Design procedure to minimize power consumption and delays in WSAN[1]

Diego Martínez *, Patricia Balbastre **, Francisco Blanes **, José Simo ** and Alfons Crespo **
* Departamento de Automática y Electrónica, Universidad Autónoma de Occidente
Cali, Colombia (e-mail: dmartinez@uao.edu.co)
** Departamento de Informática de Sistemas y Computadores, Universidad Politécnica de Valencia
Valencia, España (e-mail: patricia@disca.upv.es, pblanes@disca.upv.es, jsimo@disca.upv.es, alfons@disca.upv.es)

Abstract

Current trends in the development of industrial applications enforce the use of wireless networks to communicate the system nodes mainly to increase flexibility and reliability of these applications and to reduce the implementation cost. However, in control applications, as consequence of the latency and jitter generated by the network, not always the results achieved by the experimental results and desired performance are coherent. This is due to the imprecise models for system analysis and design used and the non appropriated validation methods and platforms to support these models. Therefore this paper presents a method to achieve an optimal system configuration in order to fulfil the desired performance in control applications with a significant energy saving and minimum delay.

1. Introduction

As a consequence to the increasing complexity of control systems, most of activities have been distributed over different nodes, where control loops are closed through a communication network. These systems are called Networked Control Systems (NCS). The implementation of NCS also reduces the impact of failures in a system component and facilitates the diagnosis, maintenance and traceability processes.

The MAC (Medium Access Control) algorithm used by the network characterizes the latency and jitter occurred during the transmission period. It produces discrepancy between experimental and simulation results. This is because imprecise models for analyzing and designing these systems are used, and to make use of inadequate validation methods and platforms that do not support the used models. The use of wireless networks to communicate the system nodes enables the development of new applications on wireless sensors and actuators networks (WSAN), which increases the application flexibility and reliability, at the same time its impact on the implementation cost reduction is significant.

In these applications the functions sensor, controller and actuator are distributed on the system nodes. As a result of physical connections of some nodes with the system, some functions will have a pre-assigned location, like sensor and actuator. While the other functions are assigned to any node according criteria related to fulfilment of time constraints, optimize the power consumption, minimizing latencies in the system, among others.

Consequently, to limited resources in nodes and the constraints imposed by applications, it is needed to develop methods for the cooperation between different levels of the system to achieve optimal solutions respecting to a particular criterion. One of the most important issues in the design process is that the minimization of some factors may cause the increase of another, then a compromise between them must be reached.

This paper presents a design methodology that includes different levels of node architecture, with the aim of finding solutions to ensure fulfilment of the temporal constraints and minimise the power consumption.

The paper is organized as follows. In section 2 is presented the related work. A proposal for the nodes architecture and a method to schedulability analysis for real time constraints in WSAN is presented in section 3. Section 4 presents the design procedure proposed. Simulation results of a case study are presented in section 5. Finally in section 6 conclusions and future work are presented.

2. Related works

This work integrates several aspects like network protocols, real-time analysis of messages and dynamic

[1] This work was partially supported by the Turtle project (SUBPROGRAMA AVANZA I+D Ministerio de Industria, Turismo y Comercio de España), and the SIDIRELI project DPI2008-06737-C02-01/02 (Ministerio de Ciencia e Innovación de España and European funds FEDER).

voltage analysis. In this section we analyse the related work.

2.1. Networks protocols

In [3], [5], [7] and [12] is analyzed the use of Bluetooth, IEEE 802.11b and 802.11e as control networks, and present a four layers architecture to achieve a predictable behaviour in IEEE 802.11b. Results presented allow concluding that IEEE 802.11e EDCA (Enhanced Distributed Channel Access) mode offers a good alternative to fulfil the real-time requirements of industrial applications.

[14] examines several methods for reducing the power consumption at different levels of the communication stack in wireless sensor networks, one of which is the Time Division Multiple Access (TDMA) MAC algorithm that contributes to ensure time transmission bounds, to minimize collisions, and power-aware.

In the literature, there are several commercial devices that use Zigbee [26] and WirelessHART [24], which have lower power consumption than systems supported by IEEE 802.11. Zigbee is supported by the IEEE 802.15.4-2003 protocol, which implements two basic types of media access, without synchronization and synchronization through beacons. The advantage of the MAC algorithm without synchronization is to facilitate the scalability and network autoconfiguration, however, there is not full guarantee of the time transmission bounds.

In the IEEE 802.15.4 synchronization mode the maximum time to transmit information can be bounded using guaranteed time slots (GTS) within a superframe. It is possible to assign a maximum of seven slots, with a minimum frame period of 15.36 ms, which may be enough in some cases. In [6] a method for the allocation of GTS is presented. However, the use of GTS is restricted to networks with star topology, which limits the reliability and scalability of the application.

WirelessHART is based on the physical layer of IEEE 802.15.4-2006 protocol, but specifies new levels of Data link, Network, Transport and Application, [8]. WirelessHART uses a MAC mode by TDMA, with 100 slots per second. Additionally WirelessHART allows developing mesh topologies networks providing redundant paths that permit routing messages through different routes to avoid broken links, interferences, and physical obstacles.

2.2. Feasibility testing for messages and real time tasks

By the effect of delays on the performance of NCS these applications have end-to-end real time constraints.

In the general formulation, the problem of assessing the feasibility of a real time distributed system is NP-hard. In order to overcome this inherent difficulty, problem restrictions and heuristics must be used. A common approach is to statically allocate application tasks at system nodes and locally utilize either a well known scheduling algorithm like Rate Monotonic (RM) or Earliest Deadline First (EDF), [19].

Distributed applications are characterized by precedence relationship between their tasks. If the tasks are statically allocated to single processors, end-to-end timing constraints can be analyzed by a theory which assumes release jitter [1]. Several papers have been developed oriented to analyze end-to-end schedulability, which have used tasks scheduling algorithms like RM and EDF, and MAC protocols based on TDMA, Token and Priorities [13], [15], [19], [20], [21]. These works consider the use of buffers to store messages in the network nodes and, employ a scheduling method to deliver the messages. They are based on finding the maximum response time of all messages.

About tasks schedulers, in the context of wireless sensor networks one of the most commonly used operating system is TinyOS [22]. It was designed to be used in systems with limited resources, such as 8-bit microcontrollers with small memory. It is supported by a programming model based on components and guided by events, with event handlers priority higher than tasks, which are executed based on scheduling policy First-Come First-Served (FCFS). However, such schedulers are not appropriate for real time systems. Zigbee products from Chipcon and Texas Instruments use a scheduler based on static priorities.

Although fixed priority scheduling is the most popular on-line scheduling policy in real-time systems, usage of the EDF policy is starting to get more attention in industrial environments, due to its benefits in the use of system resources. EDF is currently available in real-time languages such as RTSJ. It is also available in real time operating systems like SHark and Erika.

2.3. Dynamic voltage scaling

Several papers have been developed about Dynamic Voltage Scaling (DVS) ensuring fulfilment of real time constraints. In [11] a methodology based on heuristics for DVS is presented, which requires a low computation time. [18] shows a method to achieve the optimal operating frequency to minimum power consumption, however this method is very complex and therefore its use online is not appropriate. [9] proposes a method combining DVS with a task scheduler based on a task elastic model, which, according to various performance targets, adjusts the period of the system tasks. [25] uses a feedback control scheduling for the DVS processor and an EDF scheduler to scheduling tasks.

In [16] performances of several algorithms including static voltage scaling were examined, which selects the lowest operation frequency to fulfil real time constraints.

By using static voltage scaling, the operation frequency is assigned statically and is not modified unless the tasks set change. The advantages of this

method are its easy implementation and the very low load generated to the system. However, as result of performing the analysis using the worst case execution time it is very restrictive, so the highest savings energy is not obtained.

3. Nodes architecture and schedulability analysis

Design of these systems is characterized by application constraints. In [10] a classification of medical, industrial, environmental and agricultural applications is presented. Based on this study a node architecture was proposed, which is presented in Figure 1. This architecture is aimed for applications that require a bandwidth below to 250 kbps, have low computational requirements and operate in areas of small size, also enable the fulfilment of real time constraints and facilitate the implementation of strategies for saving energy.

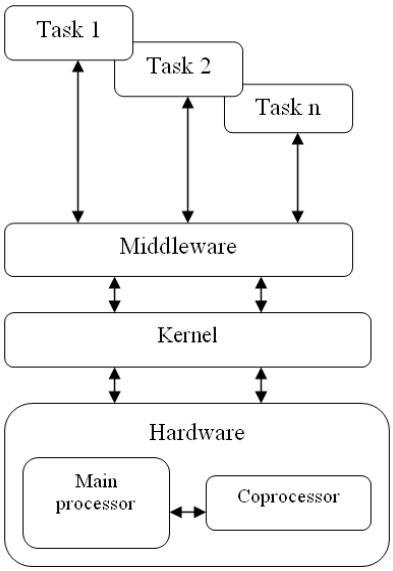

Figure 1. Logical architecture for nodes.

The roles of different levels of the architecture are:

- Tasks: perform activities related to the application.
- Middleware: Receive requests from tasks to execute a defined scenario, understanding it as a set of tasks with computation requirements and constraints known, and send commands to the kernel, which based on a static voltage scaling algorithm, sets the correct values of frequency and voltage to operate the processor, and select the tasks to be executed by the kernel for this scenario.
- Kernel: is a minimum kernel which performs the execution of tasks based on an EDF scheduling policy.
- Hardware: It consists of two processors, the main processor and the coprocessor. The main processor executes the software and allows the use of DVS strategies to save energy, processors with XScale architecture can be used. The coprocessor implements the physical, data link and network communications layers. So these activities do not affect the performance of processing functions, ensuring the application quality of service and the synchronization of network nodes. In relation to media access control uses a TDMA algorithm, which is appropriate to fulfil with real-time constraints and to facilitate the implementation of energy saving strategies.

In this proposal was assumed that all nodes are linked to the same network, only one hop is required to transmit a message, and the message storage in buffers was discarded. Additionally, because the size of messages in industrial applications is small compared to the amount of data supported by each message in current standard protocols (maximum payload in physical layer PDU of 127 bytes for WirelessHART and ZigBee), also was assumed that each message is sent in the interval reserved for every node in the TDMA network, then the maximum network delay is equal to the period to repeat the guaranteed time slots in the TDMA.

3.1. System Model and Notation

Whereas a general framework where the NCS's tasks, *Sensor*, *Controller* and *Actuator*, are executed in different nodes, in that sequence and using mutual exclusion, the following functions and concepts are defined:

- T_{SF}, is the period to repeat the slots in the TDMA based network.
- D_{CGR}, is the end–to–end deadline, measure from *Sensor* task start until *Actuator* task finalize, according to the control performance goals.
- T_S, is the sampling period used by the *Sensor* task, which is defined according to the dynamic system and comply with $D_{CGR} \leq T_S$.
- $\tau = \{Task_1, Task_2, ..., Task_n\}$, is a tasks set feasible by EDF policy with $Task_i = (WCET_i, D_i, P_i)$; $WCET_i$, D_i and P_i are the respective values of worst case execution time, deadline and period of task T_i.
- $WCRT_i$, is the worst case response time for a $Task_i$.

3.2. Schedulability Analysis

According to schedulability analysis in EDF [17]:

- $H_\tau(t) = \sum_{i=1}^{n} C_i \left\lfloor \frac{t + P_i - D_i}{P_i} \right\rfloor$, is the amount of computation time that has been attended by the processor until time *t* to fulfil with all deadlines in the system.

- Initial critical interval (ICI), is the time interval between time zero and the first time such that no outstanding computation exist, $[0, R)$.

The schedulability test consists in to verify that

$$H_\tau(t) < t \quad \forall \quad t \le R \qquad (1)$$

So to verify the end–to–end schedulability for tasks *Sensor*, *Controller* and *Actuator*, according to the previous assumptions, will consist in verifying if D_S, D_C and D_A fulfil with expressions presents in table 1, with D_S, D_C and D_A are the deadlines of *Sensor*, *Controller* and *Actuator* tasks respectively.

In order to select D_S, D_C and D_A for implementation a value between [$D^{min}{}_j$, $D^{max}{}_j$] can be used, which $D^{min}{}_j$ and $D^{max}{}_j$ are the minimum and maximum values of D_j to local and end–to–end schedulability. $D^{min}{}_j$ is obtained from *Deadlinemin* algorithm presented in [2].

Depending on the physical characteristics of the system, three more architectures can be presented for the control architecture, and previous parameters must be modified as follows:

- *Sensor* and *Controller* in the same node: $WCRT_S^{'} = WCRT_S$.

- *Controller* and *Actuator* in the same node: $WCRT_C^{'} = WCRT_C$.

- *Sensor* and *Actuator* in the same node: In this case there are not changes.

4. WSAN DESIGN PROCEDURE

The design procedure proposed is presented in figure 2. The general considerations are:

- Number and location of nodes is known.

- Schedulability tests and system configuration analysis are applied offline.

- In order to choose the best system configuration, for any scenario, an optimization process is used.

- Delays produced by switching tasks and changes of system configuration were not taken in to account.

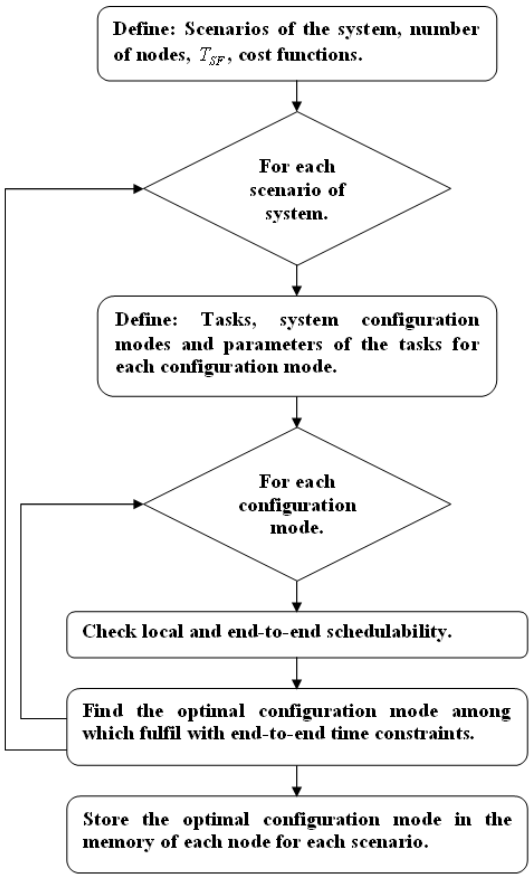

Figure 2. WSAN design procedure.

- In order to activate tasks for each application scenario, and considering than in these applications the number of task unassigned to a specific node and its code is reduced, it is supposed that each node has a replica of every unassigned task which execute depending on the scenario of the system.

Table 1. Tasks Parameters to verify end–to–end schedulability

Task Sensor	Task Controller	Task Actuator
Periodic, $P_S = T_S$	Sporadic, $P_{C_{Min}} = T_S$	Sporadic, $P_{A_{Min}} = T_S$
$WCET_S \le WCRT_S \le D_S$	$WCET_C \le WCRT_C \le D_C$	$WCET_A \le WCRT_A \le D_A$
$D_S^{\max} = D_{CGR} - (WCRT_C^{'} + WCRT_A + T_{SF})$	$D_C^{\max} = D_{CGR} - (WCRT_S^{'} + WCRT_A + T_{SF})$	$D_A^{\max} = D_{CGR} - (WCRT_S^{'} + WCRT_C^{'})$
$WCRT_S^{'} = WCRT_S + T_{SF}$	$WCRT_C^{'} = WCRT_C + T_{SF}$	

- There are not communication errors.
- Main processor and coprocessor are not synchronized.
- Local and end-to-end schedulability are verified by to apply equation (1) and equations in table 1 respectively.
- The periods and deadlines of tasks are defined by the application. To get the task's WCET for each processor operation frequency, is assumed to be known the task's WCET for the lowest frequency and then the WCET for the other frequencies is calculate using a frequency scaling factor, λ, therefore:

$$f_i = \lambda_i * f_{low}, \text{ then } WCET_{f_i} = \frac{1}{\lambda_i} WCET_{f_{low}}$$

In XScale processors different operating frequencies can be obtained from multiplying by powers of two the system clock frequency.

For present the proposed design procedure a case study with three nodes was considered, on which a NCS and an online faults detection system were implemented.

4.1. General assumptions for the case study

To simplify the space of solutions to the problem, we assume homogeneous nodes with two processor operating frequencies, low and high. In order to observe easily the effect of different configuration modes in the power consumption a scaling factor, λ, of 10 was considered.

To calculate the WCET for sensor tasks when high frequency is used, a value to represent the response time of analog-to-digital converter (ADC) has been added, which does not scale.

Network parameters are:

- Frame size = 144 bits.
- Data rate = 250 kbps.
- Guaranteed time slots = 1 ms.
- T_{SF} = 35 ms.

4.2. Faults detection system assumptions

This application consists of two Tasks. *D_Sensor* captures the samples needed to implement the fault detection algorithms, is periodic and is executed until to get the number of samples required by the algorithms. The other task, called *Diagnostic_techniques*, is executed only one time and performs the fault detection algorithms, this task doesn't have real time constrains, the purpose is to finalize its execution as soon as possible, then for analysis effects a deadline of 5 s was considered.

It is assumed that, according to the physical configuration of the system, *D_Sensor* will be allocated on node 1, and *Diagnostic_techniques* can be executed on any of the three nodes. The assumed WCET for these tasks are presented in table 2.

In the case considered, it is assumed that the system degradation due to a fault does not occur quickly, so the diagnostic is applied only one time per day, in addition this task does not require a strict deadline.

Table 2

Task	WCET, low frequency	WCET, high frequency
D_Sensor	2 ms	0.3 ms
Diagnostic_ techniques	1000 ms	100 ms

When *D_Sensor* is activated it gets 2048 samples of variables related to failures to be detect, with a sample period of 4 ms. When this task complete the measurements, sends a signal to the middleware in order to activate the *Diagnostic_Techniques* task, which processes the samples and generates a status report of the plant. This means that these two tasks are executed in mutual exclusion. The periods and deadlines of these tasks are, $T_{D\_Sensor} = 4ms$ and $D_{D\_Sensor} = 4ms$, the *Diagnostic_Techniques* task has a period of $24h$ and deadline $D_{Diagnostic\_techniques} = 5000ms$.

4.3. NCS assumptions

Assumptions about the NCS system, figure 3, are:

- Single Input Single Output (SISO) system.
- The NCS requires three tasks, *C_Sensor*, *C_Controller* and *C_Actuator*, which are allocated on nodes 1, 2 and 3 respectively.
- The delay from *C_Sensor* task start until *C_Actuator* task finalized, τ_r, is less than or equal to T_S.
- The *C_Sensor* task is time-driven, with a sampling period T_S, but *C_Controller* and *C_Actuator* tasks are event-driven.

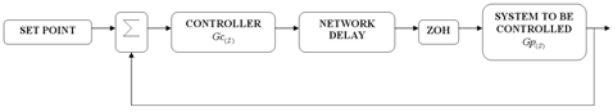

Figure 3. Representation of the NCS.

For the case study $Gp_{(S)} = \frac{0.3}{0.86s + 1}$. In order to reduce the settling-time the following PI control algorithm was used:

$$U_{(z)} = \frac{k_p z + (k_i T_s - k_p)}{z-1}\varepsilon_{(z)},\ kp=30.2778,\ ki=25.463.$$

It is assumed an accepted performance for the NCS like presented in figure 4, which is generated with $T_S = 110ms$ and $\tau_r = 80ms$, then $D_{CGR} = 80ms$.

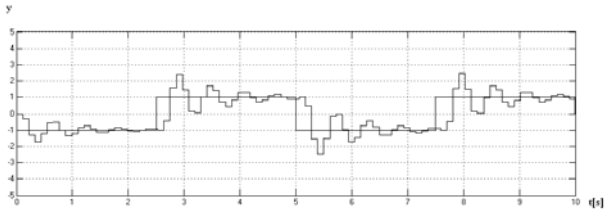

Figure 4. Control system response to a square input signal, with $T_S = 110ms$ and $\tau_r = 80ms$.

WCET for *C_Sensor*, *C_Controller* and *C_Actuator* are presented in table 3.

Table 3

Task	WCET, low frequency	WCET, high frequency
C_Sensor	2 ms	0.3 ms
C_Controller	8 ms	0.8 ms
C_Actuator	1 ms	0.1 ms

4.4. Cost functions

To analyze each of the targets two cost functions have been proposed, $F(Co)_{S\_Comp}$ for assessing the cost of a configuration taking into account parameters related to the computer system, and $F(Co)_{Control}$ to evaluate quality of the control parameters that affect the dynamic characteristics of the control system, each of which has the following form:

$$F(C_o) = \sum_0^i K_i \frac{C_i(C_o)}{C_i} + \sum_0^i K_{Ci} F_C(C_i, C_i(C_o))$$

Ki is the weight assigned to the parameter *i*, *Ci(Co)* is the solution of a given configuration *Co*, *Ci* is the design constraint applied to the i-th parameter and it is used as a normalization factor; *Kci* is a weight assigned to the correction factors and *Fc* is a correction function. Its individual weights depend on the context peculiarities for a specific application. The correction function does not contribute to the cost function when the solution is within the allowable search space. The expression of the correction function is:

$$F_C(C_i, C_i(C_o)) = \max\left\{0, \frac{[C_i(C_o) - C_i]}{C_i}\right\}^2$$

The target for the computer system is to generate an optimal solution to increase the lifetime of the WSAN, taking into consideration that all nodes depend on batteries for their operation. It is restricted mainly by the availability of energy in each of the network nodes. This is achieved by using the lowest operation frequency for each node, but getting a similar percentage of use in the nodes. Possible actions to influence the power consumption are:

- Voltage and frequency scaling in the processor. It affects the power consumption in a single node and scale the $WCET_i$ and $WCRT_i$ in the respective node.
- Migration of tasks between the nodes, which is achieved by activating the respective task in the selected node. It tries to find to the balance of power consumption and affects $WCRT_i$ in several nodes.

In the approach used in this work an indirect analysis of power consumption for different system configurations is realized, taking into account the following indexes:

- fop_{M_j}, is the average of the nodes operation frequency for the configuration *j* in the WSAN.
- Δfop_{DEj}, is the standard deviation for the nodes operation frequencies.
- ΔU_{DEj}, is the standard deviation of the utilization in the main processor, with the utilization for *i* tasks is $U = \sum_{i=1}^{n} \frac{WCET_i}{T_i}$.

Regarding the performance of the control system, two parameters were considered. These are:

- r_M, is the average of delays τ_r.
- Δr_{DE}, is the standard deviation of delays τ_r, ideally it should tends to zero, then the delay can be easily compensated by the control algorithm.

The maximum operating values are selected according to the architecture for synthesis of components which have been selected, the applications constraints and the parameters for each scenario. These values restrict the choice of the best system configuration for any scenario.

4.5. Obtaining the optimal configuration mode

There are three possible scenarios for the case study proposed:

I. Executing the control application.

II. Executing the control application and the *D_Sensor* task.

III. Executing the control application and the *Diagnostic_techniques* task.

To get the temporal parameters for each system configuration, necessaries to apply costs functions, the

case was simulated by using Truetime [4]. The values assigned to parameters were:

- Computer system: $K_{f_M} = 0.5$, $K_{\Delta f_{DE}} = 0.2$, $K_{\Delta U_{DE}} = 0.3$, $K_{cf_M} = K_{C\Delta f_{DE}} = K_{C\Delta U_{DE}} = 150$.
- Control system: $Kr_M = 0.6$, $K\Delta r_{DE} = 0.4$, $K_{cr} = K_{\Delta r} = 150$.

40 configurations were obtained, out of which 30 fulfil with end–to–end deadlines. In order to find the optimal configuration mode the Pareto front was observed. Figure 5 shows the Pareto front for scenario III. As can be seen, the optimization of both criteria is a multiobjective problem.

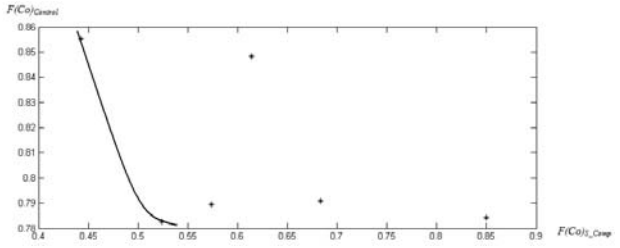

Figure 5. Pareto front for scenario III.

Because the Pareto front is convex, it is possible to find the optimal solution to multiobjective problem using a linear combination of weights, then the new cost function is:

$$F(C_o) = K_{S\_Comp} F(C_o)_{S\_Comp} + K_{Control} F(C_o)_{Control}$$

$K_{Cómputo}$ and $K_{Control}$ are coefficients indicating individual weight for each function for the desired solution quality, which depends on the type of application. The coefficients used for the case study were: $K_{S\_Comp} = 0.4$, $K_{Control} = 0.6$.

The results of the optimization process were:

- Scenario I, the optimal operation mode is when all nodes are operating at high frequency.
- Scenario II, the optimal operation mode is when node 1 is operating to high frequency, and nodes 2 and 3 to low frequency.
- Scenario III, the optimal operation mode is when all nodes are operating at high frequency, and *Diagnostic_techniques* is executed in node 3.

5. Simulation results

[23] presents the following model of power consumption for XScale processors:

$$P = P_{Static} + P_{Dinamic} \Rightarrow P = VI_{Leakage} + \frac{1}{2} C_L V_{dd}^{\ 2} f$$

The model used for simulation, considering the same factor for scaling frequency and voltage is:

- During the execution time of each task, $P = 0.01 + 0.02\left(\frac{f_{operation}}{f_{low}}\right)^3$
- At the moment when the ICI interval is finished, the processor will start a low power state, then $P = 0.001$.

The simulation results coincide with the analysis made by the optimization process. Figures 6 and 7 show the response of the control system for a configuration mode which minimize $F(Co)_{Control}$, and the configuration mode which minimize $F(Co)_{S\_Comp}$ in scenario III respectively.

As can be seen, because the two configurations fulfil with end-to-end deadlines proposed, in both cases the control system response is coherent with accepted performance for the NCS, although there is a better response in the configuration presented in figure 6 by presenting a smaller delay.

However, the power consumption in the nodes is higher in the configuration presented in figure 6, then the lifetime of the application is decreased, and the system will be unstable 7s after the simulation start as a consequence of the completion of energy at node 2.

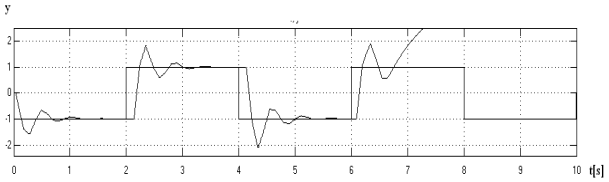

Figure 6. Response of the control system for a configuration mode which minimize $F(Co)_{Control}$ in scenario III.

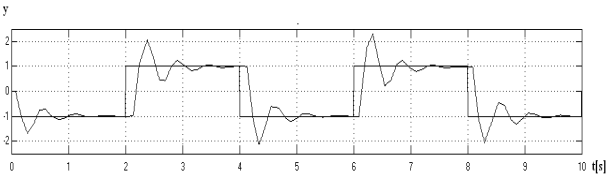

Figure 7. Response of the control system for the configuration mode which minimize $F(Co)_{S\_Comp}$ in scenario III.

6. Conclusions

A method to get the optimal configuration of the system, supported on cost functions related to power consumption and delays in the control loops was presented. The results show the importance of realizing balance between parameters in the design of WSAN in order to minimize the power consumption and delays. In addition by the use of the design method proposed the control specifications have been fulfilled.

Future work will focus on to include the routing protocols in the design procedure, and to automate the optimization process and integrate heuristics to facilitate the process and reducing the number of options examined.

References

[1] Audsley N., Burns A., Richardson M., Tindell K., and Wellings A.J., "Applying New Scheduling Theory to Static Priority Pre-emptive Scheduling". *Software Engineering Journal*, September 1993.

[2] Balbastre P., Ripoll I., and Crespo A. "Minimum Deadline Calculation for Periodic Real-Time Tasks in Dynamic Priority Systems". *IEEE Transactions on computers*, vol. 57, No. 1, January 2008.

[3] Cena I G., Bertolotti A C., C. Zunino V. "Industrial Applications of IEEE 802.11e WLANs". *IEEE International Workshop on Factory Communication Systems*, 21-23 May 2008.

[4] Cervin A., Henriksson D., Lincoln B., Eker J., Årzén K.: "How Does Control Timing Affect Performance?, Analysis and Simulation of Timing Using Jitterbug and TrueTime". *IEEE Control Systems Magazine*, June 2003.

[5] Hristu-Varsakelis D., Levine W. S. (Eds.): *Handbook of Networked and Embedded Control Systems*. Pag: 677-720. Birkhäuser 2005.

[6] Koubâa A., Alves M., and Tovar E., "GTS Allocation Analysis in IEEE 802.15.4 for Real-Time Wireless Sensor Networks", *14th International Workshop on Parallel and Distributed Real-Time Systems (WPDRTS 2006)*. Rhodes Island (Greece): IEEE, 2006.

[7] Lee S., Park J. H., Ha K. N., Lee K. C. "Wireless Networked Control System Using NDIS-based Four-Layer Architecture for IEEE 802.11b". *IEEE International Workshop on Factory Communication Systems*, 21-23 May 2008.

[8] Lennvall T., Svensson S., Hekland F. "A Comparison of WirelessHART and Zigbee for Industrial Applications". *IEEE International Workshop on Factory Communication Systems*, 21-23 May 2008.

[9] Marinoni M., Buttazzo G. "Elastic DVS Management in Processors With Discrete Voltage/Frequency Modes". *IEEE Transactions on industrial informatics*, vol. 3, No. 1, 2007.

[10] Martínez D., Blanes F., Simo J., Crespo A. "Wireless sensor and actuator networks: Characterization and case study for confined spaces healthcare and control applications". *Scalable computing: practice and experience scientific, International journal for parallel and distributed computing.* Vol. 10, no.3. pp. 291 – 305. 2009.

[11] Mejia-Alvarez P., Levner E., and Mosse D., "Power-Optimized Scheduling Server for Real-Time Tasks," *Proc. IEEE Real-Time and Embedded Technology and Applications Symp. (RTAS '02)*, p. 239, 2002.

[12] Moraes R., Portugal P., Vasques F., Fonseca J. A. "Limitations of the IEEE 802.11e EDCA Protocol when Supporting Real-Time Communication". *IEEE International Workshop on Factory Communication Systems*, 21-23 May 2008.

[13] Palencia Gutierrez, J.C.; Gutierrez Garcia, J.J.; Gonzalez Harbour, M. "On the schedulability analysis for distributed hard real-time systems". *Real-Time Systems, 1997. Proceedings., Ninth Euromicro Workshop* on 11-13 June 1997 Page(s):136 – 143.

[14] Pantazis, N.A.; Vergados, D.D. "A survey on power control issues in wireless sensor networks". *IEEE Communications Surveys & Tutorials*, 4th Quarter 2007. vol 9, No. 4.

[15] Pereira N., Tovar E. and Andersson B. "Exact Analysis of TDMA with Slot Skipping". *13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA 2007)*. IEEE Computer Society 2007.

[16] Pillai P. and Shin K.G., "Real-Time Dynamic Voltage Scaling for Low-Power Embedded Operating Systems," *Proc. ACM Symp. Operating Systems Principles*, pp. 89-102, 2001.

[17] Ripoll I., Crespo A., and Mok A. "Improvement in feasibility testing for real-time tasks". *Journal of Real-Time Systems*, 11:19–40, 1996.

[18] Saewong S. and Rajkumar R. "Practical Voltage-Scaling for Fixed-Priority RT-Systems". *Proceedings of the 9th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'03)* 2003 IEEE.

[19] Spuri M. "Holistic Analysis for Deadline Scheduled Real-Time Distributed Systems". *Tech. Rep. RR-2873, INRIA*, France, April 1996.

[20] Tindell K. and Clark J., "Holistic Schedulability Analysis for Distributed Hard Real-Time Systems", *Microprocessors and Microprogramming* 40, 1994.

[21] Tindell K., Burns A., and Wellings A.J., "Analysis of Hard Real-Time Communications", *The Journal of Real-Time Systems 9*, 1995.

[22] TinyOS: http://www.tinyos.net/

[23] Varma A., Debes E., Kozintsev I. and Jacob B., "Instruction-Level Power Dissipation in the Intel XScale Embedded Microprocessor". *Proceedings of the SPIE, 17th Annual Symposium on Electronic Imaging Science & Technology*, vol. 5683, pp. 1-8. 2005.

[24] WirelessHART: http://www.hartcomm2.org/hart_protocol/wireless_hart/wirelesshart_datasheet.pdf

[25] Zhu Y. and Mueller F. "Feedback EDF Scheduling Exploiting Dynamic Voltage Scaling". *Proceedings of the 10th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'04)*, 2004 IEEE.

[26] Zigbee Specification. http://www.Zigbee.org