. The token pass procedure takes place whenever the master has accomplished their message cycles. The token is passed to its successor (NS) by transmitting a token frame. If after transmitting the token frame and within a *slot time* (T_{SL}) the token transmitter receives a valid frame it assumes that the NS owns the token. When this procedure is disturbed any of the following events can occur [12]:

- ***Fatal Token Error***. This error occurs if during a token transmission a station does not receive its own token frame. In this case there is a fatal error in the transceiver. The station stops its activity in the logical ring, enters in the *Offline* state and the token is lost. In this scenario the LAS contents are also lost. Although this behavior results apparently from a permanent fault, transient faults can also produce the same results [12];

- ***Normal Token Error***. This error occurs when two consecutive token frames contains errors (not fatal ones). The station leaves the ring and enters in the *Listen token* state. In this scenario the token is lost and LAS contents are maintained;
- ***Slot Time Error***. This error occurs if after a token transmission and within the *slot time* an error pattern occurs during the remaining bus *idle time*. In this case the token owner assumes that another master is active on the bus and enters the *Active Idle* state. This is due to the fact that these errors generate UART characters which are interpreted as a frame header. Whether the token is not successfully passed then this causes a token loss.

B. Station Outage

A *Station Outage* event occurs when a station is removed from the logical ring. Outside the ring, the station cannot perform their communication functions which lead to their outage.

In [12] *Station Outage* events are observed as consequence of 3 scenarios in which token frames are affected by errors:

- ***TS Address Error***. This event happens as result of undetected errors. When a station does not possess the token and it listens two consecutive tokens with Source Address (SA) equal to the station address, the station leaves the logical ring and enters in *Listen Token* state;
- ***LAS Inconsistency***. In order to keep the LAS updated, active stations constantly have to monitor token frames in the bus. However, some undetected token frames errors can lead to a *LAS Inconsistency*. In this case a station verifies that its LAS contents are inconsistent with the real logical ring organization. When this error occurs the station leaves the logical ring until re-establishing its LAS. This error was observed when a token frame is corrupted and the resulted frame violates the insertion and removal station rules [12]. As consequence, the logical ring partially collapses, and in most situations the only stations which are not affected are the ones involved in the token pass procedure;
- ***Station Jump over***. This event has its source in the mechanisms that allow at removing stations from the logical ring. In the case of an absence of reaction from the NS, the station that owns the token retries to pass the token for two times. When this value is reached and if the token is not passed, the station tries to pass it to the next station in LAS. This process is repeated until the token pass procedure succeeds. The stations which are skipped in the token pass process are then removed from the LAS. This behavior is due to: **(i)** Undetected token errors (SA or DA errors) by other stations; **(ii)** Undetected errors by the token owner loop back mechanism but which are detected by others stations. The loop back mechanism consists of the activation of both transmitter and receiver channels of the transceiver. This mechanism is used by the token owner at the token transmission with two purposes: To detect errors in the token transmission and to verify the state of its transceiver. Thus when the loop back mechanism is activated several physical conditions can lead to the occurrence of undetected errors at token transmitter station. One of these physical conditions is related with electrical signal strength that creates the errors. The signal strength is influenced by the bus electrical parameters, such as capacitance resistance and inductance. Thus the attenuation can reduce the signal strength at loop back location by a magnitude that cannot be recognized by the token sender but which is detected by other station in different bus locations.

C. Recovery Mechanisms

PROFIBUS recovers from *System Outage* and *Station Outage* events by the following two mechanisms respectively:

- **Timeout**. Timeout is implemented by a timer which monitors the stations bus activity and idle time. This timer is started either after *Power On* (in *Listen Token* state) or after receiving the last bit of a frame. It ends after receiving the first bit of a frame [5]. If the idle time reaches timeout, the bus is regarded as inactive and then a new token is generated at the station where the timeout was triggered. The timeout is defined as follows:

$$T_{TO} = 6 \cdot T_{SL} + 2 \cdot n \cdot T_{SL} \quad (1)$$

where n is the station address and T_{SL} is the *Slot Time*. Thus by using this equation and assuming n the lower station address in the logical ring, the time to recovery from a *System Outage* event can be computed.

This mode of operation causes misbehaviors in some circumstances. Indeed, in noisy environments it has to be expected multiple restarts of the timer, which increase the recovery time. This is due to the faults effect in the bus idle signal. When the bus is idle an error produce a transition from logical level 1 to 0 that generates a start bit and the associated UART character, which restarts the timer;

- **Station Insertion**. Each station in the logical ring is responsible for the insertion of new station, the address of which are situated in the range from TS to NS. This address range is called *GAP* and is represented by the *GAP List* [5]. Thus each station in the logical ring examines its address range periodically in the interval given by the *GAP Update Time* (T_{GUD}) for changes concerning masters and slave stations. This is accomplished when there is still Token Holding Time available, by examining one address per token receipt, using the Request FDL Status frame. The T_{GUD} is multiple of the T_{TR} and is defined as follow:

$$T_{GUD} = G \cdot T_{TR} \quad ; 1 \le G \le 100 \quad (2)$$

Thus the time to add a station to the logical ring depends on the following parameters: the GAP range, T_{GUD} and T_{RR}. Fault occurrence cause an increase of the insertion time due to any of the following reasons: **(i)** Message retry, which increases the T_{RR} – in the limit T_{TH} is not

enough to perform the Request FDL Status. **(ii)** Occurrence of station removals that lead to the increase of GAP range.

V. OUTAGE TIMES ANALYSIS

This section discusses a methodology, based on the PROFIBUS fault behavior, to measure the *System* and *Station Outage* times.

According to the PROFIBUS characteristics it is assumed the importance to consider a method to carry out experimental results. In fact, some EMI faults can interfere in the PROFIBUS behavior that imposes an experimental evaluation to analyze its performance in such fault conditions. Parameters such as *System Outage Time* and *Station Outage Time* are relevant to establish PROFIBUS performance.

Two analysis need to be performed in order to get complete knowledge on the operation of PROFIBUS. The first one, is related to get knowledge on the occurrence, and the related probability, of the events causing them. This analysis was already performed by the authors in a previous work [12]. The second one, adds a step on the knowledge about *System Outage Time* and *Station Outage Time* by estimating their expected (mean) values, which enables to characterize the temporal behavior of PROFIBUS.

Therefore, an experimental analysis is developed within a fault injection framework. This one is supported by a hardware platform built around DSTni-LX-002 microcontroller [14]. Basically they are needed two infrastructures [12, 13]:

- The first one to implement a PROFIBUS network able of being handled at *Fieldbus Data Link* Layer (FDL).
- The second one implements an appropriate faults injection framework. It comprehends two components: an injector to manipulate the bus state and a monitor to catch up the relevant bus activity (e. g. fault activation and related effects).

A. Experimental Setup

In order to perform the experiments the hardware platform was configured with 9 communication nodes. All this nodes are configured as masters and its activity is only related with the maintenance of the logical ring (i.e. they only generates Token and FDL Request Status frames). Their addresses are fixed and were generated according to uniform distribution, such as: Sad={9, 20, 25, 32, 35, 38, 51, 69, 83}. All parameters related with the FDL layer (Tab. 1) are maintained constant between experiments.

Table 1. - FDL parameters

Parameter	Value
Bit rate	500kbit/s => tbit=2µs
T_{TR} - Target rotation time	20ms –10000 tbit
T_{ID1} - Idle Time 1	37 tbit
T_{ID2} - Idle Time 2	100 tbit
T_{SL} – Slot Time	200 tbit -
T_{RDY} – Ready Time	11 tbit
HAS	126
T_{GUD}	60000 tbit (G=6)

Different scenarios were analyzed by changing the following parameters:

- **Bit error rate (BER).** To verify error sensitivity, faults are injected into the communication bus according to a geometric distribution with different error rates. This distribution is chosen because faults are injected at discrete time instants (*tbit* multiples) and because it has no memory [15];
- **Error length (BEL).** In order to verify the effects of fault length in the behavior of fault-tolerant mechanisms, and in particularly the effects of undetected token errors, a set of different error lengths were used: 1, 2 and 4 bits.

VI. EXPERIMENTAL RESULTS

In order to analyze PROFIBUS temporal behavior, the following metrics are defined:

- ***System Outage Time***: is the expected interruption time due to a token loss. This time is measured as the difference between the trigger of timeout timer and the time of the last frame transmitted on the bus;
- ***Station Outage Time***: is the time expected for a station to be inserted in the logical ring. This time is the difference between the insertion time and the station removal time (in order to avoid to measure the outage time twice, when a token loss occurs (*System Outage*) it ends the computation of the *Station Outage Time*);
- ***Bus Cycle Time***: is the time expected to complete one token rotation. This time is measured by the reception of two consecutive token frames. It begins after the reception of a token frame and ends at the reception of following token frame.

Since the network behavior is naturally stochastic (faults are injected in random instants), it is necessary to define adequate estimators for the measures taken [15]. Since the main aim is to obtain expected (mean) values, the estimators are defined as following: let m be the number of independent experiments, $X_{k,j}$ the total time of the observed events of type k (*System Outage, Station Outage, Bus Cycle*), in the experiment j and $N_{k,j}$ the total number of observed k events in the experiment j. By assuming that at the beginning of every experiment the ring is in steady-state, the type k estimator is given by:

$$\hat{\mu}_k = \frac{1}{m}\sum_{j=1}^{m}\frac{X_{k,j}}{N_{k,j}} \qquad (3)$$

The experiment length is established in 3.5 seconds in order to guarantee that the network steady-state regime is achieved. Behavioral data are obtained according to the *independent replication* method [15]. Estimators are defined using a 95% confidence interval with 5% of relative error (interval width) for the most frequent outage event (*System Outage*). Remain figures – *Station Outage Time* and *Bus*

Cycle Time were obtained with a relative error of 25% and 2% respectively.

A. System Outage Time

Fig. 2 shows the expected value for the *System Outage Time* as function of the BER and the BEL.

The results show that *System Outage Time* is not affected by the BEL. This is due to the fact that token losses are independent from the error pattern. However the *System Outage Time* is highly sensitive to the BER. It is observed that for BER=10^{-3} the expected *System Outage Time* is about 25 times higher than its theoretical value, obtained by the equation (1) T_{TO}=9600µs. This difference is clearly due to the timeout mechanism misbehavior for high BERs and justified by the errors periodicity at high BERs. In this case, the time between errors is smaller than 9600µs (timeout value of lower station address: 9). Thus, the timeout timer is restarted by every occurrence of bus errors for long periods, which significantly increases the time to recovery from a *System Outage Time*.

Fig. 3 shows the most relevant occurrence in the relative frequency of the *System Outage Time* as function of the BER. It can be observed that 90% of the *System Outage Time* is recovered within first times. But with the increase of the BER this value decreases, and only less than 10% is recovered in the first times for BER=10^{-3}.

For the experiment conditions the *Worst Case Recovery Time* (WCRcT) – maximum time to recover from an outage event - is obtained for BER=10^{-3} and its value is 2.09s.

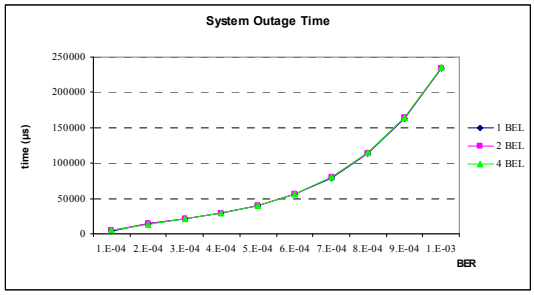

Fig. 2 - *System Outage Time*

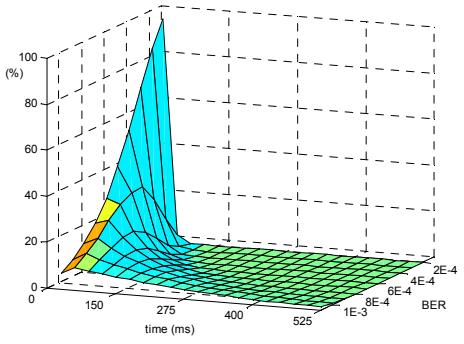

Fig.3 - Relative frequency - *System Outage Time*

B. Station Outage Time

The results (Fig. 4) show that the *Station Outage Time* is highly sensitive to undetected errors. This is observed by comparing the BEL=2 results with the BEL=1 and BEL=4 ones. This behavior is justified by the fact that some conditions that lead to a *Station Outage* event are caused by undetected errors. The events which are affected by the referred conditions are: *TS Address Error* and *LAS Inconsistency*.

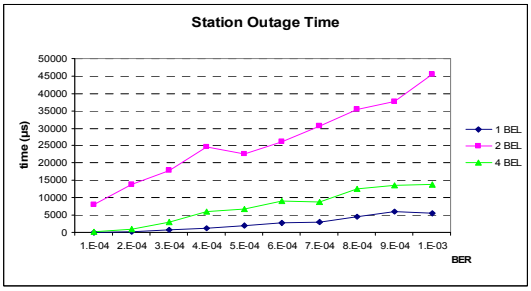

Fig. 4 - *Station Outage Time*

Fig. 5 shows the relative frequency *Station Outage Time* as function of BER for BEL=2 error pattern scenarios. Data shows that most of *Station Outage Time* recoveries are produced at relative lower times. The 3 spikes observed after 200ms correspond to recovery scenarios where the *LAS Inconsistency* event has its main contribution. In this case a significant number of stations are removed from logical ring due to inconsistent LAS - as consequence the *Station Outage Time* increases.

Major disturbances in the station insertion process are expected for high BERs, but smaller values are presented in Figs. 4 and 5 due to both station measurement method and high occurrence of *System Outage* events for high BERs.

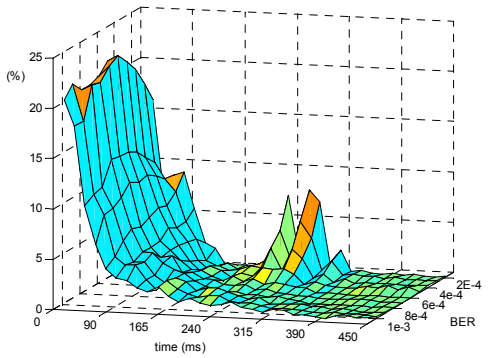

Fig. 5 - Relative frequency – *Station Outage Time*

C. Bus Cycle Time

Fig. 6 shows the expected *Bus Cycle Time* as function of the BER and the BEL. The results show that the BEL has very small effect on the expected Bus Cycle Time. In opposite the BER has an important influence. For lower BERs it is only observed a slight variation in respect to the expected Bus Cycle Time without errors. But for high BERs (above $4x10^{-4}$) the Bus Cycle Time verifies a very high increase.

This is an indicator that the performance of the PROFIBUS is highly affected in noisy environments.

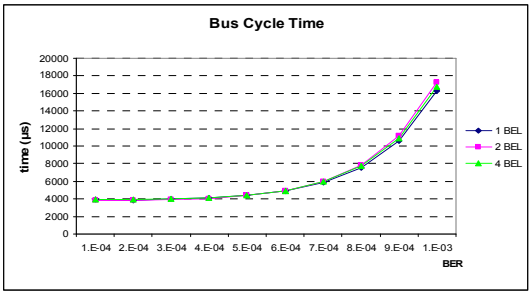

Fig. 6 - *Bus Cycle Time*.

VII. CONCLUSIONS

The paper presents a performance evaluation of PROFIBUS networks. This evaluation is performed through verifying the protocol response time when it works in faults scenarios. The evaluation was focused at the outage time associated with two scenarios of interruption of service: (i) *System Outage*, related to the token loss. (ii) *Station Outage*, that results from logical ring station removal. The evaluation is supported by a hardware platform which injects faults in a real PROFIBUS network.

The network operation is disturbed by a set of fault injection experiments and the following performance attributes are obtained, including: (i) *System Outage Time*, (iii) *Station Outage Time* (iii) *Bus Cycle Time*,

The analysis of data related to the above metrics allows establishing the following conclusions:

- The PROFIBUS network performance can be highly affected when operated in faulty environments such as industrial ones. This conclusion is based on the observation of the *Bus Cycle Time* for high BERs. In this case the magnitude of the Bus Cycle Time is so high that limits real-time response of the protocol;
- The *System Outage* is the worst case event for the protocol response time. The System Outage presents high sensitivity to the BER and its recovery mechanism (timeout) misbehaves in faulty scenarios. For a high BER the timeout mechanism presents an important recovery time. When it is compared with their theoretical value, it presents a 25 higher for the expected recovery time and 217 higher for the Worst Case Recovery Time. This is caused by multiple restarts of the timeout timer that are produced by errors when the bus is in idle state;
- The *Station Outage Time* is sensitive either to the BER or to the BEL. The station insertion mechanism used to recover from *System Outage* events presents a short recovery time compared with the timeout mechanism.

VIII. REFERENCES

[1] J. P. Thomesse, "A review of the Fieldbuses", *Annual Reviews in Control,* Vol. 22, pp. 35-45, 1998.

[2] H. Kim, A. White, K. Shin, "Effects of Electromagnetic Interference on Controller-Computer Upsets and System Stability", *IEEE Transactions on Control Systems Technology*, Vol. 8, pp. 351-357, 2000.

[3] H. Kim, K. Shin, "On the Maximum Feedback Delay in a Linear/Nonlinear Control System with Input Disturbances Caused by Controller-Computer Failures", *IEEE Transactions on Control Systems Technology*, Vol. 2, No. 2, pp. 110-122, 1994.

[4] K. Shin, H. Kim, "Derivation and Application of Hard Deadlines for Real-Time Control Systems", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 22, No. 6, pp. 1403-1413, 1992.

[5] EN 50170, *General Purpose Field Communication System*, Volume 2/3 (PROFIBUS), CENELEC, 1996.

[6] E. Tovar, F. Vasques, "Real-time Fieldbus Communications using Profibus Networks"; *IEEE Transactions on Industrial Electronics*, Vol. 46, No.6, pp. 1241-1251, 1999.

[7] E. Tovar, F. Vasques, "Setting Target Rotation Time in Profibus Based Real-Time Distributed Applications", *Proceedings of the 15th IFAC Workshop on Distributed Computer Control Systems*, 1998.

[8] H. Seung, K. Ki, "Implementation and Performance Evaluation of Profibus in the Automation Systems" in *Proceedings of the IEEE International Workshop on Factory Communication Systems*, 1997.

[9] A. Willing, A. Wolisz, "Ring Stability of the PROFIBUS Token-Passing Protocol Over Error-Prone Links", IEEE Transactions on Industrial Electronics, Vol. 48, No. 5, pp. 1025-1033, 2001.

[10] A. Willing, "Analysis and Tuning of the PROFIBUS Token Passing Protocol for Use over Error Prone Links", TKN Technical Report TKN-99-001, 1999.

[11] A. Willing, "Markov Modeling of the PROFIBUS Ring Membership over Error Prone Links", TKN Technical Re-port TKN-99-004, 1999.

[12] J. Carvalho, A. Carvalho, P Portugal, "Assessment of PROFIBUS Networks Using a Fault Injection Framework", *Proceedings of 10th Int. Conf. on Emerging Technologies and Factory Automation – ETFA'05*, 2005.

[13] J. Carvalho, P. Portugal, A. Carvalho, "A Framework for Dependability Evaluation of PROFIBUS Networks", *Proceedings of International Symposium in Industrial Electronics – ISIE'03*, 2003.

[14] *DSTni-LX Data Book, Revision E*, Grid Connect, 2003.

[15] A. Law, W. Kelton, *Simulation Modeling and Analysis – 3rd Edition*, McGraw-Hill, 2000.